

Ildar Batyrshin
Grigori Sidorov (Eds.)

LNAI 7094

Advances in Artificial Intelligence

10th Mexican International Conference
on Artificial Intelligence, MICAI 2011
Puebla, Mexico, November/December 2011, Proceedings, Part I

1
Part I



Springer

Lecture Notes in Artificial Intelligence 7094

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

Ildar Batyrshin Grigori Sidorov (Eds.)

Advances in Artificial Intelligence

10th Mexican International Conference
on Artificial Intelligence, MICAI 2011
Puebla, Mexico, November 26 – December 4, 2011
Proceedings, Part I

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada

Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Ildar Batyrshin

Mexican Petroleum Institute (IMP)

Eje Central Lazaro Cardenas Norte, 152

Col. San Bartolo Atepehuacan

Mexico D.F., CP 07730, Mexico

E-mail: batyr1@gmail.com

Grigori Sidorov

National Polytechnic Institute (IPN)

Center for Computing Research (CIC)

Av. Juan Dios Bátiz, s/n, Col. Nueva Industrial Vallejo

Mexico D.F., CP 07738, Mexico

E-mail: sidorov@cic.ipn.mx

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-642-25323-2

e-ISBN 978-3-642-25324-9

DOI 10.1007/978-3-642-25324-9

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011940855

CR Subject Classification (1998): I.2, I.2.9, I.4, F.1, I.5.4, H.3-4

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The Mexican International Conference on Artificial Intelligence (MICAI) is a yearly international conference series organized by the Mexican Society of Artificial Intelligence (SMIA) since 2000. MICAI is a major international AI forum and the main event in the academic life of the country's growing AI community.

This year's event was very special: we celebrated the 25th anniversary of SMIA and 10th anniversary edition of the MICAI series.

MICAI conferences traditionally publish high-quality papers in all areas of artificial intelligence and its applications. The proceedings of the previous MICAI events have been published by Springer in its *Lecture Notes in Artificial Intelligence* (LNAI) series, vol. 1793, 2313, 2972, 3789, 4293, 4827, 5317, 5845, 6437 and 6438. Since its foundation in 2000, the conference has been growing in popularity and improving in quality.

The proceedings of MICAI 2011 have been published in two volumes. The first volume, *Advances in Artificial Intelligence*, contains 50 papers structured into five sections:

- Automated Reasoning and Multi-agent Systems
- Problem Solving and Machine Learning
- Natural Language Processing
- Robotics, Planning and Scheduling
- Medical Applications of Artificial Intelligence

The second volume, *Advances in Soft Computing*, contains 46 papers structured into five sections:

- Fuzzy Logic, Uncertainty and Probabilistic Reasoning
- Evolutionary Algorithms and Other Naturally Inspired Algorithms
- Data Mining
- Neural Networks and Hybrid Intelligent Systems
- Computer Vision and Image Processing

Both books will be of interest for researchers in all fields of AI, students specializing in related topics and for the general public interested in recent developments in AI.

The conference received 348 papers submitted for evaluation, by 803 authors from 40 countries; of these, 96 papers were selected for publication after a peer-reviewing process carried out by the international Program Committee. The acceptance rate was 27.5%.

The distribution of submissions by country or region is represented in Fig. 1, where the square of each circle corresponds to the number of submitted papers. Table 1 shows more detailed statistics. In this table, the number of papers is by authors: e.g., for a paper by 2 authors from USA and 1 author from UK, we added 2/3 to USA and 1/3 to UK.

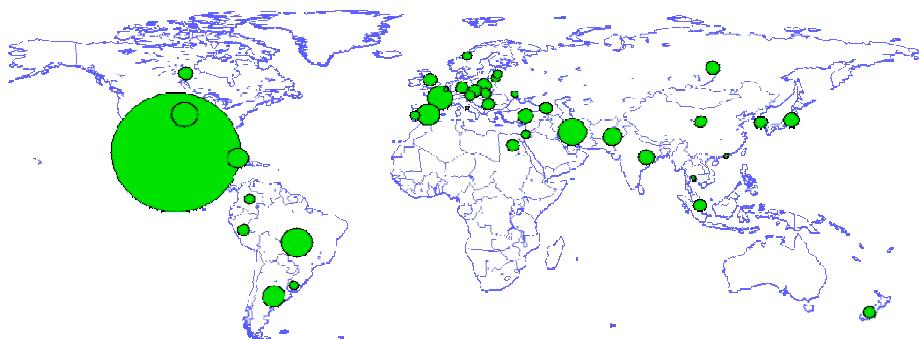


Fig. 1. Distribution of submissions by country or region.

Table 1. Submitted and accepted papers by country or region.

Country or region	Authors	Subm.	Acc.	Country or region	Authors	Subm.	Acc.
Argentina	13	7	3	Latvia	1	1	1
Austria	3	1.53	0.33	Lithuania	9	1	—
Belgium	1	0.25	—	Mexico	527	227.64	62.27
Brazil	35	13.25	3	New Zealand	5	2	1
Canada	8	2.6	1.6	Norway	1	1	—
China	5	2	—	Pakistan	11	4.92	1.42
Colombia	3	1.5	0.5	Peru	3	2	1
Cuba	15	6.21	1.75	Poland	5	3	1
Czech Rep.	4	2.5	1	Portugal	4	1	—
Egypt	5	2	—	Russian Federation	7	2.67	1
France	25	8.95	3.12	Serbia	4	2	—
Georgia	2	2	—	Singapore	2	2	1
Germany	3	2	1	Slovakia	2	1.5	—
Hong Kong	1	0.33	0.33	Spain	24	7.07	2.42
India	8	3.42	0.75	Thailand	1	0.33	—
Iran	16	11	2	Turkey	4	3	—
Israel	3	1.17	0.67	Ukraine	2	0.5	0.5
Italy	1	0.17	—	United Kingdom	6	2.32	1.32
Japan	7	3	1	United States	19	9.18	3.03
Korea, Rep. of	5	2	—	Uruguay	3	1	—

The authors of the following papers received the Best Paper Award on the basis of the paper's overall quality, significance and originality of the reported results:

- 1st place: *SC Spectra: A New Soft Cardinality Approximation for Text Comparison*, by Sergio Jimenez Vargas and Alexander Gelbukh (Colombia, Mexico)
- 2nd place: *Fuzzified Tree Search in Teal Domain Games*, by Dmitrijs Rutko (Latvia)
- 3rd place: *Multiple Target Tracking with Motion Priors*, by Francisco Madrigal, Jean-Bernard Hayet and Mariano Rivera (Mexico)

In addition, the authors of the following papers selected among articles where the first author was a full-time student (excluding the papers listed above) received the Best Student Paper Award:

- 1st place: *Topic Mining Based on Graph Local Clustering*, by Sara Elena Garza Villarreal and Ramon Brena (Mexico)
- 2nd place: *Learning Probabilistic Description Logics: A Framework and Algorithms*, by Jose Eduardo Ochoa-Luna, Kate Revoredo and Fabio Gagliardi Cozman (Brazil)
- 3rd place: *Instance Selection Based on the Silhouette Coefficient Measure for Text Classification*, by Debangana Dey, Thamar Solorio, Manuel Montes y Gomez and Hugo Jair Escalante (USA, Mexico)

We want to thank all the people involved in the organization of this conference. In the first place, these are the authors of the papers published in this book: it is their research work that gives value to the book and to the work of the organizers. We thank the Track Chairs for their hard work, the Program Committee members and additional reviewers for their great effort spent on reviewing the submissions.

We would like to express our sincere gratitude to the Benemérita Universidad Autónoma de Puebla (BUAP), the Rector's Office of the BUAP headed by Dr. Enrique Agüera Ibañez; Dr. José Ramón Eguibar Cuenca, Secretary General of the BUAP; Alfonso Esparza Ortiz, Treasurer General of the BUAP; José Manuel Alonso of DDIE; Damián Hernández Méndez of DAGU; Dr. Lilia Cedillo Ramírez, Vice-rector of Extension and Dissemination of Culture of the BUAP; Dr. Gabriel Pérez Galmichi of the Convention Center; Dr. Roberto Contreras Juárez, Administrative Secretary of the Faculty of Computer Science of the BUAP; and to MC Marcos González Flores, head of the Faculty of Computer Science of the BUAP, for their warm hospitality related to MICAI 2011 and for providing the infrastructure for the keynote talks, tutorials and workshops, as well as for their valuable participation and support in the organization of this conference.

Their commitment allowed the opening ceremony, technical talks, workshops and tutorials to be held at the Centro Cultural Universitario, an impressive complex of buildings that bring together expressions of art, culture and academic affairs associated with the BUAP.

We are deeply grateful to the conference staff and to all members of the Local Committee headed by Dr. David Eduardo Pinto Avendaño. In particular, we would like to thank Dr. Maya Carrillo for chairing the logistic affairs of the conference, including her valuable effort for organizing the cultural program; Dr. Lourdes Sandoval for heading the promotion staff; as well as Dr. Arturo Olvera, head of the registration staff, Dr. Iván Olmos, Dr. Mario Anzures, and Dr. Fernando Zacarías (sponsors staff) for obtaining additional funds for this conference.

We also want to thank the sponsors that provided partial financial support to the conference: CONCYTEP, INAOE, Consejo Nacional de Ciencia y Tecnología (CONACYT) project 106625, TELMEX, TELCEL, Universidad Politécnica de

Puebla, UNIPUEBLA and Universidad del Valle de Puebla. We also thank Consejo de Ciencia y Tecnología del Estado de Hidalgo for partial financial support through the project FOMIX 2008/97071. We acknowledge support received from the following projects: WIQ-EI (Web Information Quality Evaluation Initiative, European project 269180), PICCO10-120 (ICYT, Mexico City Government) and CONACYT-DST (India) project “Answer Validation through Textual Entailment.”

The entire submission, reviewing and selection process as well as putting together the proceedings were supported for free by the EasyChair system (www.easychair.org). Last but not least, we are grateful to Springer for their patience and help in preparation of this volume.

September 2011

Ildar Batyrshin
Grigori Sidorov

Conference Organization

MICAI 2011 was organized by the Mexican Society of Artificial Intelligence (SMIA, Sociedad Mexicana de Inteligencia Artificial) in collaboration with Benemérita Universidad Autónoma de Puebla (BUAP), Centro de Investigación en Computación del Instituto Politécnico Nacional (CIC-IPN), Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Universidad Nacional Autónoma de México (UNAM), Universidad Autónoma de México (UAM), Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM), Universidad Autónoma de Estado de Hidalgo (UAEH) and Instituto Mexicano de Petróleo (IMP), Mexico.

The MICAI series website is www.MICAI.org. The website of the Mexican Society of Artificial Intelligence, SMIA, is www.SMIA.org.mx. Contact options and additional information can be found on these websites.

Conference Committee

General Chair	Raúl Monroy
Program Chairs	Ildar Batyrshin and Grigori Sidorov
Workshop Chair	Alexander Gelbukh
Tutorials Chairs	Felix Castro Espinoza and Sofía Galicia Haro
Keynote Talks Chair	Jesus A. Gonzalez
Financial Chair	Grigori Sidorov
Grant Chairs	Raúl Monroy, Grigori Sidorov and Ildar Batyrshin
Best Thesis Awards Chair	Miguel Gonzalez
Doctoral Consortium Chairs	Oscar Herrera and Miguel Gonzalez
Organizing Committee Chair	David Pinto Avendaño

Track Chairs

Natural Language Processing	Sofia Galicia Haro
Machine Learning and Pattern Recognition	Mario Koeppen
Hybrid Intelligent Systems and Neural Networks	Sergio Ledesma Orozco
Logic, Reasoning, Ontologies, Knowledge Management, Knowledge-Based Systems, Multi-agent Systems and Distributed AI	Miguel González and Raul Monroy
Data Mining	Felix Castro Espinoza
Intelligent Tutoring Systems	Alexander Gelbukh
Evolutionary Algorithms and Other Naturally Inspired Algorithms	Nareli Cruz Cortés
Computer Vision and Image Processing	Oscar Herrera
Fuzzy Logic, Uncertainty and Probabilistic Reasoning	Alexander Tulupiyev
Bioinformatics and Medical Applications	Jesús A. González
Robotics, Planning and Scheduling	Fernando Montes

Program Committee

Carlos Acosta	Mario Chacon
Hector-Gabriel Acosta-Mesa	Lee Chang-Yong
Luis Aguilar	Niladri Chatterjee
Ruth Aguilar	Zhe Chen
Esma Aimeur	Carlos Coello
Teresa Alarcón	Ulises Cortes
Alfonso Alba	Stefania Costantini
Rafik Aliev	Raúl Cruz-Barbosa
Adel Alimi	Nareli Cruz-Cortés
Leopoldo Altamirano	Nicandro Cruz-Ramirez
Matias Alvarado	Oscar Dalmau
Gustavo Arechavaleta	Ashraf Darwish
Gustavo Arroyo	Justin Dauwels
Serge Autexier	Radu-Codrut David
Juan Gabriel Aviña Cervantes	Jorge De La Calleja
Victor Ayala-Ramirez	Carlos Delgado-Mata
Andrew Bagdanov	Louise Dennis
Javier Bajo	Bernabe Dorronsoro
Helen Balinsky	Benedict Du Boulay
Sivaji Bandyopadhyay	Hector Duran-Limon
Maria Lucia Barrón-Estrada	Beatrice Duval
Roman Barták	Asif Ekbal
Ildar Batyrshin (Chair)	Boris Escalante Ramírez
Salem Benferhat	Jorge Escamilla Ambrosio
Tibebe Beshah	Susana C. Esquivel
Albert Bifet	Claudia Esteves
Igor Bolshakov	Julio Cesar Estrada Rico
Bert Bredeweg	Gibran Etcheverry
Ramon Brena	Eugene C. Ezin
Paul Brna	Jesus Favela
Peter Brusilovsky	Claudia Feregrino
Pedro Cabalar	Robert Fisher
Abdiel Emilio Caceres Gonzalez	Juan J. Flores
Felix Calderon	Claude Frasson
Nicoletta Calzolari	Juan Frausto-Solis
Gustavo Carneiro	Olac Fuentes
Jesus Ariel Carrasco-Ochoa	Sofia Galicia-Haro
Andre Carvalho	Ma.de Guadalupe Garcia-Hernandez
Mario Castelán	Eduardo Garea
Oscar Castillo	Leonardo Garrido
Juan Castro	Alexander Gelbukh
Félix Agustín Castro Espinoza	Onofrio Gigliotta
Gustavo Cerda Villafana	Duncan Gillies

Fernando Gomez	Sergio Ledesma-Orozco
Pilar Gomez-Gil	Yoel Lledo Mezquita
Eduardo Gomez-Ramirez	Eugene Levner
Felix Gonzales	Derong Liu
Jesus Gonzales	Weiru Liu
Arturo Gonzalez	Giovanni Lizarraga
Jesus A. Gonzalez	Aurelio Lopez
Miguel Gonzalez	Omar Lopez
José-Joel Gonzalez-Barbosa	Virgilio Lopez
Miguel Gonzalez-Mendoza	Gabriel Luque
Felix F. Gonzalez-Navarro	Sriram Madurai
Rafael Guzman Cabrera	Tanja Magoc
Hartmut Haehnel	Luis Ernesto Mancilla
Jin-Kao Hao	Claudia Manfredi
Yasunari Harada	J. Raymundo Marcial-Romero
Pitoyo Hartono	Antonio Marin Hernandez
Rogelio Hasimoto	Luis Felipe Marin Urias
Jean-Bernard Hayet	Urszula Markowska-Kaczmar
Donato Hernandez Fusilier	Ricardo Martinez
Oscar Herrera	Edgar Martinez-Garcia
Ignacio Herrera Aguilar	Jerzy Martyna
Joel Huegel	Oscar Mayora
Michael Huhns	Gordon McCalla
Dieter Hutter	Patricia Melin
Pablo H. Ibarguengoytia	Luis Mena
Mario Alberto Ibarra-Manzano	Carlos Merida-Campos
Héctor Jiménez Salazar	Efrén Mezura-Montes
Moa Johansson	Gabriela Minetti
W. Lewis Johnson	Tanja Mitrovic
Leo Joskowicz	Dieter Mitsche
Chia-Feng Juang	Maria-Carolina Monard
Hiroharu Kawanaka	Luís Moniz Pereira
Shubhalaxmi Kher	Raul Monroy
Ryszard Klempous	Fernando Martin Montes-Gonzalez
Mario Koeppen	Manuel Montes-y-Gómez
Vladik Kreinovich	Oscar Montiel
Sergei Kuznetsov	Jaime Mora-Vargas
Jean-Marc Labat	Eduardo Morales
Susanne Lajoie	Guillermo Morales-Luna
Ricardo Landa Becerra	Enrique Munoz de Cote
H. Chad Lane	Angel E. Munoz Zavala
Reinhard Langmann	Angelica Munoz-Melendez
Bruno Lara	Masaki Murata
Yulia Ledeneva	Rafael Murrieta
Ronald Leder	Tomoharu Nakashima

Atul Negi	Andriy Sadovnychyy
Juan Carlos Nieves	Carolina Salto
Sergey Nikolenko	Gildardo Sanchez
Juan Arturo Nolazco Flores	Guillermo Sanchez
Paulo Novais	Eric Sanjuan
Leszek Nowak	Jose Santos
Alberto Ochoa O. Zezzatti	Nikolay Semenov
Iván Olier	Pinar Senkul
Ivan Olmos	Roberto Sepulveda
Constantin Orasan	Leonid Sheremetov
Fernando Orduña Cabrera	Grigori Sidorov (Chair)
Felipe Orihuela-Espina	Gerardo Sierra
Daniel Ortiz-Arroyo	Lia Susana Silva-López
Mauricio Osorio	Akin Sisbot
Elvia Palacios	Aureli Soria Frisch
David Pearce	Peter Sosnin
Ted Pedersen	Humberto Sossa Azuela
Yoseba Penya	Luis Enrique Sucar
Thierry Peynot	Sarina Sulaiman
Luis Pineda	Abraham Sánchez
David Pinto	Javier Tejada
Jan Platos	Miguel Torres Cisneros
Silvia Poles	Juan-Manuel Torres-Moreno
Eunice E. Ponce-de-Leon	Leonardo Trujillo Reyes
Volodimir Ponomaryov	Alexander Tulupiyev
Edgar Alfredo Portilla-Flores	Fevrier Valdez
Zinovi Rabinovich	Berend Jan Van Der Zwaag
Jorge Adolfo Ramirez Uresti	Genoveva Vargas-Solar
Alonso Ramirez-Manzanares	Maria Vargas-Vera
Jose de Jesus Rangel Magdaleno	Wamberto Vasconcelos
Francisco Reinaldo	Francois Vialatte
Carolina Reta	Javier Vigueras
Carlos A Reyes-Garcia	Manuel Vilares Ferro
María Cristina Riff	Andrea Villagra
Homero Vladimir Rios	Miguel Gabriel Villarreal-Cervantes
Arles Rodriguez	Toby Walsh
Horacio Rodriguez	Zhanshan Wang
Marcela Rodriguez	Beverly Park Woolf
Katia Rodriguez Vazquez	Michal Wozniak
Paolo Rosso	Nadezhda Yarushkina
Jianhua Ruan	Ramon Zatarain
Imre J. Rudas	Laura Zavala
Jose Ruiz Pinales	Qiangfu Zhao
Leszek Rutkowski	

Additional Reviewers

Aboura, Khalid
Acosta-Guadarrama, Juan-Carlos
Aguilar Leal, Omar Alejandro
Aguilar, Ruth
Arce-Santana, Edgar
Bankevich, Anton
Baroni, Pietro
Bhaskar, Pinaki
Bolshakov, Igor
Braga, Igor
Cerda-Villafana, Gustavo
Chaczko, Zenon
Chakraborty, Susmita
Chavez-Echeagaray, Maria-Elena
Cintra, Marcos
Confalonieri, Roberto
Darriba, Victor
Das, Amitava
Das, Dipankar
Diaz, Elva
Ezin, Eugene C.
Figueroa, Ivan
Fitch, Robert
Flores, Marisol
Gallardo-Hernández, Ana Gabriela
Garcia, Ariel
Giacomin, Massimiliano
Ibarra Esquer, Jorge Eduardo
Joskowicz, Leo
Juárez, Antonio
Kawanaka, Hiroharu
Kolesnikova, Olga
Ledeneva, Yulia
Li, Hongliang
Lopez-Juarez, Ismael
Montes Gonzalez, Fernando
Murrieta, Rafael
Navarro-Perez, Juan-Antonio
Nikodem, Jan
Nurk, Sergey
Ochoa, Carlos Alberto
Orozco, Eber
Pakray, Partha
Pele, Ofir
Peynot, Thierry
Piccoli, María Fabiana
Ponomareva, Natalia
Pontelli, Enrico
Ribadas Pena, Francisco Jose
Rodriguez Vazquez, Katya
Sánchez López, Abraham
Sirotkin, Alexander
Suárez-Araujo, Carmen Paz
Villatoro-Tello, Esaú
Wang, Ding
Yaniv, Ziv
Zepeda, Claudia

Organizing Committee

Local Chair
Logistics Staff
Promotion Staff
Sponsors Staff
Administrative Staff
Registration Staff
David Pinto Avendaño
Maya Carrillo
Lourdes Sandoval
Ivan Olmos, Mario Anzures, Fernando Zacarías
Marcos González and Roberto Contreras
Arturo Olvera

Table of Contents – Part I

Automated Reasoning and Multi-Agent Systems

Case Studies on Invariant Generation Using a Saturation Theorem Prover	1
<i>Kryštof Hoder, Laura Kovács, and Andrei Voronkov</i>	
Characterization of Argumentation Semantics in Terms of the MM^r Semantics	16
<i>Mauricio Osorio, José Luis Carballido, Claudia Zepeda, and Zenaida Cruz</i>	
Learning Probabilistic Description Logics: A Framework and Algorithms	28
<i>José Eduardo Ochoa-Luna, Kate Revoredo, and Fábio Gagliardi Cozman</i>	
Belief Merging Using Normal Forms	40
<i>Pilar Pozos-Parra, Laurent Perrussel, and Jean Marc Thevenin</i>	
Toward Justifying Actions with Logically and Socially Acceptable Reasons	52
<i>Hiroyuki Kido and Katsumi Nitta</i>	
A Complex Social System Simulation Using Type-2 Fuzzy Logic and Multiagent System	65
<i>Dora-Luz Flores, Manuel Castañón-Puga, and Carelia Gaxiola-Pacheco</i>	
Computing Mobile Agent Routes with Node-Wise Constraints in Distributed Communication Systems	76
<i>Amir Elalouf, Eugene Levner, and T.C. Edwin Cheng</i>	
Collaborative Redundant Agents: Modeling the Dependences in the Diversity of the Agents' Errors	88
<i>Laura Zavala, Michael Huhns, and Angélica García-Vega</i>	
Strategy Patterns Prediction Model (SPPM)	101
<i>Aram B. González and Jorge A. Ramírez Uresti</i>	
Fuzzy Case-Based Reasoning for Managing Strategic and Tactical Reasoning in StarCraft	113
<i>Pedro Cadena and Leonardo Garrido</i>	

Problem Solving and Machine Learning

Variable and Value Ordering Decision Matrix Hyper-heuristics: A Local Improvement Approach	125
<i>José Carlos Ortiz-Bayliss, Hugo Terashima-Marín, Ender Özcan, Andrew J. Parkes, and Santiago Enrique Conant-Pablos</i>	
Improving the Performance of Heuristic Algorithms Based on Causal Inference	137
<i>Marcela Quiroz Castellanos, Laura Cruz Reyes, José Torres-Jiménez, Claudia Gómez Santillán, Mario César López Locés, Jesús Eduardo Carrillo Ibarra, and Guadalupe Castilla Valdez</i>	
Fuzzified Tree Search in Real Domain Games	149
<i>Dmitrijs Rutko</i>	
On Generating Templates for Hypothesis in Inductive Logic Programming	162
<i>Andrej Chovanec and Roman Barták</i>	
Towards Building a Masquerade Detection Method Based on User File System Navigation	174
<i>Benito Camiña, Raúl Monroy, Luis A. Trejo, and Erika Sánchez</i>	
A Fast SVM Training Algorithm Based on a Decision Tree Data Filter	187
<i>Jair Cervantes, Asdrúbal López, Farid García, and Adrián Trueba</i>	
Optimal Shortening of Covering Arrays	198
<i>Oscar Carrizales-Turrubiates, Nelson Rangel-Valdez, and José Torres-Jiménez</i>	
An Exact Approach to Maximize the Number of Wild Cards in a Covering Array	210
<i>Loreto Gonzalez-Hernandez, José Torres-Jiménez, and Nelson Rangel-Valdez</i>	
Intelligent Learning System Based on SCORM Learning Objects	222
<i>Liliana Argotte, Julieta Noguez, and Gustavo Arroyo</i>	

Natural Language Processing

A Weighted Profile Intersection Measure for Profile-Based Authorship Attribution	232
<i>Hugo Jair Escalante, Manuel Montes y Gómez, and Thamar Solorio</i>	
A New General Grammar Formalism for Parsing	244
<i>Gabriel Infante-Lopez and Martín Ariel Domínguez</i>	

Contextual Semantic Processing for a Spanish Dialogue System Using Markov Logic	258
<i>Aldo Fabian, Manuel Hernandez, Luis Pineda, and Ivan Meza</i>	
A Statistics-Based Semantic Textual Entailment System	267
<i>Partha Pakray, Utsab Barman, Sivaji Bandyopadhyay, and Alexander Gelbukh</i>	
Semantic Model for Improving the Performance of Natural Language Interfaces to Databases	277
<i>Rofolfo A. Pazos R., Juan J. González B., and Marco A. Aguirre L.</i>	
Modular Natural Language Processing Using Declarative Attribute Grammars	291
<i>Rahmatullah Hafiz and Richard A. Frost</i>	
EM Clustering Algorithm for Automatic Text Summarization	305
<i>Yulia Ledeneva, René García Hernández, Romyna Montiel Soto, Rafael Cruz Reyes, and Alexander Gelbukh</i>	
Discourse Segmentation for Sentence Compression	316
<i>Alejandro Molina, Juan-Manuel Torres-Moreno, Eric SanJuan, Iria da Cunha, Gerardo Sierra, and Patricia Velázquez-Morales</i>	
Heuristic Algorithm for Extraction of Facts Using Relational Model and Syntactic Data	328
<i>Grigori Sidorov, Juve Andrea Herrera-de-la-Cruz, Sofía N. Galicia-Haro, Juan Pablo Posadas-Durán, and Liliana Chanona-Hernandez</i>	
MFSRank: An Unsupervised Method to Extract Keyphrases Using Semantic Information	338
<i>Roque Enrique López, Dennis Barreda, Javier Tejada, and Ernesto Cuadros</i>	
Content Determination through Planning for Flexible Game Tutorials	345
<i>Luciana Benotti and Nicolás Bertoá</i>	
Instance Selection in Text Classification Using the Silhouette Coefficient Measure	357
<i>Debangana Dey, Thamar Solorio, Manuel Montes y Gómez, and Hugo Jair Escalante</i>	
Age-Related Temporal Phrases in Spanish and French	370
<i>Sofía N. Galicia-Haro and Alexander Gelbukh</i>	

XVIII Table of Contents – Part I

Sentiment Analysis of Urdu Language: Handling Phrase-Level Negation	382
---	-----

Afraz Zahra Syed, Muhammad Aslam, and Ana Maria Martinez-Enriquez

Unsupervised Identification of Persian Compound Verbs	394
---	-----

Mohammad Sadegh Rasooli, Heshaam Faili, and Behrouz Minaei-Bidgoli

Robotics, Planning and Scheduling

Testing a Theory of Perceptual Mapping Using Robots	407
---	-----

Md. Zulfikar Hossain, Wai Yeap, and Olaf Diegel

A POMDP Model for Guiding Taxi Cruising in a Congested Urban City	415
---	-----

Lucas Agussurja and Hoong Chuin Lau

Next-Best-View Planning for 3D Object Reconstruction under Positioning Error	429
--	-----

Juan Irving Vásquez and L. Enrique Sucar

Stochastic Learning Automata for Self-coordination in Heterogeneous Multi-Tasks Selection in Multi-Robot Systems	443
--	-----

Yadira Quiñonez, Dario Maravall, and Javier de Lope

Stochastic Abstract Policies for Knowledge Transfer in Robotic Navigation Tasks	454
---	-----

Tiago Matos, Yannick Plaino Bergamo, Valdinei Freire da Silva, and Anna Helena Reali Costa

The Evolution of Signal Communication for the e-puck Robot	466
--	-----

Fernando Montes-Gonzalez and Fernando Aldana-Franco

An Hybrid Expert Model to Support Tutoring Services in Robotic Arm Manipulations	478
--	-----

Philippe Fournier-Viger, Roger Nkambou, André Mayers, Engelbert Mephu Nguifo, and Usef Faghihi

Inverse Kinematics Solution for Robotic Manipulators Using a CUDA-Based Parallel Genetic Algorithm	490
--	-----

Omar Alejandro Aguilar and Joel Carlos Huegel

Medical Applications of Artificial Intelligence

MFCA: Matched Filters with Cellular Automata for Retinal Vessel Detection	504
---	-----

Oscar Dalmau and Teresa Alarcon

Computer Assisted Diagnosis of Microcalcifications in Mammograms: A Scale-Space Approach	515
<i>Alberto Pastrana Palma, Juan Francisco Reyes Muñoz, Luis Rodrigo Valencia Pérez, Juan Manuel Peña Aguilar, and Alberto Lamadrid Álvarez</i>	
Diagnosis in Sonogram of Gall Bladder	524
<i>Saad Tanveer, Omer Jamshaid, Abdul Mannan, Muhammad Aslam, Ana Maria Martinez-Enriquez, Afraz Zahra Syed, and Gonzalo Escalada-Imaz</i>	
Genetic Selection of Fuzzy Model for Acute Leukemia Classification	537
<i>Alejandro Rosales-Pérez, Carlos A. Reyes-García, Pilar Gómez-Gil, Jesus A. Gonzalez, and Leopoldo Altamirano</i>	
An Ontology for Computer-Based Decision Support in Rehabilitation	549
<i>Laia Subirats and Luigi Ceccaroni</i>	
Heuristic Search of Cut-Off Points for Clinical Parameters: Defining the Limits of Obesity	560
<i>Miguel Murguía-Romero, Rafael Villalobos-Molina, René Méndez-Cruz, and Rafael Jiménez-Flores</i>	
Development of a System of Electrodes for Reading Consents-Activity of an Amputated Leg (above the knee) and Its Prosthesis Application	572
<i>Emilio Soto, Jorge Antonio Ascencio, Manuel Gonzalez, and Jorge Arturo Hernandez</i>	
Predicting the Behavior of the Interaction of Acetylthiocholine, pH and Temperature of an Acetylcholinesterase Sensor	583
<i>Edwin R. García, Larysa Burtseva, Margarita Stoytcheva, and Félix F. González</i>	
Author Index	593

Table of Contents – Part II

Fuzzy Logic, Uncertainty and Probabilistic Reasoning

Intelligent Control of Nonlinear Dynamic Plants Using a Hierarchical Modular Approach and Type-2 Fuzzy Logic	1
<i>Leticia Cervantes, Oscar Castillo, and Patricia Melin</i>	
No-Free-Lunch Result for Interval and Fuzzy Computing: When Bounds Are Unusually Good, Their Computation Is Unusually Slow	13
<i>Martine Ceberio and Vladik Kreinovich</i>	
Intelligent Robust Control of Dynamic Systems with Partial Unstable Generalized Coordinates Based on Quantum Fuzzy Inference	24
<i>Andrey Mishin and Sergey Ulyanov</i>	
Type-2 Neuro-Fuzzy Modeling for a Batch Biotechnological Process	37
<i>Pablo Hernández Torres, María Angélica Espejel Rivera, Luis Enrique Ramos Velasco, Julio Cesar Ramos Fernández, and Julio Waissman Vilanova</i>	
Assessment of Uncertainty in the Projective Tree Test Using an ANFIS Learning Approach.....	46
<i>Luis G. Martínez, Juan R. Castro, Guillermo Licea, and Antonio Rodríguez-Díaz</i>	
ACO-Tuning of a Fuzzy Controller for the Ball and Beam Problem.....	58
<i>Enrique Naredo and Oscar Castillo</i>	
Estimating Probability of Failure of a Complex System Based on Inexact Information about Subsystems and Components, with Potential Applications to Aircraft Maintenance	70
<i>Vladik Kreinovich, Christelle Jacob, Didier Dubois, Janette Cardoso, Martine Ceberio, and Ildar Batyrshin</i>	
Two Steps Individuals Travel Behavior Modeling through Fuzzy Cognitive Maps Pre-definition and Learning	82
<i>Maikel León, Gonzalo Nápoles, María M. García, Rafael Bello, and Koen Vanhoof</i>	
Evaluating Probabilistic Models Learned from Data	95
<i>Pablo H. Ibargüengoytia, Miguel A. Delgadillo, and Uriel A. García</i>	

Evolutionary Algorithms and Other Naturally-Inspired Algorithms

A Mutation-Selection Algorithm for the Problem of Minimum Brauer Chains	107
<i>Arturo Rodriguez-Cristerna, José Torres-Jiménez, Iván Rivera-Islas, Cindy G. Hernandez-Morales, Hillel Romero-Monsivais, and Adan Jose-Garcia</i>	
Hyperheuristic for the Parameter Tuning of a Bio-Inspired Algorithm of Query Routing in P2P Networks	119
<i>Paula Hernández, Claudia Gómez, Laura Cruz, Alberto Ochoa, Norberto Castillo, and Gilberto Rivera</i>	
Bio-Inspired Optimization Methods for Minimization of Complex Mathematical Functions	131
<i>Fevrier Valdez, Patricia Melin, and Oscar Castillo</i>	
Fundamental Features of Metabolic Computing	143
<i>Ralf Hofestädt</i>	
Clustering Ensemble Framework via Ant Colony	153
<i>Hamid Parvin and Akram Beigi</i>	
Global Optimization with the Gaussian Polytree EDA	165
<i>Ignacio Segovia Domínguez, Arturo Hernández Aguirre, and Enrique Villa Diharce</i>	
Comparative Study of BSO and GA for the Optimizing Energy in Ambient Intelligence	177
<i>Wendoly J. Gpe. Romero-Rodríguez, Victor Manuel Zamudio Rodríguez, Rosario Baltazar Flores, Marco Aurelio Sotelo-Figueroa, and Jorge Alberto Soria Alcaraz</i>	
Modeling Prey-Predator Dynamics via Particle Swarm Optimization and Cellular Automata	189
<i>Mario Martínez-Molina, Marco A. Moreno-Armendáriz, Nareli Cruz-Cortés, and Juan Carlos Seck Tuoh Mora</i>	

Data Mining

Topic Mining Based on Graph Local Clustering	201
<i>Sara Elena Garza Villarreal and Ramón F. Brena</i>	
SC Spectra: A Linear-Time Soft Cardinality Approximation for Text Comparison	213
<i>Sergio Jiménez Vargas and Alexander Gelbukh</i>	

Times Series Discretization Using Evolutionary Programming.....	225
<i>Fernando Rechy-Ramírez, Héctor-Gabriel Acosta Mesa, Efrén Mezura-Montes, and Nicandro Cruz-Ramírez</i>	
Clustering of Heterogeneously Typed Data with Soft Computing – A Case Study	235
<i>Angel Kuri-Morales, Daniel Trejo-Baños, and Luis Enrique Cortes-Berrueco</i>	
Regional Flood Frequency Estimation for the Mexican Mixteca Region by Clustering Techniques	249
<i>Felix Emilio Luis-Pérez, Raúl Cruz-Barbosa, and Gabriela Álvarez-Olguín</i>	
Border Samples Detection for Data Mining Applications Using Non Convex Hulls	261
<i>Asdrúbal López Chau, Xiaoou Li, Wen Yu, Jair Cervantes, and Pedro Mejía-Álvarez</i>	
An Active System for Dynamic Vertical Partitioning of Relational Databases.....	273
<i>Lisbeth Rodríguez, Xiaoou Li, and Pedro Mejía-Álvarez</i>	
Efficiency Analysis in Content Based Image Retrieval Using RDF Annotations	285
<i>Carlos Alvez and Aldo Vecchietti</i>	
Automatic Identification of Web Query Interfaces	297
<i>Heidy M. Marin-Castro, Victor J. Sosa-Sosa, and Ivan Lopez-Arevalo</i>	
Neural Networks and Hybrid Intelligent Systems	
A GRASP with Strategic Oscillation for a Commercial Territory Design Problem with a Routing Budget Constraint	307
<i>Roger Z. Ríos-Mercado and Juan C. Salazar-Acosta</i>	
Hybrid Intelligent Speed Control of Induction Machines Using Direct Torque Control	319
<i>Fernando David Ramirez Figueroa and Alfredo Victor Mantilla Caeiros</i>	
A New Model of Modular Neural Networks with Fuzzy Granularity for Pattern Recognition and Its Optimization with Hierarchical Genetic Algorithms	331
<i>Daniela Sánchez, Patricia Melin, and Oscar Castillo</i>	
Crawling to Improve Multimodal Emotion Detection	343
<i>Diego R. Cueva, Rafael A.M. Gonçalves, Fábio Gagliardi Cozman, and Marcos R. Pereira-Barretto</i>	

Improving the MLP Learning by Using a Method to Calculate the Initial Weights of the Network Based on the Quality of Similarity Measure	351
<i>Yaima Filiberto Cabrera, Rafael Bello Pérez, Yailé Caballero Mota, and Gonzalo Ramos Jimenez</i>	
Modular Neural Networks with Type-2 Fuzzy Integration for Pattern Recognition of Iris Biometric Measure	363
<i>Fernando Gaxiola, Patricia Melin, Fevrier Valdez, and Oscar Castillo</i>	
Wavelet Neural Network Algorithms with Applications in Approximation Signals.....	374
<i>Carlos Roberto Domínguez Mayorga, María Angélica Espejel Rivera, Luis Enrique Ramos Velasco, Julio Cesar Ramos Fernández, and Enrique Escamilla Hernández</i>	
Computer Vision and Image Processing	
Similar Image Recognition Inspired by Visual Cortex	386
<i>Urszula Markowska-Kaczmar and Adam Puchalski</i>	
Regularization with Adaptive Neighborhood Condition for Image Denoising	398
<i>Felix Calderon and Carlos A. Júnez-Ferreira</i>	
Multiple Target Tracking with Motion Priors	407
<i>Francisco Madrigal, Mariano Rivera, and Jean-Bernard Hayet</i>	
Control of a Service Robot Using the Mexican Sign Language	419
<i>Felix Emilio Luis-Pérez, Felipe Trujillo-Romero, and Wilebaldo Martínez-Velazco</i>	
Analysis of Human Skin Hyper-spectral Images by Non-negative Matrix Factorization	431
<i>July Galeano, Romuald Jolivot, and Franck Marzani</i>	
Similarity Metric Behavior for Image Retrieval Modeling in the Context of Spline Radial Basis Function	443
<i>Leticia Flores-Pulido, Oleg Starostenko, Gustavo Rodríguez-Gómez, Alberto Portilla-Flores, Marva Angelica Mora-Lumbreras, Francisco Javier Albores-Velasco, Marlon Luna Sánchez, and Patrick Hernández Cuamatzi</i>	
A Comparative Review of Two-Pass Connected Component Labeling Algorithms	452
<i>Uriel H. Hernandez-Belmonte, Victor Ayala-Ramirez, and Raul E. Sanchez-Yanez</i>	

A Modification of the Mumford-Shah Functional for Segmentation of Digital Images with Fractal Objects	463
<i>Carlos Guillén Galván, Daniel Valdés Amaro, and Jesus Uriarte Adrián</i>	
Robust RML Estimator - Fuzzy C-Means Clustering Algorithms for Noisy Image Segmentation	474
<i>Dante Mújica-Vargas, Francisco Javier Gallegos-Funes, Alberto J. Rosales-Silva, and Rene Cruz-Santiago</i>	
Processing and Classification of Multichannel Remote Sensing Data	487
<i>Vladimir Lukin, Nikolay Ponomarenko, Andrey Kurekin, and Oleksiy Pogrebnyak</i>	
Iris Image Evaluation for Non-cooperative Biometric Iris Recognition System	499
<i>Juan M. Colores, Mireya García-Vázquez, Alejandro Ramírez-Acosta, and Héctor Pérez-Meana</i>	
Optimization of Parameterized Compactly Supported Orthogonal Wavelets for Data Compression	510
<i>Oscar Herrera Alcántara and Miguel González Mendoza</i>	
Efficient Pattern Recalling Using a Non Iterative Hopfield Associative Memory	522
<i>José Juan Carbajal Hernández and Luis Pastor Sánchez Fernández</i>	
Author Index	531

Case Studies on Invariant Generation Using a Saturation Theorem Prover^{*}

Kryštof Hoder¹, Laura Kovács², and Andrei Voronkov¹

¹ University of Manchester

² TU Vienna

Abstract. Automatic understanding of the intended meaning of computer programs is a very hard problem, requiring intelligence and reasoning. In this paper we evaluate a program analysis method, called symbol elimination, that uses first-order theorem proving techniques to automatically discover non-trivial program properties. We discuss implementation details of the method, present experimental results, and discuss the relation of the program properties obtained by our implementation and the intended meaning of the programs used in the experiments.

1 Introduction

The complexity of computer systems grows exponentially. Many companies and organisations are now routinely dealing with software comprising several millions lines of code, written by different people using different languages, tools and styles. This software is hard to understand and is integrated in an ever changing complex environment, using computers, networking, various physical devices, security protocols and many other components. Ensuring the reliability of such systems for safety-critical applications is extremely difficult. One way of solving the problem is to analyse or verify these systems using computer-aided tools based on computational logic.

In [9] a new method, called *symbol elimination*, has been proposed to automatically generate statements expressing computer program properties. The approach requires no preliminary knowledge about program behavior, but uses the power of a saturation theorem prover to derive and understand the *intended meaning* of the program. To undertake such a complex task, reasoning in the combination of first order logic and various theories is required as program components involve both bounded and unbounded data structures.

One can argue that automatic inference of program properties is a hard and creative problem whose solution improves our understanding of the relation between the computer reasoning and the human reasoning. Indeed, given a computer program, one can ask a computer programmer questions like “what are the essential properties of this program” or “what is the intended meaning of this program?” Answering such question requires intelligence. If the program is small and not highly sophisticated, one can

* This research was partly supported by Dassault Aviation. Kryštof Hoder is supported by the School of Computer Science at the University of Manchester and by an EPSRC Pathfinder grant. Laura Kovács is supported by an FWF Hertha Firnberg Research grant (T425-N23). Andrei Voronkov is partially supported by an EPSRC grant. This research is partly supported by the FWF National Research Network RiSE (S11410-N23).

expect that the programmer will be able to give some answers. For example, if the program copies one array into another, one can expect the programmer to say that the intended meaning of the program is to copy arrays and that among the most essential properties of this program are the facts that the two arrays will be equal and the first array will not be modified. These two properties are first-order properties, that is, they can be expressed in first-order logic.

The properties generated by the symbol elimination method are first-order properties, therefore one can ask a question of whether a computer program can generate such properties and whether it can generate “the intended” properties (whatever it means). This paper tries to answer this fundamental question by taking a program annotated by humans, removing these annotations, generating program properties by a computer program, and comparing the generated program properties with the intended properties.

The first implementation of symbol elimination was carried out in the first-order resolution theorem prover Vampire [7]. This implementation could be used for symbol elimination, yet not for invariant generation. It was run on an array partitioning program to demonstrate that it can generate complex loop invariants with alternating quantifiers, which could not be generated by any other existing technique. Such complex loop properties precisely capture the intended first-order meaning of the program, as well as the programmer’s intention while writing the program.

However, the practical power of symbol elimination was not clear since it required extensive experiments on programs containing loops. Such experiments were not easy to organise since they involved combining several tools for program analysis and theorem proving. Designing such a combination turned out to be non-trivial and error-prone. The first standalone implementation of program analysis in the theorem prover Vampire is described in the system abstract [8]. Experimental evaluation of the method was however not yet carried out.

This paper undertakes the first extensive investigation into understanding the power and limitations of symbol elimination for invariant generation. In addition to the fundamental AI problems mentioned above, we were also interested in the power of the method for applications in program analysis, which can be measured by the following characteristics:

1. [Strength] Is the method powerful enough to infer automatically invariants that would imply intended loop properties?
2. [Time] Is invariant generation fast enough?
3. [Quantity] What is the number of generated invariants?

The method we use to answer these questions in this work is described below.

- We use the invariant generation framework of [8] implemented directly in Vampire. Vampire can now accept as an input a program written in a subset of C, find all loops in it, and generate and output invariants for each of the loops. To this end, to make [8] practically useful, we extended the program analyser of [8] with a C parser.
- We took a number of annotated C programs coming from an industrial software verification project, as well as several standard benchmarks circulated in the verification community (for example, [2,13]), and removed annotations corresponding to loop invariants.

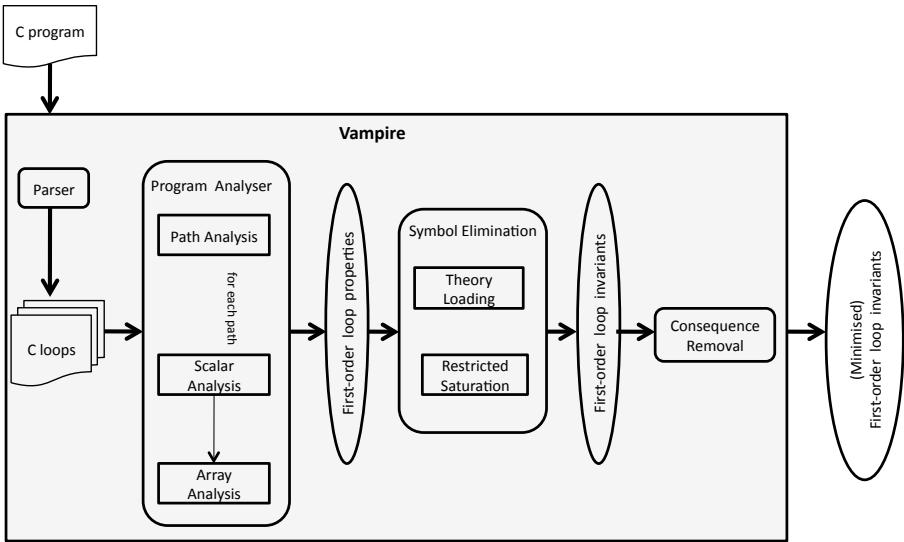


Fig. 1. Invariant Generation in Vampire

- We ran Vampire with various time limits and collected statistics relevant to the questions raised above.

The main contribution of this paper is an experimental evaluation of the symbol elimination approach, providing statistics and better understanding of the power of the method.

Our experiments show that, at least for small (but far from trivial) programs a computer program using the symbol elimination method turns out to generate in less than a second properties that imply the annotated properties, that is, the intended properties. Hence, symbol elimination confirms, in some way, the power of deductive methods in computer reasoning. However, even for relatively simple programs it also generates many other properties, which shows that further investigation may be required to bring the human and the computer understanding of the meaning of programs closer.

This paper is structured as follows. Section 3.1 overviews the program analysis framework of Vampire. Given a program in a subset of C as an input, Vampire now *automatically* generates a set of invariants for the loops occurring in the program.

Reasoning about programs requires reasoning in combined first-order theories, such as arithmetic, uninterpreted functions, arrays, etc. Since invariant generation by symbol elimination requires both theories and quantifiers, efficient handling of theories in Vampire was a major issue for us. We describe theory reasoning in Vampire in Section 3.2.

Since loop invariants are to be used in proving program properties, it is important that the generated set of invariants is not too large, yet powerful enough to imply important program properties. A framework for removing invariants implied by other invariants was introduced in [8], where invariants are removed in combination with invariant generation. In our work we use [8], as described in Sections 3.3 and 3.4. We further give

an experimental evidence that such a removal is in practice “cheap” as compared to invariant generation (Section 4).

The “quality” of a set of automatically generated invariants refers to whether it can be used to easily derive program properties required for program verification or static analysis. We discuss the quality of invariants generated by our technique in Section 3.5.

The key section of this paper is Section 4 on experimental evaluation of the symbol elimination method. The reported experimental results provide empirical evidence of the power of symbol elimination for generating complex invariant.

We briefly consider related work in Section 5 and draw conclusions in Section 6.

2 Preliminaries

Programs, Variables and Expressions. Figure 1 illustrates our approach to practical invariant generation in Vampire. Figure 1 extends [8] by the use of a program parser. Hence, Vampire now can handle C programs over integers and arrays using assignments, loops and if-then-else conditional statements with standard (C language) semantics. Non-integer variables can also be considered but regarded as uninterpreted.

We assume that the language expressions contain integer constants, variables, and some function and predicate symbols. The standard arithmetical functions, such as $+$, $-$, \cdot are considered as interpreted, while all other function symbols are uninterpreted. Likewise, the arithmetical predicate symbols $=$, \neq , \leq , \geq , $<$ and $>$ are interpreted while all other predicate symbols are uninterpreted. As usual, the expression $A[e]$ is used to denote the element of an array A at the position given by the expression e . The current version of Vampire does not handle nested loops. If the input program contains such loops, only the innermost loops will be analysed.

The programming model for invariant generation in Vampire is summarized below.

while b do $s_1; s_2; \dots; s_m$ end do (1)

where b is a boolean expression, and statements s_i ($i = 1, \dots, m$) are either assignments or (nested) if-then-else conditional statements. Some example loops for invariant generation in Vampire are given in the leftmost column of Table 3. Throughout this paper, integer program variables are denoted by lower-case letters a, b, c, \dots , whereas array variables are denoted by upper-case letters A, B, C, \dots .

Theorem proving and Vampire. The symbol elimination method described and investigated in this paper is essentially based on the use of a first-order theorem prover. Moreover, one needs a theorem prover able to deal with theories (e.g. integer arithmetic), first-order logic, and generate consequences. Theorem provers using saturation algorithms are ideal for consequence generation. Saturation actually means that the theorem prover tries to generate all, in some sense, consequences of a given set of formulas in some inference system, for example, resolution and superposition [12].

In reality, saturation theorem provers use a powerful concept of *redundancy elimination*. Redundancy elimination is not an obstacle to consequence generation, since redundant formulas are logical consequences of other formulas the theorem prover is dealing with.

All our experiments described in this paper are conducted using Vampire [11], which is a resolution and superposition theorem prover running saturation algorithms. Vampire is available from <http://www.vprover.org>.

3 Symbol Elimination and Invariant Generation in Vampire

In a nutshell, the symbol elimination method of [9] works as follows. One is given a loop L , which may contain both scalar and array variables.

(1) In the *first phase*, symbol elimination tries to *generate as many loop properties as possible*. This may sound as solving a hard problem using an even harder problem, yet the method undertakes an easy way. First, it considers all (scalar and array) variables of L as functions of the *loop counter* n . This means that for every scalar variable a , a function $a(n)$ denoting the value of a at the loop iteration n is introduced. Likewise, for every array variable A a function $A(n, p)$ is introduced, denoting the value of A at the iteration n in the position (or index) p . Thus, the language of loop is extended by new function symbols, obtaining a new, extended language. Note that some loop properties in the new language are easy to extract from the loop body, for example, one can easily write down a formula describing the values of all loop variables at the iteration $n + 1$ in terms of their values at an iteration n , by using the transition relation of the loop. In addition to the transition relation, some properties of *counters* (scalar variables that are only incremented or decremented by constant values in L) are also added. Further, the loop language is also extended by the so-called *update predicates for arrays* and their properties are added to the extended language. An update predicate for an array A essentially expresses updates made to A and their effect on the final value of A . After this step, a collection Π of valid loop properties expressed in the extended language is derived.

(2) Formulas in Π cannot be used as loop invariants, since they use symbols not occurring in the loop, and even symbols whose semantics is described by the loop itself. These formulas, being valid properties of L , have a useful property: all their consequences are valid loop properties too. The *second phase* of symbol elimination tries to *generate logical consequences of Π in the original language of the loop*. Any such consequence is also a valid property of L , and hence an invariant of L . Logical consequences of Π are generated by running a saturation theorem prover on Π in a way that the theorem prover tries to eliminate the newly introduced symbols.

(3) The *third phase* of the method, added recently to symbol elimination [8], tries to remove invariants implied by other generated invariants.

It is important to note that (the first phase of) symbol elimination can be combined with other methods of program analysis. Indeed, any valid program property can be added to Π , resulting hopefully in a stronger set of invariants.

The rest of this section describes the details of how these three phases are implemented in Vampire.

3.1 Program Analysis in Vampire

In this section we briefly overview the program analysis phase of invariant generation in Vampire, introduced in [8].

```

Loops found: 1
Analyzing loop...
-----
while (a < m)
{
    if (A[a] >= 0)
    {
        B[b] = A[a];
        b = b + 1;
    }
    else
    {
        C[c] = A[a];
        c = c + 1;
    }
    a = a + 1;
}
-----
Variable: B: (updated)
Variable: a: (updated)
Variable: b: (updated)
Variable: m: constant
Variable: A: constant
Variable: C: (updated)
Variable: c: (updated)

```

Counter: a Counter: b Counter: c Path: false: A[a] >= 0 C[c] = A[a]; c = c + 1; a = a + 1; Path: true: A[a] >= 0 B[b] = A[a]; b = b + 1; a = a + 1; Counter a: 1 min, 1 max, 1 gcd Counter b: 0 min, 1 max, 1 gcd Counter c: 0 min, 1 max, 1 gcd	7. ! [X1,X0,X3] : (X1>X0 & c(X1)>X3 & X3>c(X0)) => ?[X2] : (c(X2)=X3 & X2>X0 & X1>X2) [program analysis] 6. ! [X0]:c(X0)>=c0 (0:4) [program analysis] 5. ! [X0]:c(X0)<=c0+X0 (0:6) [program analysis] 4. ! [X1,X0,X3] : (X1>X0 & b(X1)>X3 & X3>b(X0)) => ?[X2] : (b(X2)=X3 & X2>X0 & X1>X2) [program analysis] 3. ! [X0]:b(X0)>=b0 (0:4) [program analysis] 2. ! [X0]:b(X0)<=b0+X0 (0:6) [program analysis] 1. ! [X0]:a(X0)=a0+X0 (0:6) [program analysis]
---	--

Fig. 2. Partial output of Vampire's program analysis on the Partition program of Table 3.

The analyser works with simple programs of the form described in Section 2, and generates loop properties for the first phase of symbol elimination. The analyser works as follows. First, it extracts from the program all non-nested loops and performs the following steps on every such loop.

1. Find all loop variables and classify them into variables updated by the loop and constant variables.
2. Find counters, that is, updated scalar variables that are only incremented or decremented by constant values. Save properties of counters.
3. Generate update predicates of updated array variables and save their properties.
4. Save the formulas corresponding to the transition relation of the loop.
5. Create a symbol elimination task for Vampire by putting together all saved formulas and marking symbols that have to be eliminated.

Example 1. Figure 2 shows the output of Vampire corresponding to the first four steps of the program analysing process for the Partition program of Table 3. In the output, $! [X]$ (respectively $?[X]$) denotes $\forall X$ (respectively, $\exists X$). The full example contains, apart from the loop shown in Figure 2, initialisation of some loop variables.

As shown in Figure 2, the program analyser of Vampire detects that variables B, A, a, b, c are updated in the loop, variables A, m are constants, and the variables a, b, c are counters.

Next, the path analysis of the program analyser reports that there are two program paths (listed in two blocks starting with Path), depending on the value of the test $A[a] \geq 0$. The values min and max denote the maximal and the minimal increment of the counter over all paths in the program. The value gcd is the greatest common divisor of all such increments.

Properties of counters are further generated, as listed between the lines enumerated by 1-7 of Figure 2. For example, Vampire generates axiom 1 expressing the following

property: for *every loop iteration X0*, the value $a(X0)$ of a at iteration $X0$ is given by $a0 + X0$, where $a0$ denotes the initial value of a .

3.2 Theory Reasoning in Vampire

Standard resolution and superposition theorem provers are good in dealing with quantifiers but lack any support for theories, such as those of integers, reals, arrays, lists, etc. The standard way of adding a theory to such a theorem prover is by adding a first-order axiomatisation of the theory. There is no complete axiomatisation for all the above mentioned theories (assuming arbitrary quantifiers).

Adding incomplete axiomatisations is the approach used in [9]. The new version of Vampire [8] came further than [9] and also added integers as a data type in Vampire. The method of [8] is also the approach we follow in this paper. This means that integers can be used directly instead of representing them using, for example, zero and the successor function. Vampire “knows” several standard predicates and functions on integers: addition, subtraction, multiplication, successor, division, and standard inequality relations such as \leq . Since Vampire’s users may not know much about combining arithmetic and first-order logic, automatic loading of relevant theories is taken care by Vampire. For example, if the user uses the standard integer addition function symbol $+$, then Vampire will automatically add an axiomatisation of integer linear arithmetic including axioms for $+$.

Generally, for loading existing theory axiomatisations of Vampire, the user should add to the input (in the TPTP syntax) a Vampire-specific declaration binding an input symbol to an interpreted theory symbol. For example, one can write:

```
vampire(interpreted_symbol, geq, integer_greater_equal).
```

to declare that the input symbol geq denotes the inequality \geq on integers. Given the above declaration, Vampire will add some theory axioms for this symbol to the input formulas. The user can also choose to use her own axiomatisation or to add more axioms to the axiomatisation loaded by Vampire.

The results reported in Section 4 show that Vampire’s approach to reasoning with integers is good enough for proving properties of simple loops. However, the research into various approaches to reasoning with quantifiers and theories and their relative strength is still in its infancy and hindered by a lack of publicly available benchmarks.

3.3 Symbol Elimination in Vampire

If one just adds a collection of formulas obtained by program analysis to a theorem prover and expects the prover to generate consequences of these formulas using only a given subset of functions and predicates from the input, the result will most likely be disappointing. For example, suppose that p is a symbol that cannot occur in invariants, while q, r are symbols that can occur in them. Suppose that we are also given two clauses $p \vee q$ and $\neg p \vee r$. A theorem prover may decide to derive the invariant $q \vee r$ from these two clauses but may also decide not to do anything with them, depending on the term ordering and literal selection it uses (e.g. q and r might be selected before p).

The main ingredient of the symbol elimination technique in Vampire is the concept of a *well-colored derivation* [10,7]. To define well-colored derivations (also called local

proofs or split proofs) some predicate and/or function symbols are declared to have colors, while other symbols are uncolored. A symbol, term, literal or formula using a color are called colored, otherwise they are called transparent. A derivation is called *well-colored* if any inference can use symbols of at most one color. Any inference having at least one colored premise and a transparent conclusion is called a *symbol eliminating inference*.

Following the symbol elimination approach of [9], loop invariant generation can be thus be addressed using colors, as follows. One and the same color is assigned to all additional symbols introduced for formulating properties of loops (see Section 3), for example, the loop counter. All other symbols, that is the loop variables and the theory symbols, are transparent. A loop invariant is then a transparent formula describing a valid loop property. Since one is guaranteed that any transparent consequence of the input set of formulas is a valid loop property, the problem of invariant generation reduces to the problem of generating transparent consequences of this set. This means, in a way, that the colored symbols should be eliminated.

To make saturation more effective for deriving transparent consequences, the Knuth-Bendix term ordering used in Vampire was modified in [7], so that symbol weights are infinite ordinals and any colored ground term or atom is greater than any ground transparent term or atom.

Example 2. Consider the Partition program from Table 3. As presented in Figure 2, the program analyser of Vampire generates the following valid loop property:

$$a(X_0) = a_0 + X_0 \quad \text{for every loop iteration } X_0.$$

However, this property cannot be used as an invariant as it makes use of the additional unary function symbol $a(X_0)$ denoting the value of the loop variable a at an iteration X_0 . However, since we declare the unary symbol a colored, Vampire tries to eliminate $a(X_0)$ from the set of valid properties generated by its program analyser, and derives, for example, the following transparent formula by a symbol eliminating inference:¹

$$a - a_0 \geq 0, \quad \text{where the constant } a \text{ denotes the value of } a \text{ at the end of the loop.}$$

This property is a loop invariant, as it uses only the transparent symbols. Similarly,

$$B(X_0, b(X_0)) \geq 0 \quad \text{for every loop iteration } X_0$$

is a valid loop property, but not an invariant. However, by making the unary symbol b and the binary symbol B colored, Vampire generates the following invariant from this and other loop properties:

$$0 \leq X < b \Rightarrow B(X) \geq 0,$$

where the unary symbol B and the constant b are the corresponding transparent symbols denoting the final values of loop variables with the same names.

3.4 Pruning Generated Invariants

Symbol elimination can generate invariants implied by other generated invariants. For example, any inference applied to two invariants gives an invariant. For this reason,

¹ As described in [7], for every program variable v two transparent variables v_0 and v are used, denoting the initial and the final values of this loop variable. Further, a colored unary function symbol v is introduced, such that $v(X)$ denotes the value of the loop variable v at iteration X .

Table 1. Symbol elimination on programs from [2,13], by running Vampire with 1 and 10 seconds of time limit

Loop	Symbol Elimination within 1s				Symbol Elimination within 10s					
	# SEI	# Min SEI	% Redundancy	\forall -Inv	$\forall\exists$ -Inv	# SEI	# Min SEI	% Redundancy	\forall -Inv	$\forall\exists$ -Inv
Initialisation	15	5	67%	yes	no	40	5	88%	yes	no
Copy	24	5	79%	yes	no	25	5	80%	yes	no
Find	151	13	91%	yes	no	474	21	96%	yes	no
Vararg	1	1	0%	yes	no	1	1	0%	yes	no
Partition	166	38	77%	yes	yes	849	59	93%	yes	yes
Partition_Init	168	24	86%	yes	yes	692	127	82%	yes	yes
Shift	41	12	71%	yes	no	111	16	86%	yes	no

[7] only stores invariants obtained by an inference having at least one colored premise, that is, a symbol-eliminating inference. However, even among conclusions of symbol-eliminating inferences there are typically many invariants implied by others.

To improve invariant generation, a new mode, called the *consequence-elimination mode*, was added to Vampire in [8]. In this mode, Vampire obtains a set S of clauses (i.e. invariants) as an input and tries to find its proper subset S' equivalent to S . In the process of computing S' , Vampire is run with a small time limit. Naturally, one is interested in having S' as small as possible.

In the experiments reported in this paper, we made use of [8] and ran Vampire in the consequence elimination mode using four different strategies with a 20 seconds time limit. Our experimental results show that typically between 80% and 90% of all invariants obtained by symbol elimination are redundant, and hence discarded. It is usually the case that all or nearly all of the discarded invariants are discovered in a few milliseconds already by the first strategy.

3.5 Proving Invariants, Postconditions, and Assertions

The set of invariants and loop properties resulting from Vampire's program analysis and symbol elimination can be used to prove loop properties. Note that proving a loop property can be done in at least two different ways.

(a) First, we can add the negation of the property to the formulas obtained by program analysis and try to prove that the resulting set of formulas is unsatisfiable. It is easier than invariant generation since one does not have to take care of colors and can use arbitrary proofs, ordering and strategies, including goal-oriented ones.

(b) Second, one can prove that the property is an inductive invariant, which is a much simpler problem and can be reduced to proving a few formulas with respect to the theory.

However, both approaches assume that every loop is already annotated. Providing such annotations manually requires a considerable amount of work by highly qualified persons and thus often makes verification prohibitively expensive. Therefore, generation of invariants without using any annotations is invaluable in making verification and static analysis of programs economically feasible.

Evaluating the quality of an invariant generation technique is not easy since it cannot be measured using simple measures, such as the number of generated invariants or the speed of their generation. One can say that such a technique is powerful, if the set of

Table 2. Symbol elimination on programs sent by Dassault Aviation

Loop Shape	# of Loops	Average # of SEI	Average # of Non-Redundant SEI	% of SEI Redundancy	\forall -Inv	$\forall\exists$ -Inv
Simple	33	168	18	89.3%	yes	no
Multi-path	5	340	46	86.4%	yes	yes

invariants generated in small time implies the *intended* loop invariants, or invariants that humans would use to annotate this loop for verification purposes.

To evaluate our method, we used annotated code both from academic benchmarks and an industrial verification project. In this code every loop was *annotated by its intended property* in the form of loop invariants and/or postconditions. So we ran Vampire on the code as follows.

1. First, we generated a set of invariants using symbol elimination, as described in Sections 3.1–3.4;
2. Then we checked, also using Vampire, whether the intended loop property is implied by this set of invariants and whether the intended property described a postcondition, if the latter was provided.

Example 3. In the fourth column of Table 3, we show one of the intended invariant of the Partition program. This invariant follows from the two invariants generated by Vampire, which are presented in the fifth column of the table.

4 Experimental Results

The experiments described in this section were carried out using two benchmark suites. One is a collection of 6 loops taken from various research papers (Tables 3 and 1). The other one is a collection of 38 loops taken from programs provided by Dassault Aviation. We used a computer with a 2 GHz Intel Core i7 CPU processor and 4GB RAM, and ran experiments using the Vampire version 0.6. The consequence elimination phase of Vampire was run with a 20 seconds time limit.

To analyse C programs, we had to extend Vampire by a C parser. Inputs to the parser are (large) C programs. After parsing, Vampire finds all loops in the program and checks, for each loop, if it is as given in (1) and thus can be analysed. Vampire outputs a set of loop invariants for each loop (1) under analysis. Figure 1 illustrates the invariant generation process within Vampire.

To use Vampire for generating invariants of arbitrary C code, one should use it in the program analysis mode as follows:

```
vampire --mode program_analysis < filename.c
```

4.1 Challenging Benchmarks

Table 3 describes the effect of symbol elimination on 6 programs. The names of the first 5 programs and their origins (that is, the papers where they were described) are given in column 1. The program given in the last row of Table 3 is taken from our own case studies, and illustrates the possibility of generating invariants for loops using read-and-write arrays. Columns 2 contains the number of invariants generated in 1 second, while column 3 the number of invariants that remain after consequence elimination.

Column 4 contains the intended invariant (or the invariant of interest): for this invariant we checked whether it is implied by invariants generated by Vampire, and if yes, show (in column 5) the subset of the generated invariants that imply the intended one. Checking that the intended invariant follows from the generated invariants was done using Vampire, so clauses in column 5 are those extracted from the corresponding Vampire proof. Invariants are generated by Vampire in a certain order. We used this order to enumerate clauses in column 5, since it gives the reader an idea how fast the required invariants were found. For example, `inv81` for the `Partition` example means that this invariant was the 82nd generated invariant (counting invariants starts from 0). Note that some of the formulas in this column have skolem functions introduced by Vampire's clausifier. For example, sk_1 denotes a skolem function. They can be de-skolemised to give invariants with quantifier alternations.

For this benchmark suite, *all the intended invariants turned out to be logical consequences of the invariants generated by Vampire*. However, one of the intended invariants could not be proved by Vampire. Namely, for the `Shift` example, Vampire generated the invariant $\forall x(x \geq 0 \wedge x < a \Rightarrow A[x] = A[x + 1])$, while the intended invariant was $\forall x(x \geq 0 \wedge x \leq a \Rightarrow A[x] = A[0])$. These two invariants are equivalent in arithmetic, however, to prove the intended one from the generated one in first-order logic one needs induction. By adding a simple induction axiom, specific to the `Shift` example, we could also prove the intended invariant.

We were also interested in checking the number of generated invariants depending on the time spent on invariant generation. Table 1 contains statistics about invariant generation with 1 and 10 seconds time limits on symbol elimination, respectively. It also shows the percentage of generated invariants shown to be redundant. As one can see, on the average over 80% of the generated invariants were proved to be redundant. Moreover, Table 1 reports whether quantified invariants with only universal quantifier (\forall -Inv) and with quantifier alternations ($\forall\exists$ -Inv) have been generated. The relative size of the minimised set varies from example to example. Also, the intended invariant is always implied by the invariants generated in the first second (as reported in Table 3). For example, the intended (and rather complex) invariant for the `Partition` problem is implied by invariants 1 and 81 (see Table 3), while the first 166 invariants are generated in 1 second (Table 1). This suggests that the symbol elimination method generates increasingly sophisticated invariants, while natural and simple invariants are generated quickly.

4.2 Industrial Examples

We also ran Vampire's symbol elimination on 48 annotated array examples provided by Dassault Aviation. Since Vampire does not deal with pointers, we safely replaced pointers by arrays in 5 examples, and structures by arrays in 3 example loops. The 48 annotated array examples involve array copying, initialisation and shifts, and used arithmetical operations (e.g. addition, minus, plus, multiplication) and comparisons (e.g. greater, not equal) over the array content.

Vampire failed to find sufficiently strong invariants for 10 of these loops, for the following reasons. 6 loops were nested (all related to sorting algorithms) and thus cannot be analysed by the current version at all. Of the remaining 4 loops, two traversed sorted

arrays using a logarithm-time search, one accessed the array using logically complex manipulation with array indexes, and the last one computed the sum of all array elements.

The results for the remaining 38 loops are analysed in Table 2. The first row of Table 2 shows the performance of Vampire on loops having only a single path, whereas the second row gives the results for multi-path loops. The second column shows the number of such loops. The third column gives the average number of invariants generated by Vampire with a 1 second time limit. The fourth and the fifth columns show, respectively, the number of invariants in the minimised set and the percentage of invariants proved to be redundant. The last two columns show whether any quantified invariants with universal quantifiers (\forall -Inv) (respectively with quantifier alternations $\forall\exists$ -Inv) have been generated. We note that *for all 38 examples the intended invariants have been implied by the ones generated by Vampire*.

4.3 Analysis of Experiments

By studying the minimised sets of generated invariants we discovered that it still contains many redundancies and that many generated clauses could have been further improved by a better theory reasoning or algebraic simplifications.

Example 4. For the Partition_Init program, 692 invariants were generated in 10 seconds (see Table 1), out of which 127 invariants were kept in the minimised set. By further inspection of these 127 invariants, we noticed the following 2 invariants:

$$\begin{aligned} \text{inv30: } & \forall x_0, x_1, x_2 : sk_1(x_0) \neq x_1 \vee x_0 \neq x_2 \vee \neg c > x_0 \vee \neg x_0 \geq 0 \vee C(x_2) = x_1 \\ \text{inv677: } & \forall x_0, x_1 : C(x_1) = sk_1(x_0 + 2) \vee 0 \geq x_0 \vee x_0 + 2 \neq x_1 \vee \neg c > x_0 + 2 \end{aligned}$$

When running Vampire only on these two formulas (without symbol elimination and consequence generation), Vampire proves in essentially no time that $\text{inv30} \Rightarrow \text{inv677}$. Hence inv677 is redundant, but could not be proved to be redundant using the consequence elimination mode with a 20 seconds time limit.

The above example suggests thus that further refinements of integer reasoning in conjunction with first-order theorem proving are crucial for generating a minimal set of interesting invariants. We leave this issue for further research.

Based on the experiments described here, we believe that we are now ready to answer the three questions raised in Section 1 about using symbol elimination for invariant generation.

1. [Strength] For each example we tried, (i) Vampire generated complex quantified invariants as conclusions of symbol eliminating inferences (some with quantifier alternations), (ii) using the invariants inferred by Vampire, the intended invariants and loop properties of the example could be automatically proved by Vampire in essentially no time. Hence, symbol elimination proves to be a powerful method for automated invariant generation.
2. [Time] Symbol elimination in Vampire is very fast. Within a 1 second time limit a large set of complex and useful quantified invariants have been generated for each example we tried.

Table 3. Invariant generation by symbol elimination with Vampire, within 1 second time limit

Loop	# SEI	# Min SEI	Inv of interest	Generated invariants implying Inv
Initialisation [13] <i>a = 0; while (a < m) do A[a] = 0; a = a + 1 end do</i>	15	5	$\forall x : 0 \leq x < a \Rightarrow A[x] = 0$	inv7: $\forall x_0, x_1, x_2 : 0 \neq x_0 \vee x_1 \neq x_2 \vee A(x_1) = x_0 \vee \neg a > x_2 \vee \neg x_2 \geq 0$
Copy [13] <i>a = 0; while (a < m) do B[a] = A[a]; a = a + 1 end do</i>	24	5	$\forall x : 0 \leq x < a \Rightarrow B[x] = A[x]$	inv8: $\forall x_0, x_1 : A[x_0] = B[x_1] \vee x_0 \neq x_1 \vee \neg a > x_0 \vee \neg x_0 \geq 0$
Vararg [13] <i>a = 0; while (A[a] > 0) do a = a + 1 end do</i>	1	1	$\forall x : 0 \leq x < a \Rightarrow A[x] > 0$	inv0: $\forall x_0 : \neg a > x_0 \vee \neg x_0 \geq 0 \vee A(x_0) > 0$
Partition [13] <i>a = 0; b = 0; c = 0; while (a < m) do if (A[a] >= 0) then B[b] = A[a]; b = b + 1 else C[c] = A[a]; c = c + 1 end if; a = a + 1 end do</i>	166	38	$\forall x : 0 \leq x < b \Rightarrow B[x] \geq 0 \wedge \exists y : B[x] = A[y]$	inv1: $\forall x_0 : A(sk_2(x_0)) \geq 0 \vee \neg b > x_0 \vee \neg x_0 \geq 0$ inv81: $\forall x_0 : \neg b > x_0 \vee \neg x_0 \geq 0 \vee A(sk_2(x_0)) = B(x_0)$
Partition_Init [13] <i>a = 0; c = 0; while (a < m) do if (A[a] == B[a]) then C[c] = a; c = c + 1 end if; a = a + 1 end do</i>	168	24	$\forall x : 0 \leq x < c \Rightarrow A[C[x]] = B[C[x]]$	inv0: $\forall x_0 : A(sk_1(x_0)) = B(sk_1(x_0)) \vee \neg c > x_0 \vee \neg x_0 \geq 0$ inv30: $\forall x_0, x_1, x_2 : sk_1(x_0) \neq x_1 \vee x_0 \neq x_2 \vee \neg c > x_0 \vee \neg x_0 \geq 0 \vee C(x_2) = x_1$
Shift <i>a = 0; while (a < m) do A[a + 1] = A[a]; a = a + 1 end do</i>	24	5	$\forall x : 0 \leq x \leq a \Rightarrow A[x] = A[0]$	inv5: $\forall x_0, x_1, x_2 : x_0 + 1 \neq x_1 \vee A[x_0] \neq x_2 \vee A[x_1] = x_2 \vee \neg a > x_0 \vee \neg x_0 \geq 0$ inv13: $\forall x_0, x_1 : A[0] \neq x_0 \vee x_1 \neq 1 \vee A[x_1] = x_0 \vee$

3. [Quantity] Symbol elimination, even with very short time limits, can result in a large amount of invariants, ranging from one to several hundred. By interfacing symbol elimination with consequence elimination, one obtains a considerably smaller amount of non-redundant invariants: in practice, about 80% of invariants obtained by symbol elimination are normally proved to be redundant. We believe that the generated minimised set of invariants makes symbol elimination attractive for industrial software verification. It seems that the set of remaining invariants can be further reduced by better reasoning with quantifiers and theories.

5 Related Work

To the best of our knowledge, symbol elimination is a new approach that has not been previously evaluated. A related approach to symbol elimination is presented in [10] where theorem proving is used for generating interpolants as quantified invariants that imply given assertions. Predefined assertions and predicates are also the key ingredients

in [4,1,3,13] for quantified invariant inference. For doing so, predicate abstraction is employed to derive the strongest boolean combination of a given set of predicates [4,3], or invariant templates are used over predicate abstraction [13] or constraint solving [1]. Abstract interpretation is also used in [2,5], where quantified invariants are automatically inferred by an interval-based analysis over array indexes, without requiring user-given assertions. Unlike the cited works, in our experiments with symbol elimination to invariant generation, we generated complex invariants with quantifier alternations without using predefined templates, predicates and assertions, and without using abstract interpretation techniques.

Quantified array invariants are also inferred in [6], by deploying symbolic computation based program analysis over loops. Although symbolic computation offers a more powerful framework than symbol elimination when it comes to arithmetical operations, all examples reported in [6] were successfully handled by using symbol elimination for invariant generation. However, [6] can only infer universally quantified invariants, whereas our experiments show that symbol elimination can be used to derive invariants with quantifier alternations.

6 Conclusions

We describe and evaluate the recent implementation of symbol elimination in the first-order theorem prover Vampire. This implementation includes a program analysis framework, theory reasoning, efficient consequence elimination, and invariant generation.

Our experimental results give practical evidence of the strength and time-efficiency of symbol elimination for invariant generation. Furthermore, we investigated quantitative aspects of symbol elimination.

We can answer affirmatively the question whether a computer program can automatically generate powerful program properties. Indeed, the properties generated by Vampire implied the intended properties of the programs we studied. However, our research also poses highly non-trivial problems. The main problem is the large number of generated properties. On the one hand, one can say that this is an accolade to the method that it can generate many more properties than a human would ever be able to discover. On the other hand, one can say that the majority of generated properties are uninteresting. This poses the following problems that can help us understand computer reasoning and intelligence better:

1. what makes some program properties more interesting than others from the viewpoint of programmers (or applications);
2. how can one automatically tell interesting program properties from other properties generated by a computer?

Answering these fundamental questions will also help us to improve program generation methods for the purpose of applications in program analysis and verification.

References

1. Beyer, D., Henzinger, T.A., Majumdar, R., Rybalchenko, A.: Invariant Synthesis for Combined Theories. In: Cook, B., Podelski, A. (eds.) VMCAI 2007. LNCS, vol. 4349, pp. 378–394. Springer, Heidelberg (2007)

2. Gopan, D., Reps, T.W., Sagiv, M.: A Framework for Numeric Analysis of Array Operations. In: Proc. of POPL, pp. 338–350 (2005)
3. Gulwani, S., McCloskey, B., Tiwari, A.: Lifting Abstract Interpreters to Quantified Logical Domains. In: Proc. of POPL, pp. 235–246 (2008)
4. Gulwani, S., Tiwari, A.: An Abstract Domain for Analyzing Heap-Manipulating Low-Level Software. In: Damm, W., Hermanns, H. (eds.) CAV 2007. LNCS, vol. 4590, pp. 379–392. Springer, Heidelberg (2007)
5. Halbwachs, N., Peron, M.: Discovering Properties about Arrays in Simple Programs. In: Proc. of PLDI, pp. 339–348 (2008)
6. Henzinger, T.A., Hottelier, T., Kovács, L., Rybalchenko, A.: Alligators for Arrays (Tool Paper). In: Fermüller, C.G., Voronkov, A. (eds.) LPAR-17. LNCS, vol. 6397, pp. 348–356. Springer, Heidelberg (2010)
7. Hodér, K., Kovács, L., Voronkov, A.: Interpolation and Symbol Elimination in Vampire. In: Giesl, J., Hähnle, R. (eds.) IJCAR 2010. LNCS, vol. 6173, pp. 188–195. Springer, Heidelberg (2010)
8. Hodér, K., Kovács, L., Voronkov, A.: Invariant Generation in Vampire. In: Abdulla, P.A., Leino, K.R.M. (eds.) TACAS 2011. LNCS, vol. 6605, pp. 60–64. Springer, Heidelberg (2011)
9. Kovács, L., Voronkov, A.: Finding Loop Invariants for Programs over Arrays Using a Theorem Prover. In: Chechik, M., Wirsing, M. (eds.) FASE 2009. LNCS, vol. 5503, pp. 470–485. Springer, Heidelberg (2009)
10. McMillan, K.L.: Quantified Invariant Generation Using an Interpolating Saturation Prover. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 413–427. Springer, Heidelberg (2008)
11. Riazanov, A., Voronkov, A.: The Design and Implementation of Vampire. AI Communications 15(2-3), 91–110 (2002)
12. Robinson, A., Voronkov, A.: Handbook of Automated Reasoning, vol. 1. Elsevier Science, Amsterdam (2001)
13. Srivastava, S., Gulwani, S.: Program Verification using Templates over Predicate Abstraction. In: Proc. of PLDI, pp. 223–234 (2009)

Characterization of Argumentation Semantics in Terms of the MM^r Semantics

Mauricio Osorio¹, José Luis Carballido², Claudia Zepeda, and Zenaida Cruz²

¹ Universidad de las Américas, CENTIA,
Sta. Catarina MÁrtir, Cholula, Puebla, 72820 México
osoriomauri@gmail.com,

² Benemérita Universidad Atómica de Puebla
Facultad de Ciencias de la Computación,
Puebla, Puebla, México
{czepedac,jlcarballido7}@gmail.com

Abstract. Argumentation theory studies the fundamental mechanism humans use in argumentation and explores ways to implement this mechanism on computers. Dung's approach, presented in [9], is a unifying framework which has played an influential role on argumentation research. In this paper, we show that, a logic programming semantics, called MM^r , can be used to characterize the preferred argumentation semantics defined by Dung in [9]. The MM^r [12] is based on the *the minimal model semantics*. The characterization of this argumentation semantics by the MM^r semantics suggests a new perception of this argumentation semantics in terms of *logic foundations*.

Keywords: Argumentation semantics, logic programming semantics MM^r .

1 Introduction

The main purpose of argumentation theory is to study the fundamental mechanism humans use in argumentation and to explore ways to implement this mechanism on computers. Two of the most known argumentation approaches have been presented by Dung and Baroni:

— Dung's approach, presented in [9], is a unifying framework which has played an influential role on argumentation research and AI. This approach is mainly orientated to manage the interaction of arguments. The interaction of arguments is supported by four extension-based argumentation semantics: *stable semantics*, *preferred semantics*, *grounded semantics*, and *complete semantics*. The central notion of these semantics is the *acceptability of the arguments*. Even though each of these argumentation semantics represents different patterns of selection of arguments, all of them are based on the basic concept of *admissible set*. Informally speaking, an admissible set presents a coherent and defendable point of view in a conflict between arguments. According to Bench-Capon and Dunne [5], the three principal abstract argumentation semantics introduced by Dung are the

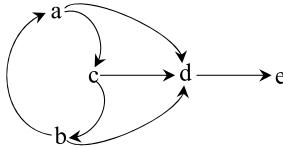


Fig. 1. Graph representation of the argumentation framework $AF = \langle \{a, b, c, d, e\}, \{(a, c), (c, b), (b, a), (a, d), (c, d), (b, d), (d, e)\} \rangle$

grounded, preferred and stable semantics. However, these semantics exhibit a variety of problems which have been illustrated in the literature [4,5,6,14]. For instance, an interesting argumentation framework which has been commented in the literature [14,4] is presented in Figure 1. The preferred semantics w.r.t. this argumentation framework is only able to infer the empty set.

— Baroni *et al*'s approach , in [4] is maybe the most general solution defined until now for improving Dung's approach. P. Baroni *et al*, have suggested that in order to overcome Dung's abstract argumentation semantics problems, it is necessary to define flexible argumentation semantics which are not necessarily based on admissible sets. For instance, the argumentation semantics $CF2$ suggested by Baroni in [4] is able to infer the extensions: $\{a, e\}$, $\{b, e\}$, $\{c, e\}$ from the argumentation framework of Figure 1. This means that $CF2$ regards argument e as an acceptable argument. Baroni *et al*'s approach is based on a solid concept in graph theory: *strongly connected components*. Based on this concept, they describe a recursive approach for generating new argumentation semantics.

At this point it is worth to mention that argumentation theory can be viewed as a special form of logic programming with negation as failure. In this case, it is desirable that normal programs represent a dispute among arguments and give an answer that infers the *winning* arguments in a dispute among arguments. This answer corresponds to the models obtained form the normal program based on a logic programming semantics. For instance, in [9] the grounded argumentation semantics is characterized by the logic programming semantics WFS (see definition of WFS in [10]); also in [9] the stable argumentation semantics is characterized by the stable logic programming semantics (see definition of stable semantics in [11]); in [16] the correspondence between complete extensions and 3-valued stable models of a logic program is defined; in [7] the preferred semantics is characterized by the p-stable semantics (see definition of p-stable semantics in [13]); and in [12] the argumentation semantics $CF2$ (introduced by P. Baroni et al. [4]) is characterized by the MM^r semantics (see definition of MM^r semantics in [12]).

The logic programming semantics MM^r has particular importance for this paper, since our main result shows that MM^r can characterize the preferred argumentation semantics defined by Dung in [9].

Hence, the MM^r semantics is a proper logic programming semantics able to capture *two different argumentation semantics*: the argumentation semantics $CF2$ proposed by Baroni and the preferred argumentation semantics proposed by Dung.

Let us observe that the characterization of the argumentation semantics $CF2$ and the preferred argumentation semantics based on the logic programming semantics MM^r , suggest a new perception of the preferred argumentation semantics and give the option to analyze these argumentation semantics from the point of view of the theoretical foundations of logic programming.

In this paper we also show, that the logic programming codification used to map an argumentation framework into a normal logic program (P_{AR}) to obtain the argumentation semantics $CF2$ based on the MM^r semantics, can be extended with some other particular clauses and the obtained normal program can be used to obtain the preferred argumentation semantics based also on the MM^r semantics.

It is important to mention that currently there exists a software implementation of the semantics MM^r . The semantics MM^r can be downloaded from <http://aplicacionesia.cs.buap.mx/~arkerz/> (Windows version) and at <http://sites.google.com/site/computingstablesemantics/downloads> (Linux version).

This software implementation can be used to test argumentation problems where $CF2$ or the preferred argumentation semantic is needed to solve the problem.

The rest of the paper is divided as follows: In §2, we present some basic concepts w.r.t. logic programming and argumentation theory, we also present the definition of MM^r semantics. In §3, we review the relationship between the argumentation semantics $CF2$ and the MM^r semantics. In §4, we show how to induce the preferred argumentation semantics from the logic programming semantics MM^r . In the last section, we present our conclusions.

2 Background

In this section, we define the syntax of the normal logic programs. We also present the definition of the MM^r semantics. Finally, we present a short description of Dung's and Baroni's argumentation approaches.

2.1 Syntax and Some Operations

A signature \mathcal{L} is a finite set of elements that we call atoms. A *literal* is either an atom a , called *positive literal*; or the negation of an atom $\neg a$, called *negative literal*. Given a set of atoms $\{a_1, \dots, a_n\}$, we write $\neg\{a_1, \dots, a_n\}$ to denote the set of atoms $\{\neg a_1, \dots, \neg a_n\}$. A *normal clause*, C , is a clause of the form

$$a \leftarrow b_1 \wedge \dots \wedge b_n \wedge \neg b_{n+1} \wedge \dots \wedge \neg b_{n+m}$$

where a and each of the b_i are atoms for $1 \leq i \leq n + m$. In a slight abuse of notation we will denote such a clause by the formula $a \leftarrow \mathcal{B}^+ \cup \neg \mathcal{B}^-$ where the set $\{b_1, \dots, b_n\}$ will be denoted by \mathcal{B}^+ , and the set $\{b_{n+1}, \dots, b_{n+m}\}$ will be denoted by \mathcal{B}^- . Given a normal clause $a \leftarrow \mathcal{B}^+ \cup \neg \mathcal{B}^-$, denoted by r , we say that $a = H(r)$ is the *head* and $\mathcal{B}^+(r) \cup \neg \mathcal{B}^-(r)$ is the *body* of the clause. If the body of a clause is empty, then the clause is known as a *fact* and can be denoted just

by: $a \leftarrow$ or $a \leftarrow \top$. We define a *normal logic program* P , as a finite set of normal clauses. We write \mathcal{L}_P , to denote the set of atoms that appear in the clauses of P . We denote by $\text{Head}(P)$ the set $\{a \mid a \leftarrow \mathcal{B}^+ \cup \neg \mathcal{B}^- \in P\}$. From now on, by *program* we will mean a normal logic program when ambiguity does not arise.

Remark 1. We want to point out that our negation symbol, \neg , corresponds to “not” in the standard use of *Logic Programming*.

A program P induces a notion of *dependency* between atoms from \mathcal{L}_P [12]. We say that a *depends immediately on* b , if and only if, b appears in the body of a clause in P , such that a appears in its head. The two place relation *depends on* is the transitive closure of *depends immediately on* [12]. The set of dependencies of an atom x , denoted by *dependencies-of*(x), corresponds to the set $\{a \mid x \text{ depends on } a\}$.

Example 1. [12] Let us consider the following program,

$$P = \{e \leftarrow e, \quad c \leftarrow c, \quad a \leftarrow \neg b \wedge c, \quad b \leftarrow \neg a \wedge \neg e, \quad d \leftarrow b\}.$$

The dependency relations between the atoms of \mathcal{L}_P are as follows: *dependencies-of*(a) = $\{a, b, c, e\}$; *dependencies-of*(b) = $\{a, b, c, e\}$; *dependencies-of*(c) = $\{c\}$; *dependencies-of*(d) = $\{a, b, c, e\}$; and *dependencies-of*(e) = $\{e\}$.

We take $<_P$ to denote the strict partial order defined as follows: $x <_P y$, if and only if, y *depends-on* x and x does not *depend-on* y . By considering the relation $<_P$, each atom of \mathcal{L}_P is assigned an order as follows: An atom x is of order 0, if x is minimal in $<_P$. An atom x is of order $n + 1$, if n is the maximal order of the atoms on which x depends.

We say that a program P is of order n , if n is the maximum order of its atoms. We can also break a program P of order n into the disjointed union of programs P_i with $0 \leq i \leq n$, such that P_i is the set of clauses for which the head is of order i (w.r.t. P). The empty program has order 0. We say that P_0, \dots, P_n are the *components* of P .

Example 2. By considering the program P in Example 1, we can see that: d is of order 2, a is of order 1, b is of order 1, e is of order 0, and c is of order 0. This means that P is a program of order 2. Thus the program P can be broken into the disjointed union of the following relevant modules or components: $P_0 = \{e \leftarrow e, \quad c \leftarrow c\}$, $P_1 = \{a \leftarrow \neg b \wedge c, \quad b \leftarrow \neg a \wedge \neg e\}$, $P_2 = \{d \leftarrow b\}$.

2.2 The MM^r Semantics

Now, we present the definition of the MM^r semantics as was defined in [12]. The definition of MM^r semantics given in [12] includes a set of transformations, denoted by \mathcal{CS} , that are applied to the program. This set of transformations renders the semantics more suitable for applications in the context of logic programming. However in this work the application of such transformations may be omitted due to the type of programs with which we are dealing. Therefore the reader can skip step two in the reduction $R(P, A)$, used to define the MM^r semantics below.

The semantics MM^r is based on the *minimal model semantics* (denoted by MM). In this paper, a logic programming semantics S is a mapping from the class of all programs into the power set of the set of (2-valued) models. From now on, we assume that the reader is familiar with the notion of an *interpretation* and *validity* [15]. An interpretation M is called a (2-valued) *model* of P if and only if for each clause $c \in P$, $M(c) = 1$. We say that M is a *minimal model* of P if and only if there does not exist a model M' of P such that $M' \subset M$, $M' \neq M$ [15]. We will denote by $MM(P)$ the set of all the minimal models of a given logic program P . Usually MM is called *minimal model semantics*.

Example 3. Let P be the program $\{a \leftarrow \neg b, b \leftarrow \neg a, a \leftarrow \neg c, c \leftarrow \neg a\}$. As we can see, P has five models: $\{a\}$, $\{b, c\}$, $\{a, c\}$, $\{a, b\}$, $\{a, b, c\}$; however, P has just two minimal models: $\{b, c\}$, $\{a\}$. Hence $MM(P) = \{\{b, c\}, \{a\}\}$.

Next we present a reduction that will be used to define the MM^r semantics.

Let P be a program and $A = \langle T; F \rangle$ be a pair of disjoint sets of atoms. The reduction $R(P, A)$ is obtained by 2 steps [12]:

1. Let $R'(P, A)$ be the program obtained in the following steps:
 - (a) We replace every atom x that occurs in the bodies of P by 1 if $x \in T$, and we replace every atom x that occurs in the bodies of P by 0 if $x \in F$;
 - (b) we replace every occurrence of $\neg 1$ by 0 and $\neg 0$ by 1;
 - (c) every clause with a 0 in its body is removed;
 - (d) finally we remove every occurrence of 1 in the body of the clauses.
2. Given the set of transformations $CS = \{RED^+, RED^-, Success, Failure, Loop\}$, in [8] it is shown that a normal program P can be reduced to another normal program $norm_{CS}(P)$ after applying those transformations a finite number of times. The program $norm_{CS}(P)$ is unique and is called the normal form of program P with respect to the system CS . We will denote $R(P, A) = norm_{CS}(R'(P, A))$. The definitions of the transformations in CS are:
 - (a) If $r \in P$ and $a \in B^-(r)$ $\exists r' \in P : H(r') = a$, then
 $RED^+(P) = (P \setminus \{r\}) \cup \{H(r) \leftarrow B^+(r) \cup \neg(B^-(r) \setminus \{a\})\}$.
 - (b) If $r \in P$ and $a \leftarrow \in P$ such that $a \in B^-(r)$, then $RED^-(P) = P \setminus \{r\}$.
 - (c) If $r \in P$ and $a \leftarrow \in P$ such that $a \in B^+(r)$, then
 $Success(P) = (P \setminus \{r\}) \cup \{H(r) \leftarrow (B^+(r) \setminus \{a\}) \cup \neg B^-(r)\}$.
 - (d) If $r \in P$ and $a \in B^+(r)$ $\exists r' \in P : H(r') = a$, then $Failure(P) = P \setminus \{r\}$.
 - (e) Let M be unique minimal model of the positive program
 $POS(P) = \{H(r) \leftarrow B^+(r) : r \in P\}$ then
 $LOOP(P) = \{r : r \in P, B^+(r) \subseteq M\}$.

Example 4. [12] Let $Q = \{a \leftarrow \neg b \wedge c, b \leftarrow \neg a \wedge \neg e, d \leftarrow b, b \leftarrow e, m \leftarrow n, n \leftarrow m\}$, and let A be the pair of sets of atoms $\langle \{c\}; \{e\} \rangle$. Thus,
 $R'(Q, A) = \{a \leftarrow \neg b, b \leftarrow \neg a, d \leftarrow b, m \leftarrow n, n \leftarrow m\}$. Hence,
 $R(Q, A) = \{a \leftarrow \neg b, b \leftarrow \neg a, d \leftarrow b\}$.

Now, in order to define the MM^r semantics, we first define the MM_c^r semantics in terms of the Minimal Model semantics, denoted by MM .

Definition 1. Given $\mathbf{A} = \{A_1 \dots A_n\}$ where the A_i , $1 \leq i \leq n$ are sets, and $\mathbf{B} = \{B_1 \dots B_m\}$ where the B_j , $1 \leq j \leq m$ are sets, we define
 $\mathbf{A} \uplus \mathbf{B} = \{A_i \cup B_j \mid A_i \in \mathbf{A} \text{ and } B_j \in \mathbf{B}\}$.

Definition 2. [12] We define the associated MM_c^r semantics recursively as follows: Given a program P of order 0, $MM_c^r(P) = MM(P)$. For a program P of order $n > 0$ we define $MM_c^r(P) = \bigcup_{M \in MM(P_0)} \{M\} \uplus MM_c^r(R(P \setminus P_0, \langle M; N \rangle))$, where $N := (\mathcal{L}_{P_0} \cup \{a \in \mathcal{L}_P \mid a \notin Head(P)\}) \setminus M$, and

$$MM^r(P) = MM_c^r(\text{norm}_{CS}(P))$$

The following example illustrates our two previous definitions.

Example 5. Let us consider the program $E = \{a \leftarrow \neg b, b \leftarrow \neg a, p \leftarrow \neg b, p \leftarrow \neg \neg p\}$. Note that $E = \text{norm}_{CS}(E)$. We are going to compute the $MM_c^r(E)$. We can verify that E is of order 1. We see that $E_0 = \{a \leftarrow \neg b, b \leftarrow \neg a\}$ is the component of order 0 of program E . Thus $MM(E_0) = MM(E_0) = \{\{a\}, \{b\}\}$. In order to obtain $MM_c^r(R(E \setminus E_0, \langle M; N \rangle))$ for each $M \in MM(E_0)$ we consider two cases:

Case a) Let us consider M to be $\{a\}$. Then E' is the program $R(E \setminus E_0, \langle M; N \rangle)$ with $E \setminus E_0 = \{p \leftarrow \neg b, p \leftarrow \neg \neg p\}$, and $N = \{b\}$. We can see that $E' = \{p \leftarrow, p \leftarrow \neg p\}$. Now we need to obtain $MM_c^r(E')$ which is the same as $MM(E') = \{\{p\}\}$. Hence, $\{M\} \uplus MM_c^r(E') = \{\{a\}\} \uplus \{\{p\}\} = \{\{a, p\}\}$.

Case b) Let us consider M to be $\{b\}$. Let E' be the program $R(E \setminus E_0, \langle M; N \rangle)$ with $E \setminus E_0 = \{p \leftarrow \neg b, p \leftarrow \neg \neg p\}$, and $N = \{a\}$. We can see that $E' = \{p \leftarrow \neg p\}$. Now we need to obtain $MM_c^r(E')$ which is the same as $MM(E') = \{\{p\}\}$. Hence, $\{M\} \uplus MM_c^r(E') = \{\{b\}\} \uplus \{\{p\}\} = \{\{b, p\}\}$.

Finally, we can see that $MM_c^r(E) = \{\{a, p\}, \{b, p\}\}$. Moreover, since E is a program with no tautologies then $MM^r(E) = MM_c^r(E)$.

2.3 Argumentation Theory

We define some basic concepts of the preferred semantics defined by Dung [9] and some results about how to regard his argumentation approach as logic programming with negation as failure. Also we review the *CF2* semantics defined by Baroni et al. in [4].

Dung's argumentation approach. All the definitions of this subsection were taken from [9]. The basic structure of Dung's argumentation approach is an argumentation framework which captures the relationships between the arguments.

Definition 3. An argumentation framework is a pair $AF := \langle AR, attacks \rangle$, where AR is a finite set of arguments, and $attacks$ is a binary relation on AR , i.e., $attacks \subseteq AR \times AR$.

Any argumentation framework could be regarded as a directed graph. For instance, if $AF := \langle \{a, b, c\}, \{(a, b), (b, c)\} \rangle$, then AF is represented as in Figure 2.

$$a \longrightarrow b \longrightarrow c$$

Fig. 2. Graph representation of the argumentation framework $AF = \langle \{a, b, c\}, \{(a, b), (b, c)\} \rangle$

We say that a attacks b (or b is attacked by a) if $\text{attacks}(a, b)$ holds. Similarly, we say that a set S of arguments attacks b (or b is attacked by S) if b is attacked by an argument in S . For instance in Figure 2, $\{a\}$ attacks b .

Definition 4. A set S of arguments is said to be conflict-free if there are no arguments a, b in S such that a attacks b .

Dung defined his semantics based on the basic concept of admissible set.

Definition 5. (1) An argument $a \in AR$ is acceptable with respect to a set S of arguments if and only if for each argument $b \in AR$: If b attacks a then b is attacked by S . (2) A conflict-free set of arguments S is admissible if and only if each argument in S is acceptable w.r.t. S .

For instance, the argumentation framework of Figure 2 has two admissible sets: $\{a\}$ and $\{a, c\}$.

Definition 6. A preferred extension of an argumentation framework AF is a maximal (w.r.t. inclusion) admissible set of AF . The set of preferred extensions of AF , denoted by $\text{preferred}(AF)$, will be referred as the preferred semantics of AF .

The only preferred extension of the argumentation framework of Figure 2 is $\{a, c\}$.

Remark 2. For the rest of the paper, we will say that an argument a is “acceptable” if a belongs to E where E is a preferred extension. Also an argument attacked by a will be called “defeated”.

CF2 semantics. Now we review the definition of CF2 semantics presented in [3] (interested readers can examine in detail the definition of CF2 semantics in [1,4]). CF2 semantics, introduced in [1], is related to the notion of strongly connected components (SCC) of an argumentation framework AF . It can be regarded as a subset of the maximal conflict-free sets of AF based on the decomposition of AF into strongly connected components, denoted as SCCS_{AF} . It has the property of treating in a “symmetric” way cycles of length odd and even. In particular when AF consists of exactly one strongly connected component, the set of extensions defined by the CF2 semantics is the set of maximal conflict free sets of AF .

SCC-recursiveness is related to the graph-theoretical notion of strongly connected components of an argumentation framework AF , namely the equivalence classes of arguments under the relation of mutual reachability via attack links, denoted as SCCSAF .

Definition 7. Given an argumentation framework $AF = \langle AR, attacks \rangle$, a set $E \subseteq AR$ is an extension of $CF2$ semantics iff

- $E \in MCF_{AF}$ if $|SCCS_{AF}| = 1$.
- $\forall S \in SCCS_{AF} (E \cap S) \in \mathcal{E}_{CF2}(AF \downarrow_{UP_{AF}(S, E)})$ otherwise.

where \mathcal{E}_{CF2} denotes the set of extensions defined by $CF2$, MCF_{AF} denotes the set of maximal conflict-free sets of AF , and, for any set $S \subseteq AR$, $AF \downarrow_S$ denotes the restriction of AF to S , namely $AF \downarrow_S = \langle S, attacks \cap (S \times S) \rangle$, and $UP_{AF}(S, E) = \{\alpha \in S \mid \exists \beta \in E : \beta \notin S, (\beta, \alpha) \in attacks\}$. We define $CF2(AF) = \{E \mid E \text{ is an extension of } CF2 \text{ semantics of } AF\}$.

3 Relation between $CF2$ and MM^r

In this subsection, we review the relationship between the argumentation semantics $CF2$ introduced by P. Baroni et al. [4] and the MM^r semantics. This relationship is shown in [12]. The proposal in [12] to induce the argumentation semantics $CF2$ from the logic programming semantics MM^r is as follows: First, given an argumentation framework AF , we use a particular mapping that associates to AF a normal program denoted by P_{AF} . Second, we obtain the MM^r models of P_{AF} . Finally, we use another mapping, called f , to obtain the argumentation semantics $CF2$ of AF . The mapping f assigns to each MM^r model of P_{AF} a set of arguments from the argumentation framework AF . Now, we present the definitions of the two mappings and the theorem that indicates the relationship between $CF2$ and MM^r (see [12] for details).

The first mapping uses the predicates $d(x)$ and $a(x)$ to represent that “the argument x is defeated” and “the argument x is accepted” respectively. This mapping also includes clauses such as $d(x) \leftarrow \neg d(y)$ to capture the idea of argument x is defeated when anyone of its adversaries y is not defeated.

Definition 8. [12] Let $AF = \langle AR, attacks \rangle$ be an argumentation framework, $P_{AF}^D = \{d(x) \leftarrow \neg d(y_1), \dots, d(x) \leftarrow \neg d(y_n) \mid x \in AR \text{ and } \{y_1, \dots, y_n\} = \{y_i \in AR \mid (y_i, x) \in attacks\}\}$; and $P_{AF}^A = \bigcup_{x \in AR} \{a(x) \leftarrow \neg d(x)\}$. We define: $P_{AF} = P_{AF}^D \cup P_{AF}^A$.

The expert reader in argumentation theory can observe that essentially, P_{AF}^D captures the basic principle of *conflict-freeness* [12]. The idea of P_{AF}^A is just to infer that any argument x that is not defeated is accepted. It is worth commenting that according to Baroni and Giacomin [2], the principle of conflict-freeness is the minimal requirement to be satisfied by any argumentation semantics.

Example 6. Now, we illustrate the mapping P_{AF} , let AF be the argumentation framework of Figure 1. We can see that $P_{AF} = P_{AF}^D \cup P_{AF}^A$ where,

$$\begin{array}{ll} P_{AF}^D : & P_{AF}^A : \\ d(a) \leftarrow \neg d(b). \quad d(d) \leftarrow \neg d(a). \quad d(e) \leftarrow \neg d(d). & a(a) \leftarrow \neg d(a). \quad a(d) \leftarrow \neg d(d). \\ d(b) \leftarrow \neg d(c). \quad d(d) \leftarrow \neg d(b). & a(b) \leftarrow \neg d(b). \quad a(e) \leftarrow \neg d(e). \\ d(c) \leftarrow \neg d(a). \quad d(d) \leftarrow \neg d(c). & a(c) \leftarrow \neg d(c). \end{array}$$

Now, we see how the MM^r semantics of the program P_{AF} can induce an *extension-based argumentation semantics*, denoted by MM_M^r , under the second mapping, called f . The mapping f assigns a set of arguments from the argumentation framework AF to each MM^r model of P_{AF} . The set of arguments assigned to an MM^r model of P_{AF} by f , corresponds to the set of arguments x in the predicates $a(x)$ (that represent that “the argument x is accepted”) that appear in the MM^r model.

Definition 9. [12] Let $AF = \langle AR, Attacks \rangle$ be an argumentation framework. The semantics MM^r induces the extension-based argumentation semantics MM_M^r as follows:

$$MM_M^r(P_{AF}) = f(MM^r(P_{AF}))$$

where f is the mapping from $2^{2^{\mathcal{L}_{P_{AF}}}}$ to $2^{2^{AR}}$, such that $f(X) = \bigcup_{M \in X} \{\{y \mid a(y) \in M\}\}$.

Example 7. Let us again consider the argumentation framework AF of Figure 1. Let us consider the normal program P_{AF} previously obtained in Example 6. This means that we are going to obtain the extension-based argumentation semantics $MM_M^r(P_{AF})$. First of all, we have to obtain the semantics MM^r of P_{AF} . It is not difficult to see that $MM^r(P_{AF})$ corresponds to the following set:

$$\{\{d(a), d(b), d(d), a(c), a(e)\}, \{d(b), d(c), d(d), a(a), a(e)\}, \{d(a), d(c), d(d), a(b), a(e)\}\}$$

Hence, $MM_M^r(P_{AF}) = f(MM^r(P_{AF})) = \{\{c, e\}, \{a, e\}, \{b, e\}\}$.

Let us point out that $MM_M^r(P_{AF})$ coincides with the set of extensions of $CF2$ of AF , i.e., $MM_M^r(P_{AF}) = CF2(AF)$.

Finally, we present the theorem that indicates the relationship between MM^r and $CF2$.

Theorem 1. [12] Given an argumentation framework $AF = \langle AR, Attacks \rangle$, and given $E \subseteq AR$, $E \in MM_M^r(P_{AF})$ if and only if $E \in CF2(AF)$.

4 Preferred Extension and MM^r Semantics

The idea about to define a mapping in order to regard argumentation frameworks as logic programs was used by Dung previously in [9]. In fact, one can see in [9] that the mapping proposed by Dung is able to characterize the grounded and stable argumentation semantics; however, to the best of our knowledge, that mapping is unable to characterize the preferred argumentation semantics by considering a logic programming semantics with negation as failure.

In this section, we propose to induce the *preferred argumentation semantics* from the logic programming semantics MM^r , in a similar way as, in Section 3, the argumentation semantics $CF2$ is induced from MM^r .

First, given an argumentation framework AF , we use a particular mapping that associates to AF a normal program denoted by Ψ_{AF} . This mapping is really close to the mapping proposed by Dung in [9] and, at the same time, it is an extension of the mapping proposed in Definition 8. Second, we obtain the MM^r

models of Ψ_{AF} . Finally, we use the mapping f of Definition 9, to obtain the preferred argumentation semantics of AF . In this way, mapping f assigns to each MM^r model of Ψ_{AF} a set of arguments from the argumentation framework AF .

The definitions of the mappings and the theorem that establishes the relationship between the preferred argumentation semantics and the logic programming semantics MM^r are described next.

The new mapping also uses the predicates $d(x)$ and $a(x)$ to represent that “the argument x is defeated” and “the argument x is accepted” respectively, and it includes clauses such as $d(x) \leftarrow \neg d(y)$ to capture the idea of argument x is defeated when anyone of its adversaries y is not defeated. This mapping indicates that for each argumentation framework AF , we can associate a normal program Ψ_{AF} that corresponds to the union of program P_{AF} (see Definition 8) with a particular set of clauses.

Definition 10. Let $AF := \langle AR, Attacks \rangle$ be an argumentation framework. We define its associated normal program as follows:

$$\Psi_{AF} := P_{AF} \cup \bigcup_{a \in AR} \{\cup_{b:(b,a) \in attacks} \{d(a) \leftarrow \wedge_{c:(c,b) \in attacks} d(c)\}\}$$

Example 8. Let $AF := \langle AR, attacks \rangle$ be the argumentation framework of Figure 2. We can see that Ψ_{AF} is the following program:

$$\begin{array}{lll} d(b) \leftarrow \neg d(a). & a(a) \leftarrow \neg d(a). & d(b) \leftarrow \top. \\ d(c) \leftarrow \neg d(b). & a(b) \leftarrow \neg d(b). & d(c) \leftarrow d(a). \\ & a(c) \leftarrow \neg d(c). & \end{array}$$

Finally, the following theorem indicates that the preferred extensions of AF correspond exactly to the MM^r models of Ψ_{AF} . This theorem also uses the extension-based argumentation semantics, denoted by MM_M^r , given in Definition 9, which is a mapping that assigns a set of arguments from the argumentation framework AF to each MM^r model of Ψ_{AF} .

Theorem 2. Given an argumentation framework $AF = \langle AR, Attacks \rangle$, and given $E \subseteq AR$, $E \in MM_M^r(\Psi_{AF})$ if and only if $E \in \text{preferred}(AF)$.

Proof. Sketch. First, we need to say that the preferred argumentation semantics has a recursive formulation, based on the graph-theoretic notion of strongly connected components (SCC) of an argumentation framework (AF) [4]. The recursive formulation of the preferred argumentation semantics has the property of SCC-recursiveness, which entails that this semantics can be characterized in terms of its base function, which plays the role of a parameter in the recursive schema [4].

On the other hand, the logic programming semantics MM^r of a program also has a recursive formulation, based on the components of the given program (see Definition 2). As the recursive formulation of the preferred argumentation semantics, the definition of MM^r also can be characterized in terms of its case base, which plays the role of a parameter in the recursive definition.

Since both, the preferred argumentation semantics and the MM^r semantics, have recursive definitions then, the proof only must consider the base cases of both of them.

We know that a set of arguments of an AF is a preferred extension, if it is a maximal set (with respect to set inclusion) and is admissible in AF (see details of this definition in [4]). Hence, we can verify that the MM^r semantics of Ψ_{AF} characterizes the preferred argumentation semantics. We can see that Ψ_{AF} includes clauses such as $d(x) \leftarrow \neg d(y)$ to capture the idea of argument x is defeated when anyone of its adversaries y is not defeated, i.e., these clauses capture the idea of conflict-free [7]. At the same time Ψ_{AF} includes clauses such as $d(x) \leftarrow d(y)$ to capture the idea of admissible [7]. \square

Example 9. Let us consider the argumentation framework AF of Figure 2 and its normal program Ψ_{AF} obtained in Example 8. We need to obtain the extension-based argumentation semantics $MM_M^r(\Psi_{AF})$. So, we can verify that $MM^r(\Psi_{AF}) = \{ \{a(a), a(c), d(b)\} \}$, and $MM_M^r(\Psi_{AF}) = f(MM^r(\Psi_{AF})) = \{ \{a, c\} \}$. Moreover, $MM_M^r(\Psi_{AF})$ coincides with the preferred argumentation semantics of AF , i.e., $MM_M^r(\Psi_{AF}) = \text{preferred}(AF)$.

5 Conclusions

The logic programming semantics MM^r , can be used to characterize two argumentation semantics: the $CF2$ introduced by P. Baroni et al. [4] and the preferred semantics defined by Dung in [9].

These characterizations suggest a new perception of these argumentation semantics in terms of logic programming foundations since, it gives the option to analyze them from the point of view of the theoretical foundations of logic.

It is interesting to know that adding some particular clauses to the normal logic program (P_{AR}) used to obtain the argumentation semantics $CF2$ based on the MM^r semantics, we obtain a new normal program whose MM^r models are used to obtain the preferred argumentation semantics. Moreover, since currently there exists a software implementation of the semantics MM^r , we have the possibility of testing argumentation problems where $CF2$ or the preferred argumentation semantic is needed to model and solve the argumentation problems in a practical way.

Acknowledgment. This work was supported by the CONACyT [CB-2008-01 No.101581] and PROMEP/103.5/10/4959.

References

1. Baroni, P., Giacomin, M.: Solving Semantic Problems with Odd-Length Cycles in Argumentation. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 440–451. Springer, Heidelberg (2003)
2. Baroni, P., Giacomin, M.: On principle-based evaluation of extension-based argumentation semantics. Artificial Intelligence 171(10–15), 675–700 (2007)

3. Baroni, P., Giacomin, M.: Skepticism relations for comparing argumentation semantics. *Int. J. Approx. Reasoning* 50(6), 854–866 (2009)
4. Baroni, P., Giacomin, M., Guida, G.: SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence* 168, 162–210 (2005)
5. Bench-Capon, T.J.M., Dunne, P.E.: Argumentation in artificial intelligence. *Artificial Intelligence* 171(10-15), 619–641 (2007)
6. Caminada, M.: Semi-Stable semantics. In: Dunne, P.E., Bench-Capon, T.J. (eds.) *Proceedings of COMMA*, vol. 144, pp. 121–130. IOS Press (2006)
7. Carballido, J.L., Nieves, J.C., Osorio, M.: Inferring Preferred Extensions by Pstable Semantics. *Revista Iberomericana de Inteligencia Artificial* 13(41), 38–53 (2009)
8. Dix, J., Osorio, M., Zepeda, C.: A general theory of confluent rewriting systems for logic programming and its applications. *Ann. Pure Appl. Logic* 108(1-3), 153–188 (2001)
9. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2), 321–358 (1995)
10. Gelder, A.V., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. *Journal of the ACM* 38(3), 620–650 (1991)
11. Gelfond, M., Lifschitz, V.: The Stable Model Semantics for Logic Programming. In: Kowalski, R., Bowen, K. (eds.) *5th Conference on Logic Programming*, pp. 1070–1080. MIT Press (1988)
12. Nieves, J.C., Osorio, M., Zepeda, C.: A schema for generating relevant logic programming semantics and its applications in argumentation theory. *Fundamenta Informaticae* 106(2-4), 295–319 (2011)
13. Osorio, M., Navarro, J.A., Arrazola, J., Borja, V.: Logics with common weak completions. *Journal of Logic and Computation* 16(6), 867–890 (2006)
14. Prakken, H., Vreeswijk, G.A.W.: Logics for defeasible argumentation. In: Gabbay, D., Günthner, F. (eds.) *Handbook of Philosophical Logic*, 2nd edn., vol. 4, pp. 219–318. Kluwer Academic Publishers, Dordrecht (2002)
15. van Dalen, D.: *Logic and structure*, 3rd augmented edn. Springer, Berlin (1994)
16. Wu, Y., Caminada, M., Gabbay, D.M.: Complete extensions in argumentation coincide with 3-valued stable models in logic programming. *Studia Logica* 93(2-3), 383–403 (2009)

Learning Probabilistic Description Logics: A Framework and Algorithms

José Eduardo Ochoa-Luna¹, Kate Revoredo², and Fábio Gagliardi Cozman¹

¹ Escola Politécnica, Universidade de São Paulo,
Av. Prof. Mello Moraes 2231, São Paulo - SP, Brazil

² Departamento de Informática Aplicada, Unirio
Av. Pasteur, 458, Rio de Janeiro, RJ, Brazil
eduardo.ol@gmail.com, katerevoredo@uniriotec.br, fgcozman@usp.br

Abstract. Description logics have become a prominent paradigm in knowledge representation (particularly for the Semantic Web), but they typically do not include explicit representation of uncertainty. In this paper, we propose a framework for automatically learning a Probabilistic Description Logic from data. We argue that one must learn both concept definitions and probabilistic assignments. We also propose algorithms that do so and evaluate these algorithms on real data.

1 Introduction

Description logics (DLs) [2] form a family of knowledge representation formalisms that model the application domain by defining the relevant concepts of the domain and then using these concepts to specify properties of objects and relation among concepts. Even though DLs are quite expressive, they have limitations, particularly when it comes to modeling uncertainty. Thus probabilistic extensions to DLs have been proposed, defining different Probabilistic Description Logics (PDLs). For instance, the PDL CRALC [6,22,7] allows one to perform probabilistic reasoning by adding uncertainty capabilities to the DL ALC [2].

PDLs have been extensively investigated in the last few years [5,8,19]. To build a PDL terminology with large amounts of data, one must invest considerable resources. Thus, machine learning algorithms can be used in order to automatically learn a PDL. To the best of our knowledge, the only proposals for learning PDLs were described in [20] and [24]. Both focused on learning CRALC , but the former focused on learning concept definitions and the latter on probabilistic inclusions.

In this paper, we argue that to completely learn a PDL one must learn its concept definitions *and* probabilistic inclusions. We expect that learning algorithms can accommodate together background knowledge and deterministic and probabilistic concepts, giving each component its due relevance. Therefore, we propose a framework for automatically learning a PDL from relational data. Focusing on CRALC , we propose algorithms that do so and evaluate these algorithms on real data; compared to the existing work mentioned in the previous paragraph, we contribute by presenting a more complete framework, and by comparing several algorithms in our experiments.

The paper is organized as follows. Section 2 reviews basic concepts of DLs, PDLs, CR \mathcal{ALC} and machine learning in a deterministic setting. Section 3 presents our algorithm for PDL learning. Experiments are discussed in Section 4, and Section 5 concludes the paper.

2 Basics

In this section we briefly review both deterministic and probabilistic components of PDL.

2.1 Description Logics

Description logics (DLs) form a family of representation languages that are typically decidable fragments of first order logic (FOL) [2]. Knowledge is expressed in terms of *individuals*, *concepts*, and *roles*. The semantics of a description is given by a *domain* \mathcal{D} (a set) and an *interpretation* $\cdot^{\mathcal{I}}$ (a functor). Individuals represent objects through names from a set $N_{\text{I}} = \{a, b, \dots\}$. Each *concept* in the set $N_{\mathcal{C}} = \{C, D, \dots\}$ is interpreted as a subset of a domain \mathcal{D} . Each *role* in the set $N_{\mathcal{R}} = \{r, s, \dots\}$ is interpreted as a binary relation on the domain.

Concepts and roles are combined to form new concepts using a set of *constructors*. Constructors in the \mathcal{ALC} logic are *conjunction* ($C \sqcap D$), *disjunction* ($C \sqcup D$), *negation* ($\neg C$), *existential restriction* ($\exists r.C$), and *value restriction* ($\forall r.C$). *Concept inclusions/definitions* are denoted respectively by $C \sqsubseteq D$ and $C \equiv D$, where C and D are concepts. Concepts ($C \sqcup \neg C$) and ($C \sqcap \neg C$) are denoted by \top and \perp respectively. Information is stored in a *knowledge base* (\mathcal{K}) divided in two parts: the TBox (terminology) and the ABox (assertions). The TBox lists concepts and roles and their relationships. A TBox is acyclic if it is a set of concept inclusions/definitions such that no concept in the terminology uses itself. The ABox contains assertions about objects.

Given a knowledge base $\mathcal{K} = < \mathcal{T}, \mathcal{A} >$, the reasoning services typically include (i) consistency problem (to check whether the \mathcal{A} is consistent with respect to the \mathcal{T}); (ii) entailment problem (to check whether an assertion is entailed by \mathcal{K} ; note that this generates class-membership assertions $\mathcal{K} \models C(a)$, where a is an individual and C is a concept); (iii) concept satisfiability problem (to check whether a concept is subsumed by another concept with respect to the \mathcal{T}). The latter two reasoning services can be reduced to the consistency problem [2].

2.2 Probabilistic Description Logics and CR \mathcal{ALC}

Several probabilistic descriptions logics (PDLs) have appeared in the literature. Heinsohn [11], Jaeger [13] and Sebastiani [25] consider probabilistic inclusion axioms such as $P_{\mathcal{D}}(\text{Professor}) = \alpha$, meaning that a randomly selected object is a Professor with probability α . This characterizes a *domain-based* semantics: probabilities are assigned to subsets of the domain \mathcal{D} . Sebastiani also allows inclusions such as $P(\text{Professor}(John)) = \alpha$, specifying probabilities over the interpretations themselves. For example, one interprets $P(\text{Professor}(John)) = 0.001$ as assigning

0.001 to be the probability of the set of interpretations where *John* is a *Professor*. This characterizes an *interpretation-based* semantics.

The PDL $\text{CR}\mathcal{ALC}$ is a probabilistic extension of the DL \mathcal{ALC} that adopts an interpretation-based semantics. It keeps all constructors of \mathcal{ALC} , but only allows concept names on the left hand side of inclusions/definitions. Additionally, in $\text{CR}\mathcal{ALC}$ one can have probabilistic inclusions such as $P(C|D) = \alpha$ or $P(r) = \beta$ for concepts C and D , and for role r . If the interpretation of D is the whole domain, then we simply write $P(C) = \alpha$. The semantics of these inclusions is roughly (a formal definition can be found in [7]) given by:

$$\forall x \in \mathcal{D} : P(C(x)|D(x)) = \alpha,$$

$$\forall x \in \mathcal{D}, y \in \mathcal{D} : P(r(x, y)) = \beta.$$

We assume that every terminology is acyclic; no concept uses itself. This assumption allows one to represent any terminology \mathcal{T} through a directed acyclic graph. Such a graph, denoted by $\mathcal{G}(\mathcal{T})$, has each concept name and role name as a node, and if a concept C directly uses concept D , that is if C and D appear respectively in the left and right hand sides of an inclusion/definition, then D is a *parent* of C in $\mathcal{G}(\mathcal{T})$. Each existential restriction $\exists r.C$ and value restriction $\forall r.C$ is added to the graph $\mathcal{G}(\mathcal{T})$ as nodes, with an edge from r to each restriction directly using it. Each restriction node is a *deterministic* node in that its value is completely determined by its parents.

The semantics of $\text{CR}\mathcal{ALC}$ is based on probability measures over the space of interpretations, for a fixed domain. Inferences, such as $P(A_o(a_0)|\mathcal{A})$ for an ABox \mathcal{A} , can be computed by propositionalization and probabilistic inference (for exact calculations) or by a first order loopy propagation algorithm (for approximate calculations) [7].

2.3 Learning Description Logics

The use of ontologies for knowledge representation has been a key element of proposals for the Semantic Web [1,9]. Considerable effort is currently invested into developing automated means for the acquisition of ontologies [16].

Most early approaches were only capable of learning simple ontologies such as taxonomic hierarchies. Some recent approaches such as YINYANG [12], DL-FOIL [9] and DL-Learner [18] have focused on learning expressive terminologies (we refer to [20] for a detailed review on learning description logics). To some extent, all these approaches have been inspired by Inductive Logic Programming (ILP) [15] techniques, in that they try to transfer ILP methods to description logic settings. The goal of learning in such deterministic languages is generally to find a correct concept with respect to given examples. Given a knowledge base \mathcal{K} , a target concept *Target* such that $\text{Target} \notin \mathcal{K}$, a set $E = E_p \cup E_n$ of positive and negative examples given as assertions for *Target*, the goal of learning is to find a concept definition $C(\text{Target} \equiv C)$ such that $\mathcal{K} \cup C \models E_p$ and $\mathcal{K} \cup C \not\models E_n$.

A sound concept definition for *Target* must cover all positive examples and none of the negative examples. A learning algorithm can be constructed as a

combination of (1) a refinement operator, which defines how a search tree can be built, (2) a search algorithm, which controls how the tree is traversed, and (3) a scoring function to evaluate the nodes in the tree defining the best one.

Refinement operators allow one to find candidate concept definitions through two basic tasks: generalization and specialization [17]. Such operators in both ILP and description logic learning rely on θ -subsumption to establish an ordering so as to traverse the search space. If a concept C subsumes a concept D ($D \sqsubseteq C$), then C covers all examples which are covered by D , which makes subsumption a suitable order. Arguably the best refinement operator for description logic learning is the one available in the DL-Learner system [17,18].

In a deterministic setting a cover relationship simply tests whether, for given candidate concept definition (C), a given example e holds; that is, $\mathcal{K} \cup C \models e$ where $e \in E_p$ or $e \in E_n$. In this sense, a cover relationship $\text{cover}(e, \mathcal{K}, C)$ indicates whether a candidate concept covers a given example [9].

In DL learning, one often compares candidates through score functions based on the number of positive/negative examples covered. In DL-Learner a fitness relationship considers the number of positive examples as well as the length of solutions when expanding candidates in the tree search.

The learning algorithm depends basically on the way we traverse the candidate concepts obtained after applying refinement operators. In a deterministic setting the search for candidate concepts is often based on the FOIL [23] algorithm. There are also different approaches (for instance, DL-Learner, an approach based on genetic algorithms [16], and one that relies on horizontal expansion and redundancy checking to traverse search trees [18]).

3 Learning with the PDL CRALC

A probabilistic terminology consists of both concepts definitions and probabilistic components (probabilistic inclusions in CRALC). The key in learning under a combined approach is to give a due relevance to each component. In this section we combine existing algorithms into a single framework that allows for logical and probabilistic learning.

We argue that negative and positive examples underlie the choice of either a concept definition or a probabilistic inclusion. In a deterministic setting we expect to find concepts covering all positive examples, which is not always possible. It is common to allow flexible heuristics that deal with these issues. Moreover, there are several examples that cannot be ascribed to candidate hypotheses¹. When we are unable to find a concept definition that covers all positive examples we assume such hypothesis as candidates to be a probabilistic inclusion and we begin the search for the best probabilistic inclusion that fits the examples.

As in description logic learning three tasks are important and should be considered: (1) refinement operators, (2) scoring functions and (3) a traverse search space algorithm.

¹ In some cases the Open World Assumption inherent to description logics prevent us for stating membership of concepts.

The refinement operator described previously is used for learning the deterministic component of probabilistic terminologies.

3.1 The Probabilistic Score Function

In our proposal, since we want to learn probabilistic terminologies, we adopt a probabilistic cover relation given in [14]:

$$\text{cover}(e, \mathcal{K}, C) = P(e|\mathcal{K}, C).$$

Every candidate hypothesis together with a given example turns out to be a probabilistic random variable which yields true if the example is covered, and false otherwise. To guarantee soundness of the ILP process (that is, to cover positive examples and not to cover negative examples), the following restrictions are needed:

$$P(e_p|\mathcal{K}, C) > 0, \quad P(e_n|\mathcal{K}, C) = 0.$$

In this way a probabilistic cover relationship is a generalization of the deterministic cover, and is suitable for a combined approach. Probabilities can be computed through Bayes' theorem:

$$P(e|\mathcal{K}, C_1, \dots, C_k) = \frac{P(C_1, C_2, \dots, C_k|T)P(T)}{P(C_1, \dots, C_k)},$$

where C_1, \dots, C_k are candidate concepts definitions, and T denotes the target concept variable. Here are three possibilities for modeling $P(C_1, \dots, C_k|T)$: (1) a naive Bayes assumption may be adopted [14] (each candidate concept is independent given the target), and then $P(C_1, \dots, C_k|T) = \prod_i P(C_i|T)$; (2) the noisy-OR function may be used [20]; (3) a less restrictive option based on tree augmented naive Bayes networks (TAN) may be handy [14]. This last possibility has been considered for the probabilistic cover relationship used in this paper. In each case probabilities are estimated by maximum (conditional) likelihood parameters. The candidate concept definition C_i with the highest probability $P(C_i|T)$ is the one chosen as the best candidate.

As we have chosen a probabilistic cover relationship, our probabilistic score is defined accordingly:

$$\text{score}(\mathcal{K}|C) = \prod_{e_i \in E_p} P(e_i|\mathcal{K}, C),$$

where C is the best candidate chosen as described before.

In the probabilistic score we have previously defined, a given threshold allows us to differentiate between a deterministic and probabilistic inclusion candidate. Further details are given in the next section.

3.2 The Algorithm to Learn Probabilistic Terminologies

The choice between a deterministic or a probabilistic inclusion is based on a probabilistic score. We start by searching a deterministic concept. If after a set

of iterations the score of the best candidate is below a given threshold, searching for a probabilistic inclusion is better than to keep searching for a deterministic concept definition. Then, the current best k -candidates are considered as start point for probabilistic inclusion search. The complete learning procedure is shown in Algorithm 1.

Require: an initial knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, a target concept $C_{\mathcal{T}}$ and a training set E .

- 1: SearchTree with a node $\{C = \top, h = 0\}$
- 2: **repeat**
- 3: choose node $N = \{C, h\}$ with highest probabilistic score in SearchTree
- 4: expand node to length $h + 1$:
- 5: add all nodes $D \in (\text{refinementOperator}(C))$ with length $= h + 1$
- 6: learn parameters for all nodes D
- 7: $N = \{C, h + 1\}$
- 8: expand alternative nodes according to horizontal expansion factor and $h + 1$ [18]
- 9: **until** stopping criterion
- 10: N' = best node in SearchTree
- 11: **if** $\text{score}(N') > \text{threshold}$ **then**
- 12: return deterministic concept $C' \in N'$ ($C_{\mathcal{T}} \equiv C'$)
- 13: **else**
- 14: call ProbabilisticInclusion(SearchTree, $C_{\mathcal{T}}$)
- 15: **end if**

Algorithm 1. Algorithm for learning probabilistic terminologies

The algorithm starts with an overly general concept definition in the root of the search tree (line 1). This node is expanded according to refinement operators and horizontal expansion criterion (line 4), i.e., child nodes obtained by refinement operators are added to the search tree (line 5). The probabilistic parameters of these child nodes are learned (line 6) and then they are evaluated with the best one chosen for a new expansion (line 3) (alternative nodes based on horizontal expansion factor are also considered (line 8)). This process continues until a stopping criterion is attained (difference for scores is insignificant); After that, the best node obtained is evaluated and if it is above a threshold, a deterministic concept definition is found and returned (line 12). Otherwise, a probabilistic inclusion procedure is called (line 14).

The Algorithm 2 learns probabilistic inclusions. It starts retrieving the best k nodes in the search tree and computing the conditional mutual information for every pair of nodes (line 2). Then an undirected graph is built where the vertices are the k nodes and the edges are weighted with the value of the conditional mutual information [21] for each pair of vertices (lines 4 and 5). A maximum weight spanning tree [4] from this graph is built (line 6) and the target concept is added as a parent for each vertex (line 7). The probabilistic parameters are learned (line 8). This learned TAN-based classifier [10] is used to evaluate the possible probabilistic inclusion candidates (line 9) and the best one is returned.

Require: SearchTree previously computed and a target concept C_T

- 1: **for** each pair of candidates C_i, C_j in first k nodes of the SearchTree **do**
- 2: compute the conditional mutual information $I(C_i, C_j|C_T)$
- 3: **end for**
- 4: build an undirected graph in which vertices are the k candidates
- 5: annotate the weight of an edge connecting C_i to C_j by the $I(C_i, C_j|C_T)$
- 6: build a maximum weight spanning tree from this graph
- 7: add C_T as parent for each C_i
- 8: learn probabilities for $P(C_i|Parents(C_i))$
- 9: return the highest probabilistic inclusion $P(C_T|C') = \alpha$

Algorithm 2. Algorithm for learning probabilistic inclusions

4 Experiments

In order to evaluate the proposed framework we have divided the analysis in two stages. In a first stage, the framework was compared with the arguably best description logic learning algorithm available (the DL-Learner system). The second stage evaluated suitability of the framework for learning probabilistic CRALC terminologies in real world domains. In this sense, comparisons with two previous approaches [20,24] were performed. Experiments were run on a CORE 2 DUO 2.2 GHz computer with 4GB memory over a Linux Ubuntu platform. Each stage is detailed as follows.

4.1 Experiments on Description Logic Learning

The aim of the first stage was to investigate whether by introducing a probabilistic setting the framework behaves as well as traditional deterministic approaches in description logic learning tasks. In this preliminary evaluation (generally speaking, there is a lack of evaluation standards in ontology learning [18]) we have considered five datasets available in the DL-Learner system and reported in [18]. After a five-fold cross validation process, both accuracy and definitions length were measured. Evaluation results are shown in Table 1.

Table 1. Learning description logics results. Accur. (Accuracy) and Length (solution length)

Problem	DL-Learner		Noisy		Framework	
	Accur.	Length	Accur.	Length	Accur.	Length
trains	100	5	99.5	7	100	6
moral I	97.8	3	96.5	5	97.7	4
moral II	97.8	8	96.0	8	97.6	8
poker I	100	5	98.8	6	100	5
poker II	100	11	99.7	12	100	11

The best solution is the one that has both highest accuracy and shortest length. In Table 1, three algorithms are compared: (DL-Learner) the best description logic learning algorithm; (Noisy) a probabilistic Noisy-OR approach for learning CRALC [20] with focus on concept learning and (Framework) the combined approach proposed in this paper. The Noisy algorithm had a slightly lower performance than DL-Learner in terms of accuracy. On the other hand, the Framework (combined approach) obtained similar accuracies like DL-Learner. Further differences are observed when solution lengths are under analysis. The DL-Learner algorithm is concerned with finding shorter solutions (longer solutions are penalized in DL-Learner) whereas in the Noisy algorithm this issue is neglected. Conversely, the Framework aims at finding suitable shorter solutions because it also uses such a penalization scheme.

Despite of the fact that the framework is focused on learning probabilistic terminologies it is worth noting that two of the tested algorithms (mainly the Framework) can be competitive for learning description logics. It should be noted, however, that some induced definitions can be only meaningful when a probabilistic component is attached.

4.2 Experiments on Learning Probabilistic Terminologies

In this second stage we focused on learning of CRALC probabilistic terminologies from real world data. To do so two public data sources were explored: the Lattes curriculum platform and the YAGO-Wikipedia ontology. Due to the lack of previous learning results, comparisons have been performed by measuring accuracy and fitness of probabilistic terminologies to data. In addition, two previous approaches for learning CRALC [20,24] have also been evaluated.

In the first set of experiments, datasets were constructed from information extracted of the Lattes curriculum platform². This repository offers public relational data from Brazilian researchers. Information is given in HTML format so a parsing procedure was run to extract assertions on concepts and roles. Examples include name, institution, languages, address, etc. It was focused extraction of relational information through publication analysis, works supervised, examination board participations and so on. The universe was restricted to 1050 researchers. In Table 2 some features of the Lattes datasets are given.

Table 2. Lattes curriculum datasets

Problem	axioms	concepts	roles	examples
lattes I	2494	14	12	208
lattes II	2603	14	10	146
lattes III	2650	13	10	230
lattes IV	2350	2	13	34

After a 10-fold cross validation process, accuracy (Accur.) and amount of nodes originated (# Nodes) when constructing relational Bayesian network were

² <http://lattes.cnpq.br>

Table 3. Results of CR \mathcal{ALC} learning (Lattes). Accur. (accuracy) and # Nodes (amount of nodes of relational Bayesian network).

Problem	Noisy		Inclusion		Framework	
	Accur.	# Nodes	Accur.	# Nodes	Accur.	# Nodes
lattes I	76.74	48	76.84	51	77.13	49
lattes II	69.52	54	72.77	58	74.42	56
lattes III	73.48	56	74.68	62	78.45	58
lattes IV	71.5	78	71.56	110	72.34	75

considered for comparisons. Table 3 shows some experimental results obtained in these Lattes datasets. The best solution is the one that returns the highest accuracy and the fewest amount of nodes. The Noisy algorithm [20] got the worst values on accuracy and fewer amount of possible nodes. The Inclusion algorithm (which focuses learning of probabilistic inclusions [24]) got better results in accuracy than Noisy, however, it returned a greater amount of nodes. Finally, the Framework proposed (combined algorithm) obtained the best results on accuracy and a lower amount of nodes like the Noisy algorithm.

Further experiments were performed on datasets obtained from Wikipedia-YAGO ontology. Wikipedia articles consist mostly of free text, but also contain various types of structured information in the form of Wiki markup. Such information includes infobox templates, categorization information, images geo-coordinates, links to external Web pages, disambiguation pages, redirects between pages, and links across different language editions of Wikipedia.

In the last years, there were several projects aimed at structuring such huge source of knowledge. Examples include The DBpedia project [3], which extracts structured information from Wikipedia and turns it into a rich knowledge base, and YAGO [26], a semantic knowledge base based on data from Wikipedia and WordNet³. Currently, YAGO knows more than 2 million entities (like persons, organizations, cities, etc.). It knows 20 million facts about these entities. Unlike many other automatically assembled knowledge bases, YAGO has a manually confirmed accuracy of 95%. Several domains ranging from films, places, historical events, wines, etc. have been considered in this ontology. Moreover, facts are given as binary relationships that are suitable for our learning settings.

Subsets of Wikipedia-YAGO facts were used for learning probabilistic CR \mathcal{ALC} terminologies. Two domains were focused: first, about scientists (1335); second, about film directors (2000). Table 4 shows some features of these datasets.

In Table 5 results obtained with these datasets are exhibited. After a 10-fold cross validation process, accuracy and amount of nodes when creating relational Bayesian network were evaluated. Learning in these datasets was a more complex task than Lattes. Indeed, it can be inferred of the accuracy values (lower values) and the amount of nodes obtained. Meanwhile, the same patterns were observed, i.e., the Framework returned the best results in accuracy and a lower amount

³ wordnet.princeton.edu/

Table 4. Dataset Wikipedia-YAGO

Problem	axioms	concepts	roles	examples
scientists I	7180	11	32	438
scientists II	6534	11	32	378
directors I	8006	12	35	626
directors II	7234	12	35	389

Table 5. Results of CRALC learning (Wikipedia-YAGO). Accur. (accuracy) and # Nodes (amount of nodes in relational Bayesian network).

Problem	Noisy		Inclusion		Framework	
	Accur.	# Nodes	Accur.	# Nodes	Accur.	# Nodes
scientists I	81.4	53	82.33	56	83.44	55
scientists II	82.3	65	82.46	69	84.55	68
directors I	65.67	67	67.14	69	73.56	66
directors II	69.45	74	71.3	79	74.75	76

of nodes. On the other hand, the Inclusion algorithm had better accuracy than Noisy but a greater amount of nodes.

By evaluating these two real world datasets, we conclude that the combined framework outerperforms the other approaches in learning probabilistic CRALC terminologies. Our proposal obtains balanced results in accuracy and structural complexity.

5 Conclusion

We have presented a framework for learning deterministic/probabilistic components of terminologies expressed in CRALC, and contributed with experiments that show the value of the framework. This unified learning scheme employs a refinement operator for traversing the search space, a probabilistic cover and score relationships for evaluating candidates, and a mixed search procedure. The search aims at finding deterministic concepts; if the score obtained is below a given threshold, a probabilistic inclusion search is conducted (a probabilistic classifier is produced). Experiments with probabilistic terminologies in two real-world domains suggest that this framework do lead to improved likelihoods.

We note that the current literature does not explore in depth the use of data to learn knowledge bases in description logics that can handle uncertainty explicitly. The present contribution is a step in this direction.

Acknowledgements. The third author is partially supported by CNPq. The work reported here has received substantial support through FAPESP grant 2008/03995-5.

References

1. Antoniou, G., van Harmelen, F.: Semantic Web Primer. MIT Press (2008)
2. Baader, F., Nutt, W.: Basic description logics. In: Description Logic Handbook, pp. 47–100. Cambridge University Press (2002)
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. *Web Semant.* 7(3), 154–165 (2009)
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press (2001)
5. Costa, P.C.G., Laskey, K.B.: PR-OWL: A framework for probabilistic ontologies. In: Proceeding of the 2006 Conference on Formal Ontology in Information Systems, pp. 237–249. IOS Press, Amsterdam (2006)
6. Cozman, F.G., Polastro, R.B.: Loopy Propagation in a Probabilistic Description Logic. In: Greco, S., Lukasiewicz, T. (eds.) SUM 2008. LNCS (LNAI), vol. 5291, pp. 120–133. Springer, Heidelberg (2008)
7. Cozman, F.G., Polastro, R.B.: Complexity analysis and variational inference for interpretation-based probabilistic description logics. In: Conference on Uncertainty in Artificial Intelligence (2009)
8. d'Amato, C., Fanizzi, N., Lukasiewicz, T.: Tractable Reasoning with Bayesian Description Logics. In: Greco, S., Lukasiewicz, T. (eds.) SUM 2008. LNCS (LNAI), vol. 5291, pp. 146–159. Springer, Heidelberg (2008)
9. Fanizzi, N., d'Amato, C., Esposito, F.: DL-FOIL Concept Learning in Description Logics. In: Železný, F., Lavrač, N. (eds.) ILP 2008. LNCS (LNAI), vol. 5194, pp. 107–121. Springer, Heidelberg (2008)
10. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Machine Learning* 29, 131–163 (1997)
11. Heinsohn, J.: Probabilistic description logics. In: International Conf. on Uncertainty in Artificial Intelligence, pp. 311–318 (1994)
12. Iannone, L., Palmisano, I., Fanizzi, N.: An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence* 26(2), 139–159 (2007)
13. Jaeger, M.: Probabilistic reasoning in terminological logics. In: Principals of Knowledge Representation (KR), pp. 461–472 (1994)
14. Landwehr, N., Kersting, K., DeRaedt, L.: Integrating Naïve Bayes and FOIL. *J. Mach. Learn. Res.* 8, 481–507 (2007)
15. Lavrac, N., Dzeroski, S.: Inductive Logic Programming: Techniques and Applications. Ellis Horwood, New York (1994)
16. Lehmann, J.: Hybrid Learning of Ontology Classes. In: Perner, P. (ed.) MLD 2007. LNCS (LNAI), vol. 4571, pp. 883–898. Springer, Heidelberg (2007)
17. Lehmann, J., Hitzler, P.: Foundations of Refinement Operators for Description Logics. In: Blockeel, H., Ramon, J., Shavlik, J.W., Tadepalli, P. (eds.) ILP 2007. LNCS (LNAI), vol. 4894, pp. 161–174. Springer, Heidelberg (2008)
18. Lehmann, J., Hitzler, P.: A Refinement Operator Based Learning Algorithm for the \mathcal{ALC} Description Logic. In: Blockeel, H., Ramon, J., Shavlik, J.W., Tadepalli, P. (eds.) ILP 2007. LNCS (LNAI), vol. 4894, pp. 147–160. Springer, Heidelberg (2008)
19. Lukasiewicz, T.: Expressive probabilistic description logics. *Artif. Intell.* 172(6–7), 852–883 (2008)

20. Ochoa-Luna, J., Cozman, F.G.: An algorithm for learning with probabilistic description logics. In: Bobillo, F., et al. (eds.) Proceedings of the 5th International Workshop on Uncertainty Reasoning for the Semantic Web, Chantilly, USA, vol. 527, pp. 63–74. CEUR-WS.org (2009)
21. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: networks of plausible inference. Morgan Kaufmann (1988)
22. Polastri, R.B., Cozman, F.G.: Inference in probabilistic ontologies with attributive concept descriptions and nominals. In: 4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW) at the 7th International Semantic Web Conference (ISWC), Karlsruhe, Germany (2008)
23. Quinlan, J.R., Cameron-Jones, R.M.: FOIL: A midterm report. In: Proceedings of the European Conference on Machine Learning, pp. 3–20. Springer, Heidelberg (1993)
24. Revoredo, K., Ochoa-Luna, J.E., Cozman, F.G.: Learning Terminologies in Probabilistic Description Logics. In: da Rocha Costa, A.C., Vicari, R.M., Tonidandel, F. (eds.) SBIA 2010. LNCS, vol. 6404, pp. 41–50. Springer, Heidelberg (2010)
25. Sebastiani, F.: A probabilistic terminological logic for modelling information retrieval. In: ACM Conf. on Research and Development in Information Retrieval (SIGIR), pp. 122–130 (1994)
26. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW 2007: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706. ACM, New York (2007)

Belief Merging Using Normal Forms

Pilar Pozos-Parra¹, Laurent Perrussel², and Jean Marc Thevenin²

¹ Department of Informatics and Systems, University of Tabasco,
Tabasco - Mexico

pilar.pozos@ujat.mx

² Institut de Recherche en Informatique de Toulouse, Université Toulouse I,
Toulouse - France
{laurent.perrussel,jean-marc.thevenin}@univ-tlse1.fr

Abstract. Belief merging aims to conciliate multiple possibly inconsistent belief bases into a consistent common belief base. To handle inconsistency some operators have been proposed. Most of them do not consider inconsistent bases. *PS-Merge* is an alternative method of merging that uses the notion of Partial Satisfiability and allows us to take into account inconsistent bases. *PS-Merge* needs the bases represented as DNF formulas, nevertheless, many practical problems are easily represented in its CNF. The aim of this paper is to extend the notion of Partial Satisfiability in order to consider bases represented as CNF formulas. Moreover, we consider Prime Normal forms in order to define a method that allows us to implement *PS-Merge* for difficult theories. We also show that once the belief bases are represented as sets of normal forms, *PS-Merge* is polynomial.

1 Introduction

Belief merging is concerned with the process of aggregate the information contained in a set of (possibly inconsistent) belief bases obtained from different sources to produce a single consistent belief base. Several merging operators have been defined and characterized with the help of four components [9]: a notion of distance between models, a notion of distance between a model and a formula, a notion of distance between a model and a set of formulas and a process of minimization. Usually, this model-based merging operators only take into account consistent belief bases and consequently the information in the base, which is not responsible for conflicts is lost.

PS-Merge is an alternative method of merging that uses the notion of Partial Satisfiability [1,12]. *PS-Merge* is not based on classical distance measures on models, it takes into account inconsistent bases and provides results similar to classical merging operators. Moreover, *PS-Merge* has been implemented in Matlab language [2], the experimental evaluation uses toy examples from the literature. Others experimental assessment of algorithms for merging operators uses random belief bases as a way of evaluating performance [5,4]. There is, generally, no accepted method for evaluating belief merging algorithms, neither a library of

standard belief merging problems. Therefore, we need to borrow proposed framework for evaluating the belief merging algorithms. In order to create a common formulation for comparison we use two possible candidates that can be seen as a sort of merging: the propositional satisfiability problem (SAT), which has a set of benchmark problems at <http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html> and the course timetabling problem, which has a set of problem instances at <http://www.cs.qub.ac.uk/itc2007/Login/SecretPage.php>. Both candidates can be expressed in logic under a conjunctive normal form (CNF). Given that *PS-Merge* accepts inputs bases only in disjunctive normal form (DNF), we propose an extension of the notion of Partial Satisfiability to accept both forms: CNF and DNF. An implementation for CNF is provided and some SAT problems has been solved, however, as the number of atoms increases, the proposed implementation fails. We then propose a new definition of belief merging operator in terms of prime implicants, even when we need to restrain the inputs bases to that ones which are consistent, we can avoid the problem of dependency of syntax that *PS-Merge* has.

The rest of the paper is structured as follows. After providing some technical preliminaries, Section 3 describes the extension of Partial Satisfiability. Section 4 gives a comparison of *PS-Merge* with other approaches. Section 5 provides the characterization in terms of postulates. Section 6 shows the implementation. Section 7 describes a process of merging in terms of prime implicants and Section 8 concludes with a discussion of open issues.

2 Preliminaries

We consider a language \mathcal{L} of propositional logic using a finite ordered set of symbols $P := \{p_1, p_2, \dots, p_n\}$. There are algorithms to convert a formula $\phi \in \mathcal{L}(P)$ into disjunctive normal form (DNF_ϕ), such that $\phi \equiv \text{DNF}_\phi$ [11]. A disjunctive normal form of ϕ is a disjunction of terms $\text{DNF}_\phi = D_1 \vee \dots \vee D_m$, which each term D_i is a conjunction of literals $D_i = l_1 \wedge \dots \wedge l_k$, with $l_i = p_j$ or $l_i = \neg p_j$. A term D is an implicant of a formula ψ iff $D \models \psi$, and it is a prime implicant iff for all implicants D' of ψ such that $D \models D'$, we have $D' \models D$. We define IP_ψ as a disjunction of prime implicants of ψ such that $\psi \equiv IP_\psi$. Also there are algorithms to convert a formula $\phi \in \mathcal{L}(P)$ into conjunctive normal form (CNF_ϕ), such that $\phi \equiv \text{CNF}_\phi$ [11]. A conjunctive normal form of ϕ is a conjunction of clauses $\text{CNF}_\phi = C_1 \wedge \dots \wedge C_m$, which each clause C_i is a disjunction of literals $C_i = l_1 \vee \dots \vee l_k$, with $l_i = p_j$ or $l_i = \neg p_j$. A clause C is an implicant of a formula ψ iff $\psi \models C$, and it is a prime implicant of ψ iff for all implicants C' of ψ such that $C' \models C$, we have $C \models C'$. We define PI_ψ as a conjunction of prime implicants of ψ such that $\psi \equiv PI_\psi$.

A *belief base* K is a finite set of propositional formulas of \mathcal{L} representing the beliefs from a source (we identify K with the conjunction of its elements).

The set of models of the language is denoted by \mathcal{W} , its elements will be denoted by vectors of the form $(w(p_1), \dots, w(p_n))$ and the set of models of a formula ϕ is denoted by $\text{mod}(\phi)$. K is consistent iff there exists model of K . If ϕ

is a propositional formula or a set of propositional formulas then $\mathcal{P}(\phi)$ denotes the set of atoms appearing in ϕ . $|P|$ denotes the cardinality of set P . A *literal* is an atom or its negation.

A belief *profile* $E = \{K_1, \dots, K_m\}$ is a multiset (bag) of m belief bases. Two belief profiles E_1 and E_2 are said to be equivalent, denoted by $E_1 \equiv E_2$, iff there is a bijection g from E_1 to E_2 such that $K \equiv g(K)$ for every base K in E_1 . With $\bigwedge E$ we denote the conjunction of the belief bases $K_i \in E$, while \sqcup denotes the multiset union. For every belief profile E and positive integer n , E^n denotes the multiset union of n times E .

PS-Merge. considers only normalized languages so that each belief base is taken as the disjunctive normal form (DNF) of the conjunction of its elements. Thus if K is a belief base we identify it with $DNF(K)$, and we refer to it simply as K . The DNF of a formula is obtained by replacing $\phi \leftrightarrow \psi$ and $\phi \rightarrow \psi$ by $(\neg\phi \vee \psi) \wedge (\neg\psi \vee \phi)$ and $\neg\phi \vee \psi$ respectively, applying De Morgan's laws, using the distributivity law, distributing \vee over \wedge and finally eliminating the literals repeated in each conjunct.

Example 1. Given the belief base $K = \{a \rightarrow b, \neg c\}$ it is identified with $K = (\neg a \wedge \neg c) \vee (b \wedge \neg c)$.

Definition 1 (Partial Satisfiability). Let K be a belief base, $w \in \mathcal{W}$ and $|P| = n$, the Partial Satisfiability of K for w , denoted as $w_{ps}(K)$, is defined as:

- If $K = l_1 \wedge \dots \wedge l_s$ where l_i are literals then

$$w_{ps}(K) = \max \left\{ \sum_{i=1}^s \frac{w(l_i)}{s}, \frac{n - |\mathcal{P}(K)|}{2n} \right\}.$$

- If $K := D_1 \vee \dots \vee D_r$ where each D_i is a literal or a conjunction of literals then

$$w_{ps}(K) = \max \{w_{ps}(D_1), \dots, w_{ps}(D_r)\}.$$

Example 2. The Partial Satisfiability of the belief base of Example 1 given $P = \{a, b, c\}$ and $w = (1, 1, 1)$ is

$$w_{ps}(K) = \max \left\{ \max \left\{ \frac{w(\neg a) + w(\neg c)}{2}, \frac{1}{6} \right\}, \max \left\{ \frac{w(b) + w(\neg c)}{2}, \frac{1}{6} \right\} \right\} = \frac{1}{2}.$$

PS-Merge elects the models whose values maximize the sum of the Partial Satisfiability of the bases.

Definition 2. Let $E = \{K_1, \dots, K_m\}$ be a belief profile and *PS-Merge* be a function which maps a belief profile to a belief base, *PS-Merge*: $\mathcal{L}(P)^n \rightarrow \mathcal{L}(P)$, then the Partial Satisfiability Merge of E is *PS-Merge*(E) such that the set of models of the resulting base is:

$$\left\{ w \in \mathcal{W} \mid \sum_{i=1}^m w_{ps}(K_i) \geq \sum_{i=1}^m w'_{ps}(K_i) \text{ for all } w' \in \mathcal{W} \right\}.$$

Example 3. If we have $K_1 = (s \wedge \neg d) \vee (o \wedge \neg d)$, $K_2 = (\neg s \wedge d \wedge \neg o) \vee (\neg s \wedge \neg d \wedge o)$, and $K_3 = s \wedge d \wedge o$. As we can see in the fifth column of Table 1 the models of $PS\text{-Merge}(E)$ ¹ are the models $(0, 0, 1)$ and $(1, 0, 1)$.

Table 1. $PS\text{-Merge}(E)$ using Partial Satisfiability

<i>w</i>	K_1	K_2	K_3	<i>Sum</i>
$(1, 1, 1)$	$\frac{1}{2}$	$\frac{1}{3}$	1	$\frac{11}{6} \simeq 1.83$
$(1, 1, 0)$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{11}{6} \simeq 1.83$
$(1, 0, 1)$	1	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{14}{6} \simeq 2.33$
$(1, 0, 0)$	1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{10}{6} \simeq 1.67$
$(0, 1, 1)$	$\frac{1}{2}$	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{11}{6} \simeq 1.83$
$(0, 1, 0)$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{6}{6} = 1.5$
$(0, 0, 1)$	1	1	$\frac{1}{3}$	$\frac{14}{6} \simeq 2.33$
$(0, 0, 0)$	$\frac{1}{2}$	$\frac{2}{3}$	0	$\frac{7}{6} \simeq 1.16$

Usually, classical merging operators only take into account consistent belief bases but $PS\text{-Merge}$ allows us to merge even inconsistent bases as we can see in the following example.

Example 4. Let be $K_1 = \{a \rightarrow b, \neg c\}$ and $K_2 = \{a \rightarrow \neg b, a, b\}$ two bases, the second base is inconsistent but $PS\text{-Merge}$ allows us to merge these bases, in fact $PS\text{-Merge}(\{K_1, K_2\}) = (1, 1, 0)$.

If we analyze the profile we see that the first base has three different models $(1,1,0), (0,1,0)$ and $(0,0,0)$ which means that K_1 is “uncertain” about a and b , the second base has no model, but both together they can give us a compromise between the “uncertainty” of K_1 and the “inconsistency” of K_2 which results intuitive.

3 Normal Partial Satisfiability

In this section we propose an extension of Partial Satisfiability. We consider a normalized language so that the belief bases are taken as DNF or CNF. The DNF is obtained as before and the CNF of a formula is obtained by replacing $\phi \leftrightarrow \psi$ and $\phi \rightarrow \psi$ by $(\neg\phi \vee \psi) \wedge (\neg\psi \vee \phi)$ and $\neg\phi \vee \psi$ respectively, applying De Morgan’s laws, using the distributivity law, distributing \wedge over \vee and finally eliminating the literals repeated in each conjunct.

Example 5. Given the belief base $K = \{a \rightarrow b, \neg c\}$ it is identified with $K = (\neg a \vee b) \wedge \neg c$.

¹ If Δ is a merging operator, we are going to abuse the notation by referring to the models of the merging operator $mod(\Delta(E))$ and their respective belief base $\Delta(E)$ simply as $\Delta(E)$.

The last part of the construction of the CNF (the minimization by eliminating literals) is important since the number of literals in each disjunct affects the satisfaction degree of the disjunct. We are not applying other logical minimization methods to reduce the size of the CNF expressions since this may affect the intuitive meaning of the formulas.

Actually we generalize the definition of valuation given in [13], where given $w \in \mathcal{W}$, we can define a valuation of sentence ϕ as a mapping $w : \mathcal{L}(P) \rightarrow \{0, 1\}$ if $w(\phi \vee \psi) = \max(w(\phi), w(\psi))$, $w(\phi \wedge \psi) = \min(w(\phi), w(\psi))$ and $w(\neg\phi) = 1 - w(\phi)$. In general, a valuation for disjunction and conjunction is defined as follows: $w(\phi_1 \vee \phi_2 \vee \dots \vee \phi_n) = \max(w(\phi_1), w(\phi_2), \dots, w(\phi_n))$, $w(\phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_n) = \min(w(\phi_1), w(\phi_2), \dots, w(\phi_n))$.

Definition 3 (Normal Partial Satisfiability). Let $K \in \mathcal{L}(P)$ in DNF or CNF, $w \in \mathcal{W}$, we define the Normal Partial Satisfiability of K for w , denoted as $w_{ps}(K)$, as follows:

- If $K \in P$

$$w_{ps}(K) = w(K)$$

- If $K := \neg p$

$$w_{ps}(K) = 1 - w_{ps}(p)$$

- If $K := D_1 \vee \dots \vee D_n$

$$w_{ps}(K) = \max \{w_{ps}(D_1), \dots, w_{ps}(D_n)\}$$

- If $K := C_1 \wedge \dots \wedge C_n$

$$w_{ps}(K) = \sum_{i=1}^n \frac{w_{ps}(C_i)}{n}$$

The valuation of a conjunction is the average of literals satisfied, it can be seen as a partial support to the models that provide bigger average. We assume that it is better to be partially satisfied than not being satisfied at all, even if the choice is not exactly what you wanted. The rest of connectives are valued as usual. Then $PS\text{-Merge}(E)$ given in Definition 2 can be used for the merging process.

Example 6. Consider $K_1 = (s \vee o) \wedge \neg d$, $K_2 = \neg s \wedge (\neg d \vee \neg o) \wedge (d \vee o)$, and $K_3 = s \wedge d \wedge o$ introduced in example 3 as DNFs. As we can see in the fifth column of Table 2 the models of $PS\text{-Merge}(E)$ are the same as in example 3, $(0, 0, 1)$ and $(1, 0, 1)$.

4 Comparing Results

$PS\text{-Merge}$ using Normal Partial Satisfiability and Partial Satisfiability yields the same results and it yields similar results compared to existing techniques

Table 2. $PS\text{-Merge}(E)$ using Normal Partial Satisfiability

w	K_1	K_2	K_3	Sum
(1, 1, 1)	$\frac{1}{2}$	$\frac{1}{3}$	1	$\frac{11}{6} \simeq 1.83$
(1, 1, 0)	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{11}{6} \simeq 1.83$
(1, 0, 1)	1	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{14}{6} \simeq 2.33$
(1, 0, 0)	1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{10}{6} \simeq 1.67$
(0, 1, 1)	$\frac{1}{2}$	$\frac{3}{3}$	$\frac{2}{3}$	$\frac{11}{6} \simeq 1.83$
(0, 1, 0)	0	1	$\frac{1}{3}$	$\frac{4}{3} = 1.33$
(0, 0, 1)	1	1	$\frac{1}{3}$	$\frac{14}{6} \simeq 2.33$
(0, 0, 0)	$\frac{1}{2}$	$\frac{2}{3}$	0	$\frac{7}{6} \simeq 1.16$

such as $CMerge$, the Δ_{Σ}^2 operator and MCS (Maximal Consistent Subsets) considered in [10,6,7]. Let E be in each case the belief profile consisting of the belief bases enlisted below and let P be corresponding set of atoms ordered alphabetically.

Table 3. Comparison of $PS\text{-Merge}(E)$ and other approaches

	E	$MCS(E)$	$CMerge(E)$	$\Delta_{\Sigma}(E)$	$PS\text{-Merge}(E)$
1	$K_1 = K_2 = \{a\}$ $K_3 = \{\neg a\}$	$\{a \vee \neg a\}$	$\{a\}$	$\{(1)\}$	$\{(1)\}$
2	$K_1 = \{b\}$ $K_2 = \{a, a \rightarrow b\}$ $K_3 = \{\neg b\}$	$\{a, a \rightarrow b, b\} \vee \{a, \neg b\} \vee \{\neg b, a \rightarrow b\}$	$\{a, a \rightarrow b\}$	$\{(1, 1)\}$	$\{(1, 1)\}$
3	$K_1 = \{b\}$ $K_2 = \{a, b\}$ $K_3 = \{\neg b\}$	$\{a, b\} \vee \{a, \neg b\}$	$\{a, a \rightarrow b\}$	$\{(1, 1)\}$	$\{(1, 1)\}$
4	$K_1 = K_2 = \{a \rightarrow b\}$ $K_3 = \{a\}$ $K_4 = \{a, \neg b\}$	$\{a, a \rightarrow b\} \vee \{a, \neg b\} \vee \{\neg b, a \rightarrow b\}$	$\{a, a \rightarrow b\}$	$\{(1, 1)\}$	$\{(1, 1)\}$
5	$K_1 = \{a\}$ $K_2 = \{a \rightarrow b\}$ $K_3 = \{a, \neg b\}$	$\{a, a \rightarrow b, b\} \vee \{a, \neg b\} \vee \{\neg b, a \rightarrow b\}$	$\{a\}$	$\{(1, 1), (1, 0)\}$	$\{(1, 1)\}$
6	$K_1 = \{a, c\}$ $K_2 = \{a \rightarrow b, \neg c\}$ $K_3 = \{b \rightarrow d, c\}$	$\{a, a \rightarrow b, b \rightarrow d\}$	$\{a, a \rightarrow b, b \rightarrow d, c\}$	$\{(1, 1, 1, 1)\}$	$\{(1, 1, 1, 1)\}$
7	$K_1 = \{a, c\}$ $K_2 = \{a \rightarrow b, \neg c\}$ $K_3 = \{b \rightarrow d, c\}$ $K_4 = \{\neg c\}$	$\{a, a \rightarrow b, b \rightarrow d\}$	$\{a, a \rightarrow b, b \rightarrow d\}$	$\{(1, 1, 0, 1), (1, 1, 1, 1)\}$	$\{(1, 1, 0, 1)\}$
8	$K_1 = \{a\}$ $K_2 = \{a \rightarrow b\}$ $K_3 = \{a, \neg b\}$ $K_4 = \{\neg b\}$	$\{a, a \rightarrow b, b\} \vee \{a, \neg b\} \vee \{\neg b, a \rightarrow b\}$	$\{a \wedge \neg b\}$	$\{(1, 0)\}$	$\{(1, 0)\}$
9	$K_1 = \{b\}$ $K_2 = \{a \rightarrow b\}$ $K_3 = \{a, \neg b\}$	$\{a, a \rightarrow b, b\} \vee \{a, \neg b\} \vee \{\neg b, a \rightarrow b\}$	$\{a \wedge b\}$	$\{(1, 1)\}$	$\{(1, 1)\}$
10	$K_1 = \{b\}$ $K_2 = \{a \rightarrow b\}$ $K_3 = \{a, \neg b\}$ $K_4 = \{\neg b\}$	$\{a, a \rightarrow b, b\} \vee \{a, \neg b\} \vee \{\neg b, a \rightarrow b\}$	$\{a \vee \neg b\}$	$\{(0, 0), (1, 0), (1, 1)\}$	$\{(1, 1), (0, 0)\}$

5 Postulates

In [6,8] Konieczny and Pino-Pérez proposed the basic postulates (A1)-(A6), rephrased without reference to integrity constraints:

² As modeled in [6], merging operator Δ_{Σ} is equivalent to the merging operator proposed by Lin and Mendelzon in [10] called $CMerge$.

Definition 4. Let E, E_1, E_2 be belief profiles, K_1 and K_2 be consistent belief bases. Δ be an operator which assigns to each belief profile E a belief base $\Delta(E)$. Δ is a merging operator if and only if it satisfies the following postulates:

- (A1) $\Delta(E)$ is consistent
- (A2) if $\bigwedge E$ is consistent then $\Delta(E) \equiv \bigwedge E$
- (A3) if $E_1 \equiv E_2$, then $\Delta(E_1) \equiv \Delta(E_2)$
- (A4) $\Delta(\{K_1, K_2\}) \wedge K_1$ is consistent if and only if $\Delta(\{K_1, K_2\}) \wedge K_2$ is consistent
- (A5) $\Delta(E_1) \wedge \Delta(E_2) \models \Delta(E_1 \sqcup E_2)$
- (A6) if $\Delta(E_1) \wedge \Delta(E_2)$ is consistent, then $\Delta(E_1 \sqcup E_2) \models \Delta(E_1) \wedge \Delta(E_2)$

The postulates describe the principles that a belief merging operator should satisfy. Among them, minimal change (A2), syntax irrelevance (A3), and fairness (A4) are key postulates. (A1) ensures the extraction of information from the profile.

We analyze the minimal set of properties *PS-Merge* satisfies. Clearly *PS-Merge* satisfies (A1), which simply requires of the result of merging to be consistent. *PS-Merge* satisfies (A2).

Proposition 1. $\bigwedge E \not\models \perp$ implies $PS\text{-Merge}(E) \equiv \bigwedge E$.

Proof. Let $E = \{K_1, \dots, K_m\}$ a profile with its beliefs bases expressed in DNF, such that $\bigwedge E \not\models \perp$. There are $l > 0$ models w_1, \dots, w_l that satisfy each base thus for every model w_r we can find m disjoints d_1, \dots, d_m belonging to each base K_1, \dots, K_m respectively, that are satisfied by w_r consequently the Normal Partial Satisfiability of the bases for every w_r is valued in 1, i.e. $w_{rps}(K_j) = 1$ for $1 \leq r \leq l$ and $1 \leq j \leq m$. So we can affirm that $\sum_{i=1}^m w_{rps}(K_i) = m$ for each model of the profile. Notice that every disjoint has the values $\sum_{i=1}^n \frac{w(C_i)}{n}$ (see Definition 3), moreover it is less or equal to 1, from this fact we can affirm that if a model w does not satisfied a base K then $w_{ps}(K) < 1$ and we can conclude that $\sum_{i=1}^m w_{ps}(K_i) < m$ for the models that does not satisfy the profile. Hence a model w is included in the merge iff w is a model of $\bigwedge E$, i.e. we obtain only models of the conjunction of the bases as a result of *PS-Merge* when the profile is consistent.

If the profile is expressed in CNF, such that $\bigwedge E \not\models \perp$. There are $l > 0$ models w_1, \dots, w_l that satisfy each base thus for every model w_r , as before we find that Normal Partial Satisfiability of the bases for every w_r is valued in 1, i.e. $w_{rps}(K_j) = 1$ for $1 \leq r \leq l$ and $1 \leq j \leq m$. So we can affirm that $\sum_{i=1}^m w_{rps}(K_i) = m$ for each model of the profile. Notice that every base has the values $\sum_{i=1}^n \frac{w(C_i)}{n}$, which it is less or equal to 1, therefore we can affirm that if a model w does not satisfied a base K then $w_{ps}(K) < 1$ and we can conclude that $\sum_{i=1}^m w_{ps}(K_i) < m$ for the models that does not satisfy the profile. Hence a model w is included in the merge iff w is a model of $\bigwedge E$, i.e. we obtain only models of the conjunction of the bases as a result of *PS-Merge* when the profile is consistent.

If there is not “redundant” information, i.e. formulas including disjoints of the style $a \wedge \neg a$ in a DNF, formulas including conjoints of the style $a \vee \neg a$ in a CNF, then (A3) and (A4) are satisfied. Also *PS-Merge* satisfies (A5) and (A6).

Remark 1. By definition the *PS-Merge* is commutative i.e. the result of the merging does not depend on any order of the bases of the profile.

There are two important classes of merging operators, majority and arbitration operators. The behavior of majority operators is to say that if an opinion has a large audience, then it will be the opinion of the group. A postulate that captures this idea is the postulate (M7) of [6].

$$(M7) \forall K \exists n \in \mathbb{N} \quad \Delta(E \sqcup \{K\}^n) \models K$$

PS-Merge satisfies the postulate (M7).

6 Algorithm PS – Merge

We can easily modify the algorithm proposed in [2] to consider CNF if the belief profile is given as follows:

- V : Number of variables of E
- B : Number of bases of E
- C : Vector of number of conjuncts of each base in E
- L : Matrix of occurrences of literals by each conjunct of each base of E

Example 7. Consider the profile of example 6, with $K_1 = (s \vee o) \wedge \neg d$, $K_2 = \neg s \wedge (\neg d \vee \neg o) \wedge (d \vee o)$, and $K_3 = s \wedge d \wedge o$ then we format the profile E as:

- $V = 3$
- $B = 3$
- $C = (2, 3, 3)$

$$- L = \begin{pmatrix} s & \neg s & d & \neg d & o & \neg o \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The algorithm has been implemented in Matlab and tested with all the examples presented in this article with an average case performance of 0.4 sec, and some other with a greater number of variables for example 15 variables with 35 bases and a total of 140 conjuncts with an average case performance of 1500 sec. The results indicate that problems of modest size can be treated using commodity hardware and short computation times. The complexity is polynomial.

Algorithm 1. *PS-Merge*

Data: V : Number of variables of E B : Number of bases of E D : Vector of number of conjuncts of each base in E L : Matrix of occurrences of literals by each conjunct of each base of E **Result:** $Solution_Set$: The set of models in $PS\text{-Merge}(E)$ **begin**

```

    Solution  $\leftarrow \emptyset$ 
    Max-Sum  $\leftarrow 0$ 
     $W \leftarrow$  Matrix whose rows are all the possible models for  $V$  variables
    for  $s = 1 \dots B$  do
         $IC_s \leftarrow \sum_{k=1}^{s-1} C_k + 1$ 
        for  $i = 1 \dots 2^V$  do
            Sum  $\leftarrow 0$ 
            for  $s = 1 \dots B$  do
                ps-conjunct  $\leftarrow 0$ 
                for  $c = IC_s \dots IC_s + C_s$  do
                    satisfied  $\leftarrow 0$ 
                    for  $j = 1 \dots V$  do
                        if  $W_{i,j} = 1$  then
                            satisfied  $\leftarrow L_{c,2j-1}$ 
                        if  $W_{i,j} = 0$  then
                            satisfied  $\leftarrow L_{c,2j}$ 
                    ps-conjunct  $\leftarrow ps\text{-conjunct} + satisfied$ 
                PS  $\leftarrow \frac{ps\text{-conjunct}}{C_s}$ 
                Sum  $\leftarrow Sum + PS$ 
            if  $Sum > MaxSum$  then
                Solution  $\leftarrow \{i\}$ 
                MaxSum  $\leftarrow Sum$ 
            else if  $Sum = MaxSum$  then
                Solution  $\leftarrow Solution \cup \{i\}$ 
    Solution_Set = {ith-row of  $W | i \in Solution$ }
end

```

7 Prime Implicant-Based Merging

Due to huge amount of variables needed for complex problems such as the timetabling problem designed for the First International Timetabling Competition (ITC)³, the MATLAB implementation can not be used since it run out of memory space.

The problem consists of finding an optimal timetable within the following framework: there is a set of events $E = \{E_1, E_2, \dots, E_{n_E}\}$ to be scheduled in a set of rooms $R = \{R_1, R_2, \dots, R_{n_R}\}$, where each room has 45 available timeslots, nine for each day in a five day week. There is a set of students $S = \{S_1, S_2, \dots, S_{n_S}\}$ who attend the events, and a set of features $F = \{F_1, F_2, \dots, F_{n_F}\}$ satisfied by rooms and required by events. Each event is attended by a number of students, and each room has a given size, which is the maximum number of students the room can accommodate. A feasible timetable is one in which all events have been assigned a timeslot and a room so that the following hard constraints are satisfied: no student attends more than one event at the same time; the room is big enough for all the attending students and satisfies all the features required by the event; and only one event is scheduled in each room at any timeslot. The problem may be precisely formulated as:

- let $\{F_1, F_2, \dots, F_{n_F}\}$ be a set of variables representing the features;
- $R_i = \{F'_1, F'_2, \dots, F'_{n_{R_i}}\}$ where $F'_i \in F$ for $i = 1, \dots, n_{R_i}$ and n_{R_i} is the number of feature satisfied by room R_i ;
- $\{N_1, \dots, N_{n_R}\}$ be a set of integer numbers indicating the maximum of students each room can accommodate;
- $E_i = \{F''_1, F''_2, \dots, F''_{n_{E_i}}\}$ where $F''_i \in F$ for $i = 1, \dots, n_{E_i}$ and n_{E_i} is the number of feature required by event E_i ;
- $S_i = \{E'_1, E'_2, \dots, E'_{n_{S_i}}\}$ where $E'_i \in E$ for $i = 1, \dots, n_{S_i}$ and n_{S_i} is the number of events student S_i attends; and
- $T = \{T_1, \dots, T_{45}\}$ be a set of timeslots.

Find a feasible solution, i.e. a set of pairs $\{(T'_1, R'_1), \dots, (T'_{n_E}, R'_{n_E})\}$ such that:

- $T'_i \in T$ and $R'_i \in R$
- $\neg(T'_i = T'_j)$ if $E_i \in S_k$ and $E_j \in S_k$ and $\neg(i = j)$
- $E_i \subseteq R'_i$ and $|\{S_j | j = 1, \dots, n_S \text{ and } E_i \in S_j\}| \leq N_k$ where $R'_i = R_k$;
- $\neg \exists i \exists j (\neg(i = j) \wedge T'_i = T'_j \wedge R'_i = R'_j)$.

The following actions are performed in order to formulate the problem in terms of merging:

- the number of students for each event is calculated and stored,
 $n_i = |\{S_j | j = 1, \dots, n_S \text{ and } E_i \in S_j\}|$
- a list of possible rooms is created for each event,
 $e_i = \{R_j | E_i \subseteq R_j \text{ and } n_i \leq N_j\}$

³ <http://www.idsia.ch/Files/ttcomp2002/>

The actions allows us to eliminate the feature set, event set and capacity requirements, defining a new event set $\{e_1, \dots, e_{n_E}\}$ that includes the eliminated information. The new event set will be used for satisfying hard constraint (2).

The parameters to be used for the *PS-Merge* operator as follows:

- create the propositional variables set $V = \{R_{i,s,r} | R_r \in e_i, s = 1, \dots, 45 \text{ and } i = 1, \dots, n_E\}$,
- for every event e_i create the belief source $K_i = \bigvee_{R_{i,s,r} \in V} R_{i,s,r}$,
- create the source $K_{n_E+1} = \bigwedge_{R_{i_1,s,r}, R_{i_2,s,r} \in V} R_{i_1,s,r} \wedge \neg(i_1 = i_2) \rightarrow \neg R_{i_2,s,r}$.
- create the source $K_{n_E+2} = \bigwedge_{R_{i_1,s,r_1}, R_{i_2,s,r_2} \in V, j \in \{1, \dots, n_S\}} R_{i_1,s,r_1} \wedge \neg(i_1 = i_2) \wedge E_{i_1}, E_{i_2} \in S_j \rightarrow \neg R_{i_2,s,r_2}$.

The problem can be solve by maximizing the satisfaction of the belief sources as follows: $PS\text{-Merge}(\{K_1, \dots, K_{n_E+2}\})$. Theoretically the problem is solved, however, in practice the implementation is insufficient. And then we propose to use Prime Implicant Form in order to improve the merging method.

Let $E = \{K_1, \dots, K_m\}$ given by $IP_E = \{IP_{K_1}, \dots, IP_{K_m}\}$. Then we can extend each term $D_{K_i}^k$ in IP_{K_i} with all the contradictory literal of each term $D_{K_j}^l$ belonging to the base IP_{K_j} : $\Gamma = \{D | D = D_{K_i}^k \cup (\bigcup_{j=1, j \neq i}^m D_{K_j}^l - \bar{D}_{K_i}^k)\}$.

The contradictory literals between terms D_1 and D_2 are defined as:

$$\gamma(D_1, D_2) = D_1 \cap \bar{D}_2.$$

IP-Merge(E) consists of finding the models that are the closest to the models of the initial belief bases. In this context, the merging using prime implicants consists of choosing the terms in Γ that have the smallest set of contradictory literals i.e. *IP-Merge*(E) is an image of a function from the belief profiles to belief bases such that the set of models of the resulting base is:

$$\left\{ D \in \Gamma \mid \min(|\gamma(D_{K_i}^k, \bigcup_{j=1, j \neq i}^m D_{K_j}^l)|) 1 \leq i \leq m \right\}.$$

8 Conclusion

PS-Merge operator has been initially proposed in [12]; as opposed to all other merging techniques, its definition is not base on a classical distance between models but on a notion of partial satisfiability. *PS-Merge* resolves conflicts among the belief bases in a natural way and does not need to calculate a partial pre-order over the set of models since Partial Satisfiability can be calculated for a single model. Then the implementation is possible using advantages of hight level of language programming [2]. However, it is necessary to take into account that before calculating the Partial Satisfiability of a formula it must be transformed into DNF. In the literature we can find many real-world collective decision problems that can be expressed in a CNF. Then we propose an extension of *PS-Merge* in order to consider CNF. We introduces the notion of Normal Partial Satisfiability in order to considers both kind of normal forms. The implementation has

been extended and we obtain the same results for all the examples provided in [12,2]. Unlike other approaches *PS-Merge* can consider belief bases which are inconsistent, since the source of inconsistency can refer to specific atoms and the operator takes into account the rest of the information. Moreover, given the problem of computation for complex problem we survey a new merging based on the notion of prime implicant, which requires less computation. In future work, we would like to implement this new belief merging operator.

References

1. Borja Macías, V., Pozos Parra, P.: Model-based belief merging without distance measures. In: Proceedings of the Sixth International Conference on Autonomous Agents and Multiagent Systems, Hawai'i, USA, pp. 613–615. IFAAMAS (2007)
2. Macías, V.B., Parra, P.P.: Implementing *PS-merge* Operator. In: Aguirre, A.H., Borja, R.M., Garciá, C.A.R. (eds.) MICAI 2009. LNCS, vol. 5845, pp. 39–50. Springer, Heidelberg (2009)
3. Everaere, P., Konieczny, S., Marquis, P.: The strategy-proofness landscape of merging. *Journal of Artificial Intelligence Research* 28, 49–105 (2007)
4. Gorogiannis, N., Hunter, A.: Implementing semantic merging operators using binary decision diagrams. *Int. J. Approx. Reasoning* 49(1), 234–251 (2008)
5. Hue, J., Papini, O., Wurbel, E.: Syntactic propositional belief bases fusion with removed sets. In: Symbolic and Quantitative Approaches to Reasoning and Uncertainty, pp. 66–77 (2007)
6. Konieczny, S., Pino-Pérez, R.: On the logic of merging. In: Cohn, A.G., Schubert, L., Shapiro, S.C. (eds.) KR 1998: Principles of Knowledge Representation and Reasoning, pp. 488–498. Morgan Kaufmann (1998)
7. Konieczny, S., Pino Pérez, R.: Merging with Integrity Constraints. In: Hunter, A., Parsons, S. (eds.) ECSQARU 1999. LNCS (LNAI), vol. 1638, pp. 233–244. Springer, Heidelberg (1999)
8. Konieczny, S., Pino-Pérez, R.: Merging information under constraints: a logical framework. *Journal of Logic and Computation* 12(5), 773–808 (2002)
9. Liberatore, P., Schaerf, M.: Arbitration (or how to merge knowledge bases). *IEEE Transactions on Knowledge and Data Engineering* 10(1), 76–90 (1998)
10. Lin, J., Mendelzon, A.: Knowledge base merging by majority. In: Pareschi, R., Fronhofer, B. (eds.) Dynamic Worlds: From the Frame Problem to Knowledge Management. Kluwer Academic (1999)
11. Marchi, J., Perrussel, L.: Prime normal forms in belief merging. In: FLAIRS Conference (2011)
12. Pozos Parra, P., Borja Macías, V.: Partial satisfiability-based merging. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 225–235. Springer, Heidelberg (2007)
13. van Dalen, D.: Logic and Structure, 2nd edn. Springer, Heidelberg (1983)

Toward Justifying Actions with Logically and Socially Acceptable Reasons

Hiroyuki Kido and Katsumi Nitta

Interdisciplinary Graduate School of Science and Engineering

Tokyo Institute of Technology, Japan

{kido,nitta}@ntt.dis.titech.ac.jp

Abstract. This paper formalizes argument-based reasoning for actions supported by believable reasons in terms of nonmonotonic consequences and desirable reasons in terms of Pareto optimality and maximizing social welfare functions. Our unified approach gives a four-layer practical argumentation framework structured with a propositional modal language with defaults and defeasible inference rules associated with practical reasoning. We show that the unified argument-based reasoning justifies an argument whose conclusion is supported by Pareto optimal, social welfare maximizing and nonmonotonic consequence reasons. Our formalization contributes to extend argument-based reasoning so that it can formally combine reasoning about logical believability and social desirability by benefiting from economic notions.

1 Introduction

1.1 Background and Objectives

Argumentation in artificial intelligence gives formal methods finding out persuasive reasons on the basis of exchanging arguments. Nowadays, Dungian semantics and an abstract argumentation framework [1] are known as a canonical way for realizing argument-based reasoning due to the fact that they give a general framework for various nonmonotonic reasoning. Recently, in [2], the authors propose a generalization of Dungian semantics, called practical argumentation semantics, that can solve any problems evaluating Pareto optimal solutions. From the viewpoint that argumentation is a decision method, we think that it is worth noting that practical argumentation semantics expands the scope of argumentation to decide not only believability, but also social desirability in terms of Pareto optimality. However, the research has drawbacks that, first of all, it does not address decisions based on persuasive reasons argument-based reasoning characteristically handles. Second of all, it does not give mechanisms to distinguish among Pareto optimal solutions. These drawbacks include the problems of how to give internal structures of arguments and defeat functions in order to develop argument-based reasoning on a specific logical language and how to make a correlation between reasons in arguments and economic notions on the logical language. We think that this reason-based approach for economic

notions has significance as an alternative method to value-based decisions. In fact, compared to value-based approaches, reason-based approaches have attractive features that they give the way we normally think and talk about choices, the way suited to handle complex and real world decisions, and the way to understand the conflict characterizing the making of decisions, although it has disadvantages in weak rigor [3].

The aim of this paper is to formalize argument-based reasoning for actions supported by believable reasons in terms of nonmonotonic consequences and desirable reasons in terms of Pareto optimality and maximizing social welfare functions. Using modal propositional language with defaults and defeasible inference rules for practical reasoning, we give three kinds of structured practical argumentation frameworks for nonmonotonic consequences, for Pareto optimality, and for social welfare, respectively. We combine these frameworks into a layered practical argumentation framework to evaluate actions supported by Pareto optimal, social welfare maximizing, and nonmonotonic consequence reasons. Our contribution is summarized as follows. First of all, we extend argument-based reasoning so that it can formally combine not only logical properties, i.e., non-monotonic consequences, but also about social properties, i.e., Pareto optimality and social welfare functions on a specific logical language. Second of all, we propose a unified method to evaluate actions with logically and socially acceptable reasons by proposed layered practical argumentation framework.

1.2 Related Work

In [4], the authors apply Pareto optimality and social welfare functions for evaluating desirable extensions, i.e., argumentation consequences, given by Dungian semantics. Whereas this is the first report introducing these notions in argumentation, it does not address justifying these notions by means of argument-based reasoning. Game theoretic approaches tend to assume fixed and immutable agents' utility functions [5]. Contrary to human decision makings, however, the approaches do not tolerate utility changes caused by new information. For instance, when I set out to buy a car, I may initially decide that I want a car with an electric sun roof. However, if I subsequently read that electric sun roofs are unreliable and tend to leak, then this might well change my preferences [6]. Furthermore, people may not understand an explanation as a sequence of complex equations provided by the approaches. Our argument-based approach allows agents to justify their beliefs by providing mechanisms for making counterarguments and its explanations are reasons we normally use when we think and talk. In [7], the authors give pessimistic and optimistic criteria for qualitative decision making. The approach, however, does not handle justifying conclusions based on arguments and counterarguments under knowledge with exceptions.

This paper is organized as follows. In Section 2, we give a motivating example and Section 3 gives logical preliminaries. Section 4 gives internal structures of practical argumentation frameworks. Section 5 defines and analyzes a combination of three kinds of frameworks. Section 6 shows an illustrative example and Section 7 describes conclusions and future work.

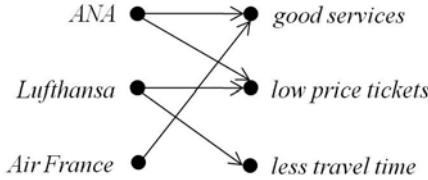


Fig. 1. The relationships between means and desires

2 Motivating Example

Let us consider the situation where two agents i and j are looking for an airline to get Barcelona for a trip. Agent i desires to get good services and to get low price tickets. He prefers to realize both desires rather than each desire and each desire rather than nothing, but both are incompatible for him. Agent j desires to get good services and to spend less travel time. She prefers to realize both desires rather than less travel time, less travel time rather than good services, and good services rather than nothing. They have common default knowledge that using ANA realizes to get good services and low price tickets, using Lufthansa realizes to get low price tickets and less travel time, and using Air France realizes to get good services. Figure 1 is the default knowledge expressed using a directed graph where each right node represents a desire and each left node represents a possible means realizing the desires. An arrow represents that the means realizes the desire. Now, we consider the situation that agents get new conflicting knowledge that agents cannot get low price tickets even if they buy airline tickets provided by ANA because economy class seats provided by ANA are all booked up, and using ANA rejects to get low price tickets.

The question we are interested in is which airlines are logically and socially acceptable, and how agents find out them. Our original idea is to evaluate alternative airlines on the basis of reasons why agents want to use them, i.e., in terms of realized or rejected desires associated with using airlines. We introduce the notions of nonmonotonic consequences as formal criteria for logical acceptability, and Pareto optimality and social welfare functions as formal criteria for social acceptability. Our technical proposal solves these whole evaluations by four-layer argumentation-based reasoning where agents argue about what is true in terms of nonmonotonic consequences at the first step, about what is good in terms of Pareto optimality at the second and third steps, and about what is fair in terms of maximizing social welfare functions at the fourth step. In this example, let A be an argument stating that using ANA is desirable because it provides good services and low price tickets. Then A is not justified at the first step because it is defeated by an argument stating that agents cannot get low price tickets even if they use ANA because economy class seats provided by ANA are all booked up. A justified argument A' related to A is the argument stating that using ANA is desirable because it provides good services. Intuitively, the situation corresponds

to deleting the edge from ANA to low price tickets in Figure 1. At the second step, we make agents argue on the basis not of inconsistency, but of subjective preferences of reasons. Then, under j , A' is defeated by B stating that using Lufthansa is desirable because it provides low price tickets and less travel time, due to the fact that the reasons of B , i.e., low price tickets and less travel time, are preferable to the reasons of A' , i.e., good services, for j . Based on the defeat relations, we show that A' , B and C stating that using Air France is desirable because it provides good services are justified, and their conclusions, i.e., using ANA, Lufthansa and Air France, are supported by Pareto optimal reasons. Similarly, the third step justifies undesirable airlines supported by Pareto optimal reasons. In this example, we show that an argument D , stating that using ANA is undesirable because it throws out low price tickets, is justified and therefore whose reason is Pareto optimal. We exclude A' that concludes the opposite of the conclusion of D from consideration. This approach is motivated by the cognitive experimental report that to the extent of basing people's decisions on reasons for and against the options under consideration, they are likely to focus on reasons for choosing an option when deciding which to choose, and to focus on reasons for rejecting an option when deciding which to reject [3]. At the fourth step, we make agents argue on the basis not of subjective preferences, but of a social preference provided by social welfare functions. Then B defeats C with respect to our social welfare in the sense that the reason of B is more preferable to C for j although they are incompatible for i . As a result, we show that only B is justified and whose reason maximizes our social welfare function. Therefore, we conclude that using Lufthansa is supported by logically and socially acceptable reasons in terms of nonmonotonic consequences, Pareto optimality and the social welfare function.

3 Logical Preliminaries

The practical argumentation semantics [2] is a general rule for evaluating argumentation consequences, i.e., extensions, of given practical argumentation frameworks. It is adequate for arguing about practical problems because it tolerates subjective defeat relations caused by difference of desires, values, morality, and so on. The formal definitions with some terminological changes are as follows.

Definition 1. [2] The practical argumentation framework is defined as 3-tuple $PRAF = \langle \text{Args}, \text{Agents}, \text{Defeat} \rangle$ where Args is a set of arguments, Agents is a set of agents, and Defeat is a function that maps Agents into $2^{\text{Args} \times \text{Args}}$.

- A set $S \subseteq \text{Args}$ of arguments is conflict-free to a set $N \subseteq \text{Agents}$ of agents if A does not defeat B under any agent $i \in N$, for all arguments $A, B \in S$.
- An argument $A \in \text{Args}$ is acceptable to a set $N \subseteq \text{Agents}$ of agents with respect to a set $S \subseteq \text{Args}$ of arguments if each argument defeating A under an agent $i \in \text{Agents}$ is defeated by an argument $B \in S$ under an agent $j \in N$.

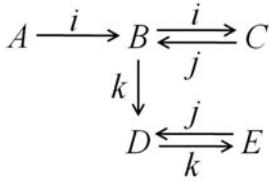


Fig. 2. *PRAF* expressed with a labeled directed graph

Table 1. The preferred and grounded extensions

N	PE	GE
\emptyset	$\{A\}$	$\{A\}$
$\{i\}$	$\{A, C\}$	$\{A, C\}$
$\{j\}$	$\{A, C, E\}$	$\{A\}$
$\{k\}$	$\{A\}$	$\{A\}$
$\{i, j\}$	$\{A, C, E\}$	$\{A, C\}$
$\{i, k\}$	$\{A, C, D\}$	$\{A, C\}$
$\{j, k\}$	$\{A, C, D\}, \{A, C, E\}$	$\{A\}$
$\{i, j, k\}$	$\{A, C, D\}, \{A, C, E\}$	$\{A, C\}$

- A set $S \subseteq \text{Args}$ of arguments is admissible to a set $N \subseteq \text{Agents}$ of agents if S is conflict-free to N and each argument in S is acceptable to N with respect to S .
- A set $S \subseteq \text{Args}$ of arguments is a preferred extension to a set $N \subseteq \text{Agents}$ of agents if S is a maximal admissible set to N with respect to the set inclusion.
- A characteristic function $F^N : 2^{\text{Args}} \mapsto 2^{\text{Args}}$ is defined as follows.
 - $F^N(S) = \{A \in \text{Args} \mid A \text{ is acceptable to } N \subseteq \text{Agents with respect to } S \subseteq \text{Args}\}$
- A set of $S \subseteq \text{Args}$ of arguments is a grounded extension to a set $N \subseteq \text{Agents}$ of agents if S is the least fixed point of F^N .

For given $\text{PRAF} = \langle \text{Args}, \text{Agents}, \text{Defeat} \rangle$, we say that x defeats y under i if there exists $i \in \text{Agents}$ and $(x, y) \in \text{Defeat}(i)$, an argument $A \in \text{Args}$ is credulously justified under N , denoted by $A \in P^N(\text{PRAF})$, if A is a member of some preferred extension to N , and skeptically justified under N , denoted by $A \in G^N(\text{PRAF})$, if A is a member of the grounded extension to N .

Example 1. $\text{PRAF} = \langle \{A, B, C, D, E\}, \{i, j, k\}, \text{Defeat} \rangle$, where $\text{Defeat}(i) = \{(A, B), (B, C)\}$, $\text{Defeat}(j) = \{(C, B), (E, D)\}$, and $\text{Defeat}(k) = \{(B, D), (D, E)\}$, is a practical argumentation framework. Figure 2 is a labeled directed graph for PRAF where an arrow from x to y with label i means x defeats y under i . Table 1 shows the preferred extensions PE and the grounded extensions GE to a set N of agents. For instance, $S = \{A, C, D\}$ is the preferred extension to $N = \{i, k\}$ because S is conflict-free to N and it is a maximal set of all admissible sets to N : $\emptyset, \{A\}, \{A, C\}, \{A, D\}$, and $\{A, C, D\}$. $\{A, C\}$ is the grounded extension to N because it is the least fixed point of $F^{\{i, k\}}$, i.e., $F^{\{i, k\}}(\emptyset) = \{A\}$, $F^{\{i, k\}}(\{A\}) = \{A, C\}$, and $F^{\{i, k\}}(\{A, C\}) = \{A, C\}$.

Pareto optimality is known as a formal canonical criterion of society's well being or goodness, and a solution is Pareto optimal if no agents can be made better off without making someone else worse off. Dungean semantics and an abstract argumentation framework [1] are known as a general framework for nonmonotonic reasoning. The practical argumentation semantics has the properties that evaluating each Pareto optimal solutions and defeasible consequences with Dungean

semantics can be translated to evaluating extensions of a particular practical argumentation framework [2].

Argument-based reasoning allows us to easily introduce various kinds of defeasible or/and nondeductive inferences, denoted by \rightsquigarrow , as giving structures of arguments. The following two inference rules aim to formally capture the intuitions that having more reasons for a given conclusions makes such a conclusions more credible, and that conclusions cannot be drawn from a part of the reasons.

- Accrual [8]: $\varphi^{l_1}, \dots, \varphi^{l_n} \rightsquigarrow \varphi$
- Accrual-undercutter [8]: $\varphi^{l_1}, \dots, \varphi^{l_n} \rightsquigarrow \neg[\varphi^{l_1}, \dots, \varphi^{l_{n-i}} \rightsquigarrow \varphi] (1 \leq i \leq n)$

“ φ^{l_i} ” denotes that φ is derived from a set l_i , called a label, by applying some defeasible inference rule. “[ϕ]” denotes an objectification operator that transfers ϕ in the meta-language into the expression in the object-language, e.g, a modal propositional language. A rule “ $\phi, \phi \Rightarrow \psi \rightsquigarrow \psi \{\phi, \phi \Rightarrow \psi\}$ ” is an example of applying a label into the defeasible modus ponens. “ $\phi \Rightarrow \psi$ ” denotes an exception permitting rule, called a defeasible rule, stating that ψ is typically the case if ϕ is the case. The following two inference rules aim to formally capture the intuitions that if an agent who desires ψ and believes ω also believes that realizing ϕ in a circumstance ω realizes ψ , then this is a reason for desiring ϕ , and if an agent who desires ψ and believes ω also believes that realizing ϕ in a circumstance ω realizes $\neg\psi$, then this is a reason not to desire ϕ [9].

- Positive practical syllogism [9]: $\phi \wedge \omega \Rightarrow \psi, \mathbf{D}\psi, \omega \rightsquigarrow \mathbf{D}\phi$
- Negative practical syllogism [9]: $\phi \wedge \omega \Rightarrow \neg\psi, \mathbf{D}\psi, \omega \rightsquigarrow \neg\mathbf{D}\phi$

“ $\mathbf{D}\psi$ ” denotes agents’ desire for ψ . The positive and negative ones can be abbreviated as $\phi \Rightarrow \psi, \mathbf{D}\psi \rightsquigarrow \mathbf{D}\phi$ and $\phi \Rightarrow \neg\psi, \mathbf{D}\psi \rightsquigarrow \neg\mathbf{D}\phi$, respectively.

4 Structuring Arguments and Defeat Functions

4.1 Practical and Theoretical Arguments

There exist various formal logical languages describing intelligent agent’s knowledge, e.g., epistemic logic, doxastic logic, BDI logic, KARO logic and so on. However, they do not tolerate statements with exceptions argument-based reasoning normally assumes. We adopt the language of Reiter’s default logic [10] for formally describing default knowledge. However, it is still insufficient for representing agents’ desires practical reasoning often refers. We define a logical language \mathcal{L} which consists of a union of languages \mathcal{L}_0 and \mathcal{L}_1 . \mathcal{L}_0 is a language of propositional modal logic with modal operator \mathbf{D} standing for agents’ desires and \mathcal{L}_1 is a set of defaults defined on \mathcal{L}_0 . Every formula in \mathcal{L}_1 has a form “ $\varphi \Rightarrow \psi$ ” where φ is a conjunction of atomic formulas in \mathcal{L}_0 and ψ is an atomic formula in \mathcal{L}_0 . The modal propositional logic is assumed to have proof system called \mathbf{K} . \mathbf{K} axiomatizes all formulas instantiated by axiom schema $\mathbf{D}(\varphi \rightarrow \psi) \rightarrow (\mathbf{D}\varphi \rightarrow \mathbf{D}\psi)$, and there are two rules of proof: modus ponens (if $\vdash \varphi$ and $\vdash \varphi \rightarrow \psi$ then $\vdash \psi$) and modal generalization (if $\vdash \varphi$ then $\vdash \mathbf{D}\varphi$). For simplicity, we assume that \mathbf{D}

is only effective to atomic formulas in \mathcal{L}_0 . We interpret an action described as a proposition is true when the action has been really taken by an agent. Defining actions in the logical language is beyond the scope of this paper. Reasoning operates on a default theory $T = (F, D)$ and a set $G_i (1 \leq i \leq n)$ of agents' desires, where F is a consistent subset of \mathcal{L}_0 , D is a subset of \mathcal{L}_1 , and G_i is a consistent subset of \mathcal{L}_0 with operator \mathbf{D} . Each agent i has a preference P_i on the powerset of G_i , i.e., $P_i \subseteq 2^G \times 2^G$, expressed as at least quasi-order satisfying reflexivity and transitivity.

Every argument is structured on a tree where a set of edges between a node and its children is an inference rule, and a node is a well-formed formula or a special kind of description undercutting use of an inference rule. For argument A , we say that a node p in A is derived by an inference rule i if p is derived from its all children using i . A set of all nodes of A is a body of A , denoted by $\text{body}(A)$, and a root node or a node with depth 0 is a conclusion of A , denoted by $\text{conc}(A)$. A subtree of A , without considering the order of sibling relationships, is a subargument of A . Args denotes a set of arguments. We distinguish practical and theoretical arguments. Two kinds of practical arguments are defined recursively using the accrual inference rule, and the positive and negative practical syllogism, respectively.

Definition 2. Let A be an argument and $\varphi \in \mathcal{L}_0$. A is a positive practical argument if A is a tree with depth 0 and $\text{conc}(A) = \mathbf{D}\varphi$, or $\text{conc}(A)$ is derived by the accrual inference rule, every node with depth 1 is derived by the positive practical syllogism and every subtrees, whose root is $\mathbf{D}\varphi$ with depth 2, is a positive practical argument.

Definition 3. Let A be an argument and $\varphi \in \mathcal{L}_0$. A is a negative practical argument if A is a tree with depth 0 and $\text{conc}(A) = \neg\mathbf{D}\varphi$, or $\text{conc}(A)$ is derived by the accrual inference rule, every node with depth 1 is derived by the negative practical syllogism and every subtrees, whose root is $\neg\mathbf{D}\varphi$ with depth 2, is a negative practical argument.

We call a practical argument, that is subargument of another argument, a practical subargument.

Definition 4. Let A be a practical argument. A reason in A or a reason of $\text{conc}(A)$ with respect to A , denoted by $\text{reason}(A)$, is a set of conclusions of all practical subarguments $A' \neq A$ of A .

Example 2. Figure 3 is an example of a positive practical argument A expressed by a proof tree where the bottom and the top formulas correspond to the root and the leaves, respectively, in a tree. $\text{reason}(A) = \{\mathbf{D}a, \mathbf{D}b, \mathbf{D}d\}$.

The following accrual undercutting argument is a kind of theoretical argument and it is actually used in [9] without formal definition. We give the formal definition as follows.

Definition 5. Let A be an argument and $\varphi \in \mathcal{L}_0$. A is an accrual undercutting argument if $\text{conc}(A)$ is derived by the accrual undercutter, every node with

$$A: \quad \frac{\frac{\frac{\mathbf{D}a \quad b \Rightarrow a}{\mathbf{D}b^{l1}}}{\mathbf{D}b} \quad c \Rightarrow b \quad \frac{\mathbf{D}d \quad c \Rightarrow d}{\mathbf{D}c^{l3}}}{\mathbf{D}c^{l2}} \quad \mathbf{D}c$$

Fig. 3. A positive practical argument

depth 1 is derived by the positive or negative practical syllogism, and every subtrees, whose root is $\mathbf{D}\varphi$ or $\neg\mathbf{D}\varphi$ with depth 2, is a positive or negative practical argument.

In what follows, $Agents$ denotes a set of agents, $Pargs^+$ and $Pargs^-$ denote a set of positive and negative practical arguments, respectively, $Targs$ denotes a set of theoretical arguments, i.e., the union of a set of accrual undercutting arguments and a set of arguments constructed using any combinations of valid inferences and defeasible modus ponens, $Pargs = Pargs^+ \cup Pargs^-$, $Args = Pargs \cup Targs$, and $Reason(Pargs)$ denotes a set of reasons in $Pargs$.

4.2 Theoretical, Preference-Based and Welfare-Based Defeats

This subsection defines three kinds of defeat functions. A theoretical defeat function handles incompatibility between two arguments and a preference-based defeat function handles subjective preferability between two practical arguments.

Definition 6. Let $i \in Agents$ and $A, B \in Args$. A theoretical defeat function $Tdefeat : Agents \mapsto 2^{Args \times Args}$ is defined as follows. $(A, B) \in Tdefeat(i)$ if any of the following three conditions holds: (1) There exists φ such that $\varphi \in body(A)$ and $\neg\varphi \in body(B)$. (2) There exists φ and ϕ such that $\varphi \wedge \neg\phi \in body(A)$ and $\varphi \Rightarrow \phi \in body(B)$. (3) There exists $\varphi^{l1}, \dots, \varphi^{li}$ ($1 \leq i \leq n$) and φ such that $\varphi^{l1} \wedge \dots \wedge \varphi^{li} \not\rightarrow \varphi \in body(A)$, $\varphi \in body(B)$, the children of φ are $\varphi^{l1}, \dots, \varphi^{li}$, and φ is derived by a defeasible inference rule.

Definition 7. Let $i \in Agents$ and $A, B \in Pargs$. A preference-based defeat function $Pdefeat : Agents \mapsto 2^{Pargs \times Pargs}$ is defined as $Pdefeat(i) = \{(A, B) \mid (reason(B) \cap G_i, reason(A) \cap G_i) \notin P_i\}$.

Social welfare can be measured by various kinds of social welfare functions. In particular, Rawls one is to opt for the state that maximizes the utility of the person in the worst position. On the basis of Rawls's notion, we define a social welfare function in terms of the number of sets of desires that is more preferable to the set of desires included in the reason of an argument.

Definition 8. A social welfare function, $W : Reasons(Pargs) \mapsto \mathbb{R}$, is defined as $W(X) = \min\{|\{Y \mid (Y, X \cap G_i) \in P_i\}|^{-1}, 1 \leq i \leq n\}$.

Finally, we introduce an welfare-based defeat function handing objective preferability between two practical arguments.

Definition 9. Let $i \in Agents$ and $A, B \in Pargs$. An welfare-based defeat function $W\text{defeat} : Agents \mapsto 2^{Pargs \times Pargs}$ is defined as $W\text{defeat}(i) = \{(A, B) \mid W(\text{reason}(A)) > W(\text{reason}(B))\}$.

5 Justifying Logically and Socially Acceptable Reasons

5.1 Analyzing Structured Practical Argumentation Frameworks

This subsection defines and analyzes structured practical argumentation frameworks for nonmonotonic consequences, for Pareto optimality and for social welfare, respectively, as instantiations of practical argumentation frameworks.

Definition 10. Let $PRAF = \langle Ar, Ag, De \rangle$ be a practical argumentation framework. $PRAF$ is a structured practical argumentation framework for nonmonotonic consequences, denoted by $PRAF_{NC}$, if $Ar = Targs \cup Pargs^+ \cup Pargs^-$, $Ag = Agents$ and $De(i) = T\text{defeat}$, for all $i \in Ag$.

Proposition 1. Let $PRAF_{NC} = \langle Args, Agents, Defeat \rangle$ be a structured practical argumentation framework for nonmonotonic consequences. For any preferred extension PE of $PRAF_{NC}$ to $Agents$, there is no $\phi, \psi \in \{\text{conc}(A) \mid A \in PE\}$ such that $\phi \leftrightarrow \neg\psi$.

Proof. It is obvious from (1) in Definition 6. \square

Definition 11. Let $PRAF = \langle Ar, Ag, De \rangle$ be a practical argumentation framework. $PRAF$ is a structured practical argumentation framework for Pareto optimality, denoted by $PRAF_{PO}$, if $Ar = Pargs^+ \cup Pargs^-$, $Ag = Agents$ and $De(i) = P\text{defeat}(i)$, for all $i \in Ag$.

Proposition 2. Let $PRAF_{PO} = \langle Args, Agents, Defeat \rangle$ be a structured practical argumentation framework for Pareto optimality. For $P \in Args$, reason (P) is Pareto optimal with respect to $\{(X, Y) \in Reasons(Args)^2 \mid (X \cap G_i, Y \cap G_i) \in P_i\}$ for $i \in Agents$ iff P is a member of some preferred extension of $PRAF_{PO}$ to $Agents$.

Proof. From Theorem 1 in [2], P is a member of some preferred extension to $Agents$ iff P is Pareto optimal with respect to $\overline{P\text{defeat}(i)^{-1}}(i \in Agents)$. Now, let S_i be $\{(X, Y) \in Reasons(Args)^2 \mid (X \cap G_i, Y \cap G_i) \in P_i\}$. It can be shown that $P \in Args$ is Pareto optimal with respect to $\overline{P\text{defeat}(i)^{-1}}(i \in Agents)$ iff reason(P) is Pareto optimal with respect to S_i ($i \in Agents$), if it is shown that $(A, B) \in \overline{P\text{defeat}(i)^{-1}}$ iff $(\text{reason}(A), \text{reason}(B)) \in S_i$, for all $i \in Agents$. (\Rightarrow) $(\text{reason}(A), \text{reason}(B)) \in S_i$ because $(\text{reason}(A), \text{reason}(B)) \in Reasons(Args)^2$ due to $A, B \in Args$, and $(\text{reason}(A) \cap G_i, \text{reason}(B) \cap G_i) \in P_i$. (\Leftarrow) $(A, B) \in \overline{P\text{defeat}(i)^{-1}}$ because $(A, B) \in Args^2$ due to $(\text{reason}(A), \text{reason}(B)) \in Reasons(Args)^2$, and $(\text{reason}(A) \cap G_i, \text{reason}(B) \cap G_i) \in P_i$. \square

Definition 12. Let $PRAF = \langle Ar, Ag, De \rangle$ be a practical argumentation framework. $PRAF$ is a structured practical argumentation framework for social welfare, denoted by $PRAF_{SW}$, if $Ar = Pargs^+ \cup Pargs^-$, $Ag = Agents$ and $De(i) = W\text{defeat}(i)$, for all $i \in Ag$.

Proposition 3. Let $PRAF_{SW} = \langle \text{Args}, \text{Agents}, \text{Defeat} \rangle$ be a structured practical argumentation framework for social welfare. For $P \in \text{Args}$, $\text{reason}(P)$ maximizes the social welfare function with respect to $\text{Reasons}(\text{Args})$ iff P is a member of some preferred extension of $PRAF_{SW}$ to Agents .

Proof. (\Rightarrow) There is no $Q \in \text{Args}$ and $i \in \text{Agents}$ such that $(Q, P) \in \text{Defeat}(i)$ because there is no $Q \in \text{Args}$ such that $W(\text{reason}(Q)) > W(\text{reason}(P))$. (\Leftarrow) A contradiction is derived from the assumption that P is a member of some preferred extension and $\text{reason}(P)$ does not maximize the social welfare function. Under the assumption, there is $Q \in \text{Args}$ such that $(Q, P) \in \text{Defeat}(i)$ and $(P, Q) \notin \text{Defeat}(i)$, for all $i \in \text{Agents}$. Furthermore, if there is $R \in \text{Args}$ such that $(R, Q) \in \text{Defeat}(i)$, then $(R, P) \in \text{Defeat}(i)$ due to transitivity of $\text{Defeat}(i)$. Thus, P is not acceptable to Agents with respect to any preferred extension to Agents . \square

5.2 Analyzing Layered Practical Argumentation Frameworks

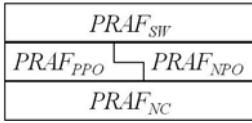
This subsection aims to show that actions supported by Pareto optimal, social welfare maximizing and nonmonotonic consequence reasons can be justified on a layered practical argumentation framework combining the above defined three structured practical argumentation frameworks. We assume the following arbitrary but fixed *argumentation theory* on which a layered practical argumentation framework is constructed: $\langle Targs, Pargs^+, Pargs^-, Agents, Tdefeat, Pdefeat, Wdefeat \rangle$. The bottom layer is a structured practical argumentation framework for nonmonotonic consequences $PRAF_{NC}$ that has restricted defeat relations from only theoretical arguments to any arguments.

- $\langle Targs \cup Pargs^+ \cup Pargs^-, Agents, Tdefeat' \rangle$ where $Tdefeat'(i) = Tdefeat(i) \cap (Targs \times (Targs \cup Pargs^+ \cup Pargs^-))$, for all $i \in \text{Agents}$

The middle layer consists of two different structured practical argumentation frameworks for Pareto optimality. For $S \subseteq Pargs^X$, we define a complement of S , denoted by \overline{S} , as $\{B \in Pargs^Y \mid \text{conc}(B) \leftrightarrow \neg\text{conc}(A), A \in S\}$ where both X and Y are + or -, and $X \neq Y$. The first one $PRAF_{PPO}$ considers only positive practical arguments skeptically justified in $PRAF_{NC}$. The second one $PRAF_{NPO}$ considers the intersection of a set of negative ones skeptically justified in $PRAF_{NC}$ and the complement of a set of positive practical arguments credulously justified arguments in $PRAF_{PPO}$.

- $\langle G^{Agents}(PRAF_{NC}) \cap Pargs^+, Agents, Pdefeat' \rangle$ where $Pdefeat'(i) = Pdefeat(i) \cap (G^{Agents}(PRAF_{NC}) \cap Pargs^+)^2$, for all $i \in \text{Agents}$
- $\langle G^{Agents}(PRAF_{NC}) \cap \overline{P}^{Agents}(PRAF_{PPO}), Agents, Pdefeat'' \rangle$ where $Pdefeat''(i) = Pdefeat(i) \cap (G^{Agents}(PRAF_{NC}) \cap \overline{P}^{Agents}(PRAF_{PPO}))^2$, for all $i \in \text{Agents}$

The top layer is a structured practical argumentation framework for social welfare $PRAF_{SW}$ whose practical arguments are restricted to the difference set between a set of credulously justified arguments in $PRAF_{PPO}$ and the complement of a set of credulously justified arguments in $PRAF_{NPO}$.

Table 2. Practical arguments in the argumentation theory**Fig. 4.** A hierarchy of a layered practical argumentation framework

arguments	conclusions	reasons	$W(reason)$
P_1	DNH	{DGS, DLP}	1/3
P_2	DNH	{DGS}	1/3
P_3	DNH	{DLP}	1/4
P_4	DLH	{DLP, DLT}	1/2
P_5	DLH	{DLP}	1/4
P_6	DLH	{DLT}	1/4
P_7	DAF	{DGS}	1/3
P_8	$\neg DHN$	{DLP}	1/4

- $< P^{Agents}(PRAF_{PPO}) \setminus \overline{P^{Agents}(PRAF_{NPO})}, Agents, Wdefeat' >$ where $Wdefeat'(i) = Wdefeat(i) \cap (P^{Agents}(PRAF_{PPO}) \setminus \overline{P^{Agents}(PRAF_{NPO})})^2$, for all $i \in Agents$

We call $PRAF_{SW}$ a *layered practical argumentation framework* constructed on $PRAF_{NC}$, $PRAF_{PPO}$ and $PRAF_{NPO}$. The framework can be hierarchized as shown in Figure 4. Now, we show a property of the layered practical argumentation framework on the basis of the properties individual frameworks have. In the following definition, $Args(PRAF)$ represents a set of arguments in $PRAF$.

Theorem 1. Let $PRAF_{SW}$ be a layered practical argumentation framework constructed on $PRAF_{NC}$, $PRAF_{PPO}$ and $PRAF_{NPO}$. For any preferred extension PE of $PRAF_{SW}$ to $Agents$, (1) there is no $\phi, \psi \in \{\text{conc}(P) \mid P \in PE\}$ such that $\phi \leftrightarrow \neg\psi$, (2) for $P \in PE$, $reason(P)$ is Pareto optimal with respect to $\{(X, Y) \in Reasons(Args(PRAF_{PPO}))^2 \mid (X \cap G_i, Y \cap G_i) \in P_i\}$ for $i \in Agents$ and (3) for $P \in PE$, $reason(P)$ maximizes the social welfare function with respect to $Reasons(Args(PRAF_{SW}))$.

Proof. (1) is obvious from $PE \subseteq Pargs^+$. (2) is derived by replacing $PRAF_{PO}$ in Proposition 2 by $PRAF_{PPO}$. (3) is derived by replacing $PRAF_{SW}$ in Proposition 3 by the layered practical argumentation framework. \square

6 Illustrative Example

We assume default theory $T = (\{BU\}, \{NH \Rightarrow GS, NH \Rightarrow LP, LH \Rightarrow LP, LH \Rightarrow LT, AF \Rightarrow GS, NH \Rightarrow \neg LP, BU \Rightarrow NH \wedge \neg LP\})$, agents i and j 's desires $G_i = \{\text{DGS}, \text{DLP}\}$ and $G_j = \{\text{DGS}, \text{DLT}\}$, and their preferences $P_i = \{(\{\text{DGS}, \text{DLP}\}, \{\text{DGS}, \text{DLP}\}), (\{\text{DGS}\}, \{\text{DGS}\}), (\{\text{DLP}\}, \{\text{DLP}\}), (\emptyset, \emptyset), (\{\text{DGS}, \text{DLP}\}, \{\text{DGS}\}), (\{\text{DGS}, \text{DLP}\}, \{\text{DLP}\}), (\{\text{DGS}, \text{DLP}\}, \emptyset), (\{\text{DGS}\}, \emptyset), (\{\text{DLP}\}, \emptyset)\}$ and $P_j = \{(\{\text{DGS}, \text{DLT}\}, \{\text{DGS}, \text{DLT}\}), (\{\text{DGS}\}, \{\text{DGS}\}), (\{\text{DLT}\}, \{\text{DLT}\}), (\emptyset, \emptyset), (\{\text{DGS}, \text{DLT}\}, \{\text{DLT}\}), (\{\text{DGS}, \text{DLT}\}, \{\text{DGS}\}), (\{\text{DGS}, \text{DLT}\}, \emptyset), (\{\text{DLT}\}, \{\text{DGS}\}), (\{\text{DLT}\}, \emptyset), ((\{\text{DGS}\}, \emptyset))\}$. GS states

$$\begin{array}{c}
 T_1 : \frac{BU \quad BU \Rightarrow NH \wedge \neg LP}{NH \wedge \neg LP} \\
 \\
 \frac{\mathbf{DGS} \quad NH \Rightarrow GS \quad \frac{T_2 :}{\mathbf{DLP} \quad NH \Rightarrow LP}}{\mathbf{DNH}^{l_i} \quad \mathbf{DNH}^{l_{i+1}}} \\
 \frac{}{\mathbf{DNH}^{l_i} \not\sim \mathbf{DNH}} \\
 \\
 \frac{\mathbf{DGS} \quad NH \Rightarrow GS \quad \frac{T_3 :}{\mathbf{DLP} \quad NH \Rightarrow LP}}{\mathbf{DNH}^{l_i} \quad \mathbf{DNH}^{l_{i+1}}} \\
 \frac{}{\mathbf{DNH}^{l_{i+1}} \not\sim \mathbf{DNH}}
 \end{array}
 \qquad
 \begin{array}{c}
 T_4 : \frac{\mathbf{DGS} \quad LH \Rightarrow GS \quad \mathbf{DLT} \quad LH \Rightarrow LT}{\mathbf{DLH}^{l_j} \quad \mathbf{DLH}^{l_{j+1}}} \\
 \frac{}{\mathbf{DLH}^{l_j} \not\sim \mathbf{DLH}} \\
 \\
 \frac{\mathbf{DGS} \quad LH \Rightarrow GS \quad \frac{T_5 :}{\mathbf{DLT} \quad LH \Rightarrow LT}}{\mathbf{DLH}^{l_j} \quad \mathbf{DLH}^{l_{j+1}}} \\
 \frac{}{\mathbf{DLH}^{l_{j+1}} \not\sim \mathbf{DLH}}
 \end{array}$$

Fig. 5. Theoretical arguments in the argumentation theory

that we get good services, LP that we get airline tickets at a low price, LT that we spend less travel time, BU that economy class seats provided by ANA are all booked up, and NH, LH and AF that we use airline company ANA, Lufthansa and Air France, respectively. Now, agents conduct argument-based reasoning from the following argumentation theory constructed from $T: < Targs = \{T_i \mid 1 \leq i \leq 5\}, Pargs^+ = \{P_i \mid 1 \leq i \leq 7\}, Pargs^- = \{P_8\}, Agents, Tdefeat, Pdefeat, Wdefeat >$. Table 2 shows key elements of practical arguments in $Pargs$ and Figure 5 shows theoretical arguments in $Targs$. Our final goal is to get preferred extensions of a layered practical argumentation framework. To this end, the first subgoal is to evaluate the grounded extension of following $PRAF_{NC}$.

$$\begin{aligned}
 & G^{\{i,j\}}(< Targs \cup Pargs^+ \cup Pargs^-, Agents, Tdefeat' >) \\
 &= G^{\{i,j\}}(< \{P_i, T_j \mid 1 \leq i \leq 8, 1 \leq j \leq 5\}, \{i, j\}, Tdefeat' >)(Tdefeat'(i) = \\
 & \quad Tdefeat'(j) = \{(T_1, P_1), (T_1, P_3), (T_2, P_2), (T_3, P_3), (T_4, P_5), (T_5, P_6), (T_1, \\
 & \quad T_2), (T_1, T_3)\}) = \{T_1, T_4, T_5, P_2, P_4, P_7, P_8\}
 \end{aligned}$$

The second subgoal is to evaluate preferred extensions of following $PRAF_{PPO}$.

$$\begin{aligned}
 & P^{Agents}(< G^{\{i\}}(PRAF_{NC}) \cap Pargs^+, Agents, Pdefeat' >) \\
 &= P^{\{i,j\}}(< \{P_2, P_4, P_7\}, \{i, j\}, Pdefeat' >)(Pdefeat'(i) = \{(P_2, P_4), (P_4, P_2), \\
 & \quad (P_4, P_7), (P_7, P_4)\}, Pdefeat'(j) = \{(P_4, P_2), (P_4, P_7)\}) = \{P_2, P_4, P_7\}
 \end{aligned}$$

The third subgoal is to evaluate preferred extensions of following $PRAF_{NPO}$.

$$\begin{aligned}
 & P^{\{i,j\}}(< G^{Agents}(PRAF_{NC}) \cap \overline{P^{Agents}(PRAF_{PPO})}, Agents, Pdefeat'' >) \\
 &= P^{\{i,j\}}(< \{P_8\}, \{i, j\}, Pdefeat'' >)(Pdefeat''(i) = Pdefeat''(j) = \emptyset) = \{P_8\}
 \end{aligned}$$

The fourth subgoal is to evaluate preferred extensions of following $PRAF_{SW}$.

$$\begin{aligned}
 & P^{\{i,j\}}(< P^{Agents}(PRAF_{PPO}) \setminus \overline{P^{Agents}(PRAF_{NPO})}, \{i, j\}, Wdefeat' >) \\
 &= P^{\{i,j\}}(< \{P_4, P_7\}, \{i, j\}, Wdefeat' >)(Wdefeat'(i) = Wdefeat'(j) = \{(P_4, \\
 & \quad P_7)\}) \\
 &= \{P_4\}
 \end{aligned}$$

Therefore, we can conclude that using Lufthansa is supported by justified, Pareto optimal and social welfare maximizing reasons. Note that argument-based reasoning consisting of $PRAF_{PPO}$, $PRAF_{NPO}$ and $PRAF_{SW}$ concludes that using ANA is supported by Pareto optimal, social welfare maximizing, and nonmonotonic consequence reasons.

7 Conclusions and Future Work

This paper proposed argument-based reasoning for actions supported by believable reasons in terms of nonmonotonic consequences and desirable reasons in terms of Pareto optimal and social welfare maximizing reasons. Using the language of modal logic with defaults and inference rules for practical reasoning, we defined three kinds of structured practical argumentation frameworks, i.e., for nonmonotonic consequences, for Pareto optimality, and for social welfare. We showed that a layered practical argumentation framework allows us to evaluate actions supported by Pareto optimal, social welfare maximizing, and nonmonotonic consequence reasons. Our contribution is to extend argument-based reasoning so that it can formally handle reasoning about social desirability by benefiting from economic notions. We, however, do not focus on alternative notions giving criteria for social desirability, e.g., compensation principle. We are now addressing the notions with argument-based reasoning from the viewpoint that argumentation is a decision method.

References

1. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming, and n -person games. *Artificial Intelligence* 77, 321–357 (1995)
2. Kido, H., Nitta, K.: Practical argumentation semantics for socially efficient defeasible consequence. In: Proc. of The Tenth International Conference on Autonomous Agents and Multiagent Systems, pp. 267–274 (2011)
3. Shafir, E., Simonson, I., Tversky, A.: Reason-based choice. *Cognition* 49, 11–36 (1993)
4. Rahwan, I., Larson, K.: Pareto optimality in abstract argumentation. In: Proc. of The 23rd National Conference on Artificial Intelligence, pp. 150–155 (2008)
5. Rosenschein, J.S., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation among Computers. The MIT Press (1994)
6. Wooldridge, M.: An Introduction to MultiAgent Systems, 2nd edn. John Wiley & Sons (2009)
7. Dubois, D., Prade, H., Sabbadin, R.: Decision-theoretic foundations of qualitative possibility theory. *European Journal of Operational Research* 128, 459–478 (2001)
8. Prakken, H.: A study of accrual of arguments, with applications to evidential reasoning. In: Proc. of The 10th International Conference of Artificial Intelligence and Law', pp. 85–94 (2005)
9. Bench-Capon, T.J.M., Prakken, H.: Justifying actions by accruing arguments. In: Proc. of The First International Conference on Computational Models of Argument, pp. 247–258 (2006)
10. Reiter, R.: A logic for default reasoning. *Artificial Intelligence* 13, 81–132 (1980)

A Complex Social System Simulation Using Type-2 Fuzzy Logic and Multiagent System

Dora-Luz Flores, Manuel Castañón-Puga, and Carelia Gaxiola-Pacheco

Universidad Autónoma de Baja California,
Baja California, Mexico
`{dflores, puga, cgaxiola}@uabc.edu.mx`
<http://www.uabc.mx>

Abstract. The need of better representation of complex systems, such social systems, has made that the use of new simulation techniques are increasingly accepted, one of these accepted techniques are multi-agent systems. In addition to represent the uncertainty that is required by them, fuzzy logic and particularly type-2 fuzzy logic are being accepted. A system with three different types of agents is presented as case of study, each agent is assigned to a role with specific goals to be achieved in both ways individually and as teams, the success or failure is determined by group performance rather than individual achievement. It is also taken into account the environment or context as another type of agent. Fuzzy inference systems are defined for each of the agents to represent the concepts interpretation.

Keywords: Fuzzy Logic, Uncertainty, Social Simulations, Intelligent Organizations.

1 Introduction

To create a complex system simulation must take into account many aspects of reality. A social system is considered a complex system and is not an easy task to include most of the aspects taken into account that this representation is as real as possible. These systems can not be understood by taking its parts independently. A change in one party may affect one or more parts of the system. The internal and external interaction is really important in the simulation of these complex systems [1].

Furthermore, one of the most important aspects of the objective function of an organism is the need to find a steady state. This state refers to the point where the organization reaches an equilibrium that does not change. Another aspect that could be considered is that the system gets increasingly complex as a result of this process, in a dynamic setting where the relationships among the parts co-evolve and thus jointly become better adapted and capable of affecting their environments. The field of complex adaptive systems is precisely devoted to the study of these last shorts of evolutionary processes, which have been studied in the last two decades and have contributed to the creation of artificial systems, artificial life, and evolutionary computation [2, 3].

With the intentions of creating smarter, more autonomous agents, artificial intelligence techniques can be incorporated into agents, reducing the quantity of expert knowledge needed to obtain satisfactory results. Traditional artificial intelligence approaches are limited in developing autonomous robots and social simulations, however fuzzy logic, neural networks, genetic algorithms, and symbolic processing may be useful as components within the autonomous systems software. In addition, cognitive architectures may also be very useful. There are systems that have the ability to incorporate sensing, reasoning, action, learning, and collaboration. Machine learning is probably the most difficult part of the problem. Collaboration is also a crucial feature of these systems. These types of systems will be most interesting and useful when many of them are networked together. Getting one system to intelligently operate in the real world is extremely challenging, getting several of them to work together is even more difficult [4].

The computational tools are very useful in helping to create models that represent aspects of the real world and most of these models aim to describe and predict phenomena are based on mathematics, discrete and absolute concepts. This tools works in some areas but fails in others it is based on completely different concepts to reality. One such area involves the management of uncertainty that is not based on random variables such as the relationship of an object or concept to another or measurements are not accurate (large, small, slow).

2 Representation of Uncertainty in a MAS

A fuzzy inference system (FIS) is based on logical rules that can work with numeric values or fuzzy input, rules are evaluated and the individual results together to form what is the output fuzzy, then, a numerical value must be passed through a process of defuzzification if its required. Figure 1 shows a block diagram of the classic structure of a FIS.

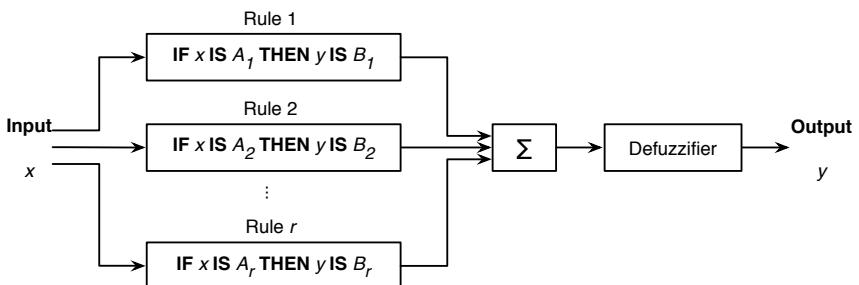


Fig. 1. Fuzzy inference system block diagram

The concept of a type-2 fuzzy set was introduced by Zadeh [5] as an extension of the concept of fuzzy sets usually type 1. A type-2 fuzzy set is characterized

by a membership function whose membership value for each element of the universe is a membership function in the range $[0, 1]$, unlike the type-1 fuzzy sets where the value of membership is a numeric value in the range $[0, 1]$. The creation of a fuzzy set depends on two aspects: the identification of a universe of appropriate values and specifying a membership function properly. The choice of membership function is a subjective process, meaning that different people can reach different conclusions on the same concept. This subjectivity is derived from individual differences in the perception and expression of abstract concepts and very little to do with randomness. Therefore, subjectivity and randomness of a fuzzy set is the main difference between the study of fuzzy sets and probability theory [6].

In type-1 fuzzy sets, once the membership function defined for a concept, this is based on the subjective opinion of one or more individuals and show no more than one value for each element of the universe. In doing so, it loses some of the ambiguity that some concepts are discussed, especially where people may have a slightly different opinion and all are considered valid. The type-2 fuzzy sets allow to handle linguistic and numerical uncertainties. Figure 2 depicts two graphics of fuzzy sets, a) with type-1 fuzzy logic, and b) with type-2 fuzzy logic.

In a) the values set shown is $A = \{(x, \mu_A(x)) | x \in X\}$ where $A \subseteq X$, X is the universe of valid values and $\mu_A(x)$ is the membership function (MF) that contains a map of each value of X with its membership value corresponding to a value between 0 and 1. For b) the values set is $\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u))\}$, where MF $\mu_{\tilde{A}}(x, u)$ has a membership value for each element of the universe as a function of membership in the range $[0, 1]$, so the footprint can be seen around the curve of a).

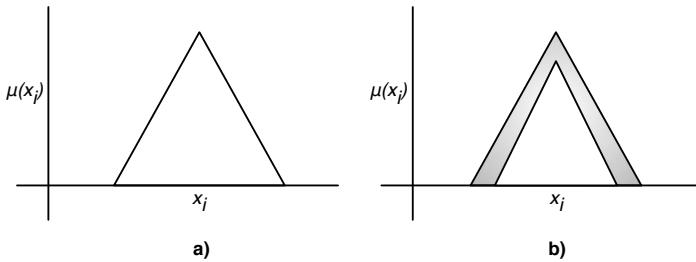


Fig. 2. Type-1 fuzzy set and type-2 fuzzy set with uncertainty

3 Simulation of a Social Complex System

The study case is to simulate a company that manufactures a particular product, there are three different types of agents: managers, workers, and customers. Customers are represented by the environment. When working in a group to accomplish a goal, it is important that each member to fulfill its role and to

share information on concepts of common interest. The success or failure will be determined by the group's performance, not individual achievements. This study case demonstrates how key information can be standardized between groups. The information is interpreted in different ways and different weights are assigned. Other applications use the interpretation of words and the assignment of different weights to each person, this is called computing with words [7–11].

3.1 Interactions between Agents

In this simulation, a manager must first have a record of how many employees (workers) are under his charge prepared to work and capabilities. The ability of employees is measured by the level of fatigue may have, each employee must notify their manager to report these data as soon as possible. Once the manager has a record of employees available, he can send a message to the agent who represents the environment to apply for a job. The agent will respond to environmental specifications for a job and wait for a response of acceptance or rejection.

Depending on the size of work and how many employees are available, a manager must decide if he has the skills to get the job done on time. If work is rejected, then it will lose credibility and minor works. He must decide if it is better to risk not complying with the agreed specifications. If a work is accepted, the manager will divide the work among the employees taking into account their capabilities, this can result in a greater burden on employees in better condition.

As an agent manager has knowledge of the conditions of each employee, he may need consideration about how tired are an agent or he can ignore this information. If he choose to assign work to an agent tired, this will cause a committed more mistakes and can take more work than planned. An agent employee to perform their assigned task the best capabilities possible, but it also has several activities that can have higher priority. Depending on the profile of an agent, the priority of activities can lead to behavior from workaholic to lazy. When an agent is in a weakened state, then their likelihood of errors increases significantly and may decide to stop working for a certain time while they recover.

Once the employee terminates the agent assigned to him, send a message to their manager to report that the task was completed. When the manager receives enough completed tasks to meet the target, this information is communicated to the environment agent and he qualifies the work based on the time it took. Once an assigned task is concluded, the cycle repeats itself again to be a work order to the environment by the manager but under different conditions because the workers have had wear and perception of the environment to the manager has been altered.

Figure 3 shows a simplified case consisting of an interacting agent of each type. It is always advisable to have more than one employee because a single worker would get tired very quickly and would have to reject many jobs before they can accept more and these can only be accepted if they are small or there

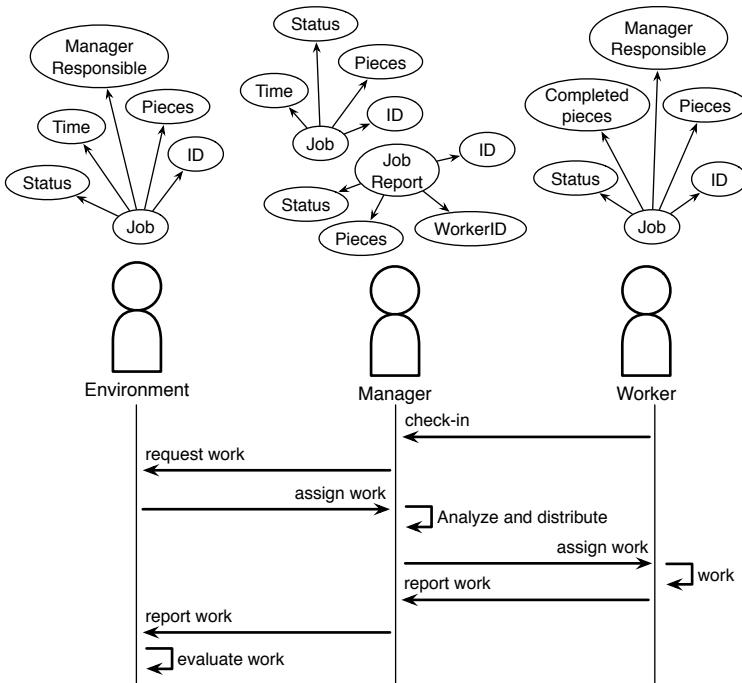


Fig. 3. Study case, messages flow and information structure

is a long time to complete. What is the top of every agent is the structure of the concept of work each one. As can be seen, there are shared concepts which allows to understand but can be given different interpretations and importance.

3.2 Use of Type-2 Fuzzy Logic

Concepts such as large/small or fast/slow can be represented as fuzzy sets, thus allowing that there may be slight variations in the definition for common concepts, an example of this can be shown in [12]. When dealing with a fixed set of actions but with different criteria to decide what action to take, you can use fuzzy logic as a viable option to define the behavior profiles. There are many applications where fuzzy logic has been used, for example, a simulation of the bird age-structured population growth based on an interval type-2 fuzzy cellular structure [13], optimization of interval type-2 fuzzy logic controllers using evolutionary algorithms [14], an improved method for edge detection based on interval type-2 fuzzy logic [15], a hybrid learning algorithm for a class of interval type-2 fuzzy neural networks [16], a systematic design of a stable type-2 fuzzy logic controller [17], and an efficient computational method to implement type-2 fuzzy logic in control applications [18]. In our case study, fuzzy rules are added to two types of agents, which will define the priorities.

The environment agent, which represents customers or market conditions, there are two concepts that are represented by fuzzy sets, job size and time required to complete as shown in Table 1. These concepts are evaluated with five fuzzy rules shown in Table 2 and can be encoded in Jess [19]. The FIS yields numerical results in both outputs (JSize and Time) and these are the specifications of a job that is sent to the agent manager.

Unlike some fuzzy inference systems, in this case we use a type-2 fuzzy set rather than a numeric value, as well as other applications are used these fuzzy sets [20, 21]. This allows for uncertainty from the beginning, reflecting the agent perception and not an accurate measurement. The concepts granularity described should be kept to a minimum to get useful results.

Table 1. Fuzzy Inference System Components

	Linguistic variables	Linguistic values
Inputs	Job size (<i>JSize</i>)	<i>small</i> <i>medium</i> <i>large</i>
	Time	<i>urgent</i> <i>notUrgent</i>
Outputs	Job size (<i>JSize</i>)	1 2 3
	Time	<i>fast</i> <i>slow</i>

Table 2. Fuzzy Inference System Rules

IF Job size IS small THEN Job size IS 1
IF Job size IS medium THEN Job size IS 2
IF Job size IS large THEN Job size IS 3
IF Time IS urgent THEN Time IS fast
IF Time IS notUrgent THEN Time IS slow

Taking into account the rules shown in table 2, using the extension developed by Gaxiola et. al. [22], this extension is a type-2 fuzzy inference system which allows to model the uncertainty of the fuzzy sets giving more flexibility to agents are controlled to tolerate small variations in decision-making. It also has the definition of some user functions that facilitate the creation of fuzzy sets, a remarkable change in functionality of the extension is the possibility of the consequent linguistic value with greater value. This form of defuzzification is similar to a zero-order Sugeno FIS and the resulting constants are linguistic values.

Unlike FuzzyJess, the use of this extension does not require a re-engineering of the rules to suit a fuzzy rule and requires no modifications to the standard code Jess interpreter.

The code presented below is written in Jess represents the FIS of the environment agent.

Fuzzy inference system of the environment agent written in Jess

```
(bind ?fis (new Fis))
  (call ?fis addInput "JSize")
    (call ?fis addToInput "JSize" "small"
      (IGaussMT2 ?11 0.15 -0.04 0))
    (call ?fis addToInput "JSize" "medium"
      (IGaussMT2 ?11 0.15 0.48 0.52))
    (call ?fis addToInput "JSize" "large"
      (IGaussMT2 ?11 0.15 1.04 1))
  (call ?fis addInput "Time")
    (call ?fis addToInput "Time" "urgent"
      (IGaussMT2 ?11 0.21 -0.08 0.08))
    (call ?fis addToInput "Time" "notUrgent"
      (IGaussMT2 ?11 0.21 0.92 1.08))
  (call ?fis addOutput "JSize")
    (call ?fis addToOutput "JSize" "1"
      (ITriT2 ?12 0 3 6 2 5 8))
    (call ?fis addToOutput "JSize" "2"
      (ITriT2 ?12 11 14 17 13 16 19))
    (call ?fis addToOutput "JSize" "3"
      (ITriT2 ?12 22 25 28 24 27 30))
  (call ?fis addOutput "Time")
    (call ?fis addToOutput "Time" "fast"
      (IGaussMT2 ?13 7.85 36.92 43.08))
    (call ?fis addToOutput "Time" "slow"
      (IGaussMT2 ?13 7.85 -0.08 6.08))
  (call ?fis addRule "[JSize:small]" "[JSize:1]")
  (call ?fis addRule "[JSize:medium]" "[JSize:2]")
  (call ?fis addRule "[JSize:large]" "[JSize:3]")
  (call ?fis addRule "[Time:urgent]" "[Time:fast]")
  (call ?fis addRule "[Time:notUrgent]" "[Time:slow]")
```

The worker agents also have a FIS which is responsible for defining the action to take based on the level of health they have. Similar to environment agent, can manipulate the parameters of fuzzy variables in order to create different profiles of behavior. Unlike the environment agent, provides a linguistic result rather than numerical. The encoding of the FIS for this type of agents are shown below.

Fuzzy inference system of the worker agent written in Jess

```
(bind ?fis (new Fis))
(call ?fis addInput "Fatigue")
  (call ?fis addToInput "Fatigue" "None"
    (ITrapaT2 ?l1 -0.07 -0.05 0.11 0.2 -0.03 -0.01 0.21
      0.3 0.9))
  (call ?fis addToInput "Fatigue" "Little"
    (ITrapaT2 ?l1 0.31 0.44 0.54 0.61 0.41 0.47 0.57
      0.71 0.9))
  (call ?fis addToInput "Fatigue" "Alot"
    (ITrapaT2 ?l1 0.71 0.86 1.19 1.19 0.80 0.95 1.26
      1.26 0.9))
(call ?fis addOutput "Action")
  (call ?fis addToOutput "Action" "Work"
    (ITriT2 ?l1 0 0.08 0.16 0.05 0.11 0.22))
  (call ?fis addToOutput "Action" "WorkSlow"
    (ITriT2 ?l1 0.4 0.49 0.56 0.45 0.52 0.62))
  (call ?fis addToOutput "Action" "Rest"
    (ITriT2 ?l1 0.8 0.85 0.94 0.86 0.93 1.0))
(call ?fis addRule "[Fatigue:None]" "[Action:Work]")
(call ?fis addRule "[Fatigue:Little]" "[Action:WorkSlow]")
(call ?fis addRule "[Fatigue:Alot]" "[Action:Rest]")
```

No FIS is added to the agent manager, this with the intention that decisions made by this agent are based on accurate measurements. Since this agent decides whether the work can be completed with available workers, such decisions have more to do with arithmetic and probability that ambiguous concepts.

4 Simulation Results

In this study case we have three types of agents and each has its own interpretation of what is work and decision process varies between numerical calculations and fuzzy reasoning. The messages exchanged can contain information such as pieces of knowledge (called facts in Jess), plain text, or even fuzzy sets. Figure 4 displays a screen-shot of interactions that occur between agents when creating a simulation, can be observed three types of agents used.

In this example we define the agents' profiles in a fuzzy way, in the case of workers can be defined as workaholics or lazies, the customer is defined as demanding or accepting. Besides the ambiguous concepts are defined fuzzy too, fatigue of workers, quality of work performed, and urgency of the work, and then each agent has its own interpretation of what each of these concepts.

As shown in the right side of the Figure 4 is the message content of a worker (named Worker3) where the manager reported that it has completed the assignment with all necessary information, such as amount of finished pieces, fatigue level, etc.

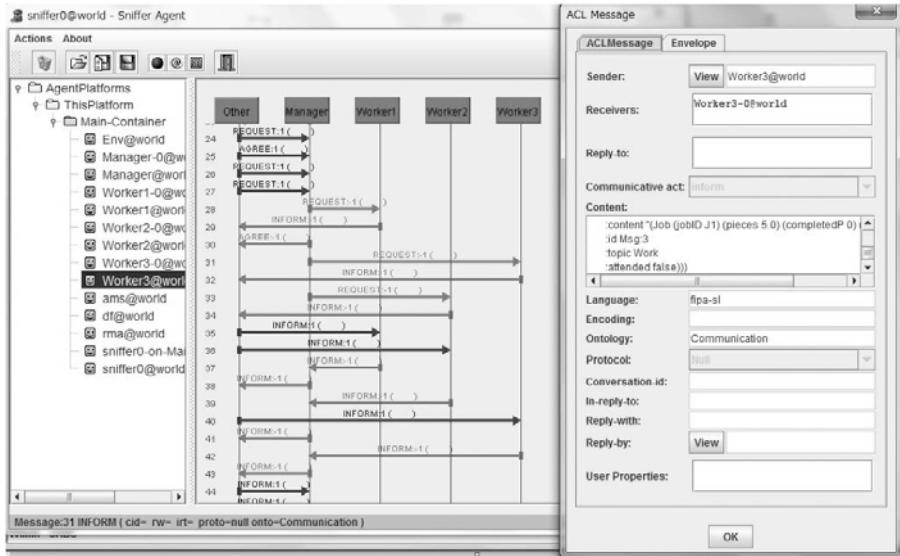


Fig. 4. A screen-shot of a simulation study case, can be observed three types of agents, environment (other), a manager and three workers, and their interactions using Jade

5 Conclusions and Future Work

The world is more complex than can be expressed in text, there are no clear boundaries to classify the significance of actions varies among cultures and vision of the environment depends on perceptions and prior knowledge. For communication to be addressed in a manner consistent with the different social rules, attitudes and nonverbal messages that enrich and change partners that transmit and receive.

In this study case there are three types of agents and each has its own interpretation of what is a job and their decision process varies between fuzzy reasoning and numerical calculations using type-2 fuzzy logic extension. The messages exchanged can contain information such as pieces of knowledge (called facts in Jess), plain text or fuzzy sets. Each agent has its own inference engine to process rules and store the acquired knowledge, thus forming either single agents initial script identical or different.

As future work we can incorporate more artificial intelligence techniques. Of particular interest are the different architectures of neural networks and genetic algorithms. These techniques will improve the autonomy of agents and the experience can be adapted architectures to the field of social and ambiguous since many of these techniques are designed with control applications as its goal.

Besides improving the agent-based model that can be implemented independent nodes, semi-independent and dependent. Dependency ratios would be dynamic and form networks of communities, corporations, governments and any other union.

References

1. Yolles, M.: *Organizations as Complex Systems: An Introduction to Knowledge Cybernetics*. Information Age Publishing, Greenwich (2006)
2. Brownlee, J.: Complex Adaptive Systems Technical Report 070302A (2007)
3. Miler, J., Page, S.: *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*, p. 284. Princeton University Press (2007)
4. Long, L.N., Hanford, S.D., Janrathitikarn, O., Sinsley, G.L., Miller, J.A.: A Review of Intelligent Systems Software for Autonomous Vehicles. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (2007)
5. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning, iii. *Information Science* 8(199249), 301–357 (1975)
6. Jang, J.S., Sun, C.T., Mizutani, E.: *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. MATLAB Curriculum Series. Prentice Hall, Upper Saddle River (1997)
7. Mendel, J.M.: The Perceptual Computer: An Architecture for Computing with Words. In: Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE 2001), Melbourne, Australia, pp. 35–38 (2001)
8. Mendel, J.M.: An Architecture for Making Judgments using Computing with Words. *Int. J. Appl. Math. Comput. Sci.* 12(3), 325–335 (2002)
9. Mendel, J.M.: Computing with Words and its Relationships with Fuzzistics. *Inf. Sci.* 177, 988–1006 (2007)
10. Mendel, J.M., Wu, D.R.: Perceptual Reasoning for Perceptual Computing. *IEEE Transactions on Fuzzy Systems* 16(6), 1550–1564 (2008)
11. Turksen, I.B.: Type-2 Representation and Reasoning for CWW. *Fuzzy Sets Syst.* 127, 17–36 (2002)
12. Wagner, C., Hagras, H.: Fuzzy Composite Concepts based on Human Reasoning. In: IEEE International Conference on Information Reuse and Integration (IRI 2010), Las Vegas, Nevada, USA, p. 308 (2010), doi:978-1-4244-8099-9
13. Leal-Ramírez, C., Castillo, O., Melin, P., Rodríguez-Díaz: Simulation of the bird age-structured population growth based on an interval type-2 fuzzy cellular structure. *Inf. Sci.* 181(3), 519–535 (2011)
14. Castillo, O., Melin, P., Alanis-Garza, A., Montiel, O., Sepúlveda, R.: Optimization of interval type-2 fuzzy logic controllers using evolutionary algorithms. *Soft Computing* 15(6), 1145–1160 (2011)
15. Melin, P., Mendoza, O., Castillo, O.: An improved method for edge detection based on interval type-2 fuzzy logic. *Expert Systems Applications* 37(12), 8527–8535 (2010)
16. Castro, J.R., Castillo, O., Melin, P., Rodríguez-Díaz: A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. *Inf. Sci.* 179(13), 2175–2193 (2009)
17. Castillo, O., Aguilar, L.T., Cázares-Castro, N.R., Cardenas, S.: Systematic design of a stable type-2 fuzzy logic controller. *Applications Soft Computing* 8(3), 1274–1279 (2008)

18. Sepúlveda, R., Castillo, O., Melin, P., Montiel, O.: An Efficient Computational Method to Implement Type-2 Fuzzy Logic in Control Applications. *Analysis and Design of Intelligent Systems using Soft Computing Techniques*, 45–52 (2007)
19. Orchard., R.: Fuzzy Reasoning in Jess: The Fuzzy J Toolkit and Fuzzy Jess. In: *Proceedings of the Third International Conference on Enterprise Information Systems, ICEIS*, pp. 533–542 (2001)
20. Jamme, E., Fleury, M., Wagner, C., Hagras, H., Ghanbari, M.: Interval Type-2 Fuzzy Logic Congestion Control for Video Streaming across IP Networks. *IEEE Transaction on Fuzzy Systems* 17(5), 1123–1142 (2009)
21. Lee, C., Wang, M., Hagras, H.: A Type-2 Fuzzy Ontology and its Application to Personal Diabetic Diet Recommendation. *IEEE Transactions on Fuzzy Systems* 18(2), 374–395 (2010)
22. Gaxiola-Pacheco, C., Flores, D.L., Castañón-Puga, M., Rodríguez-Díaz, A., Castro, J.R., Espinoza-Hernández, I.: Extending Jess with Type-2 Fuzzy Logic. In: *Advances in Soft Computing Science. Research in Computing Science*, vol. (49), pp. 121–129 (2010) ISSN 1870-4069

Computing Mobile Agent Routes with Node-Wise Constraints in Distributed Communication Systems

Amir Elalouf¹, Eugene Levner^{1,2}, and T.C. Edwin Cheng³

¹ Bar Ilan University, Ramat Gan, Israel

amir.elalouf@biu.ac.il

² Ashkelon Academic College, Ashkelon, Israel

levnere@mail.biu.ac.il

³ The Hong Kong Polytechnic University, Hong Kong

edwin.cheng@inet.polyu.edu.hk

Abstract. A basic problem in the quality-of-service (QoS) analysis of multi-agent distributed systems is to find optimal routes for the mobile agents that incrementally fuse the data as they visit hosts in the distributed system. The system is modeled as a directed acyclic graph in which the nodes represent hosts and the edges represent links between them. Each edge is assigned a cost (or benefit) and weights that represent link delay, reliability, or other QoS parameters. The agent scheduling problem is viewed as a constrained routing problem in which a maximum-benefit (or minimum-cost) route connecting the source and the destination subject to QoS constraints is to be found. We study approximation algorithms called ‘fully polynomial time approximation schemes’ (FPTAS) for solving the problem. We suggest an accelerating technique that improves known FPTAS, e.g., Hassin’s (1992); Camponogara & Shima’s (2010); and Elalouf et al. (2011) algorithms, and present new FPTASs.

Keywords: Multi-agent distributed systems, Mobile agent, Agent routing, Routing algorithm, FPTAS, Acceleration technique.

1 Introduction

The problems considered in this paper arise in distributed communication and manufacturing systems in which mobile agents incrementally fuse the data as they visit hosts in the system. Agent-based computing has been appraised as ‘a new revolution in software’ ([1]). More than that, according to Torsun [2], the entire area of distributed artificial intelligence can be viewed as ‘the cooperative problem solving using a federation of collaborating agents’. In the context of distributed communication and manufacturing systems, we define an *agent* as a software system that communicates and cooperates with other human or software systems to solve a complex problem that is beyond the capability of each individual system and such a system is capable of autonomous actions, without the direct intervention of human beings or other agents.

A *multi-agent system* is defined as a loosely coupled network of agents that work together to solve problems that are beyond the individual capabilities or knowledge of

each agent. This definition is compatible with similar definitions given by Jennings and Wooldridge [3] and Shen et al. [4]. An autonomous agent-based distributed manufacturing or communication network is a system that is able to function without the direct intervention of human beings or other agents in some environments, and that has control over its own actions and internal states.

Establishing the notion of *mobile agents* in 1994, White [5] describes a computational environment known as “Telescript” in which running programs are able to transport themselves from host to host in a computer network. Tsichritzis [6] suggests the notion of mobile computation by describing a hypothetical computing environment in which all the objects could be mobile. Within the scope of this paper, we follow the concepts in [5-6] and define a *mobile intelligent agent* (or, briefly, *migent*) as a computer program that is able to autonomously migrate from one host to another in a computer network, *at times and to places of their own choosing*, and to collaborate with each other. These collaborating agents are intelligent in the sense that they are adaptive, autonomous, cooperative, and capable of behaving reasonably in different environments. Each migent behaves in accordance with its own goal and executes its tasks in cooperation with other agents in order to achieve the goal. Such agents can learn and adapt to changes in dynamic environments. In this context, mobile agents differ from ‘applets’, which are programs downloaded as the result of a user action, then executed from beginning to end on one host.

Mobile agent-based technologies have been used in manufacturing and communication for more than two decades [8-11]. In particular, intelligent agents and multi-agent systems and a myriad of applications have been built in various domains ranging from comparatively small systems such as failure detectors in intranets to complex, mission critical systems such as the military operation control. For instance, in order to ensure the minimum power consumption and to decrease the communication bandwidth, the mobile agents are widely used in distributed sensor networks wherein they selectively visit the sensors and incrementally fuse the appropriate measurement data [10, 11]. Other typical examples of real-life mobile agents, which are most known today, are cell phones, manned/unmanned aerial/ground vehicles equipped with a sophisticated software, and web-crawlers (a goal-oriented web robots aiming at discovering/ collecting a requested information in the Internet).

In a typical application, a mobile agent visits several hosts in a network in order to complete its mission. The hosts provide information and also activities, services, and inter-agent communications. All the services and information are available at different sites and in different forms, levels of accuracy, and degrees of reliability. A given *benefit function* determines how much benefit each site contributes to an agent’s mission. The benefit function can be defined and measured differently, depending on the application. For example, it can be the information value or cost in monetary units while in other cases it is the data volume retrieved from different hosts. Since different sites may yield information at many different degrees of benefit, the mobile agent should find a best possible itinerary to visit them under resource constraints. The problem of enhancing the efficiency of mobile agents then reduces to the problem of finding resource-constrained extremal paths in a graph. The *mobile agent’s routing*

problem consists in finding an information- or resource-constrained route for each agent in a multi-agent system that provides the best system performance.

In this paper we investigate different versions of the mobile agent routing problems and develop a general three-stage technique for their efficient solution. Our technique follows and extends the earlier computational scheme suggested by Gens and Levner for the knapsack problem [12, 13]. This three-stage scheme, being originated in the early 1980s, gave excellent possibilities for constructing the FPTASs for solving the constrained routing problems (which has been done by Hassin [14], Goel et al. [15], Xue et al. [16], Lorenz and Raz [17], and Ergun et al. [18], among many others). Recently, using a similar technique, Elalouf et al. [7] improved the routing algorithms by Camponagaro and Shima [19] and Hassin [14] and, also, obtained new fast algorithms for several single-constrained routing problems.

The contribution of this paper is two-fold. First, we introduce more general (and more practical) mobile agent routing problems containing multiple node-wise constraints and imposing QoS requirements upon all intermediate nodes in a route. Second, we further develop and improve the three-stage FPTAS technique suggested in [7]. In the next section we define the problems. In Section 3 we suggest new FPTAS algorithms. Section 4 concludes the paper.

2 Problem Formulation

Following Wu et al. [11] and Camponogara and Shima [19], we address the routing problem for mobile agents aimed to fuse data in distributed networks. Rather than collecting data at a single site, the mobile agent traverses the network and fuses data. Its goal is to perform the information-collection process as efficient as possible. There exist several basic formulations of such quality-of-service-oriented optimization problems each one being dependent, on the one hand, on the form of the objective function (either maximization or minimization) and on the sign of inequality in the constraints (either \leq , or \geq), on the other hand. The first problem to be considered is the following:

Problem P1. To maximize the total benefit generated from agent travel subject to the condition that the total transit time (also called a delay) does *not exceed* a given threshold. Such a problem having a single constraint limiting the delay at the destination node only has been studied by Camponogaro and Shima [19].

The problem framework is based on a network (acyclic graph) $G = (V, E)$ consisting of a set V of nodes (representing the sites), a set E of edges (transmission links between the sites), a start node s , and a destination node t . Nodes in G are denoted by i ($i = s, 2, \dots, t$); edges in G are denoted by (i,j) ; $|V| = n$ and $|E| = m$.

Edge (i,j) is assigned integer parameters b_{ij} and r_{ij} , b_{ij} being the benefit of traversing the edge; specifically, it is the gathered information attributed to the edge; while r_{ij} being the transit time (delay) during agent's move from node i to a next node j in a route, R is a given threshold (the total delay permitted at the destination). Benefit $b(p)$ of a route p from the given node s to the given node t is defined as the

sum of the benefits of the edges in the route; the delay $r(p_k)$ in a sub-route p_k from s to any node k is defined as the sum of the delays at the edges of the sub-route.

In the above notation, Problem P1 can be cast as follows:

$$\text{Problem P1. Maximize } b(p) \equiv \sum_{(i,j) \in p} b_{ij}, \text{ s.t. } r(p) \equiv \sum_{(i,j) \in p} r_{ij} \leq R.$$

We consider more general agent routing problems in which, instead of a single constraint related to the final node in a route, we introduce multiple constraints related to each node in the route.

For instance, consider a route $p = (s, v_1, \dots, v_h = t)$ from s to t ; here h denotes the number of links (called "hops") in route p . A node-wise generalization of P1, denoted by PN1, is constructed as follows. Denote by p_k a partial path from s to $k \in V$. For each node k , delay threshold R_k defines a time slot permitted to be spent by the agent in any partial path p_k from s to v_k . Exceeding this value may lead to excessive signal energy losses and other path losses (see [11, 19], for details). Delay thresholds R_1, \dots, R_n being known in advance for each node in $G(V, E)$, we require that in any partial path p_k from s to k , $k=1, \dots, h$, delay should be at most R_k :

Problem PN1.

$$\text{Maximize } b(p) \equiv \sum_{(i,j) \in p} b_{ij}, \text{ s.t. } r(p, k) \equiv \sum_{(i,j) \in p(k)} r_{ij} \leq R_k, k = 1, \dots, h. \quad (1)$$

Here $r(p, k) \equiv \sum_{(i,j) \in p} r_{ij}$, the left-hand side of constraint k , $k=1, \dots, h$, denotes the accumulated agent delay at node k in sub-route $p(k)$ of p from s to k , and R_k are the maximum delays that are allowed to be accumulated in node k , $k = 1, \dots, h$.

A FPTAS algorithm for PN1 is described in the next section. In the concluding part of the paper, we will discuss several modifications of PN1, in which the maximum is changed to minimum and \leq in the constraint is substituted by \geq . We believe that the FPTASs for the modified problems can be developed, as well.

3 Algorithm Description

3.1 Dynamic Programming (DP) Algorithm

DP is a main building block of the FPTAS. We start with describing DP for PN1.

Let us associate with each path p a pair (B, R) , where $B=b(p)$ is the path benefit and, respectively, $R=R(p)$, the path delay. We deal with sets $S(k)$ of pairs (B, R) arranged in increasing order of B -values so that every pair in $S(k)$ corresponds to a path from a node s to a node k . In order to restore a path itself corresponding to a pair (B, R) , we define a predecessor node to each node and use a standard backtracking.

If there are two pairs in $S(k)$, $(B1, R1)$ and $(B2, R2)$ such that $B1 \geq B2$ and $R1 \leq R2$, then the pair $(B2, R2)$ is called *dominated* and may be discarded.

Notice that graph G is acyclic (since we are solving the longest-path problem with respect to benefit with non-negative lengths, the problem with cycles does not have a finite solution). Then all the nodes can be topologically ordered (see [21,22] for the details). The exact polynomial time DP algorithm for PN1 is presented in Fig.1.

Algorithm 1. The exact pseudo-polynomial DP algorithm

1. Input. $\{(b(k, j), r(k, j) \mid (k, j) \in E\}; R(k)$.
2. Output: A path with maximum benefit.
3. **Step 1.** [Initialization]. Set $S(0,0) = \{(0,0)\}$.
4. **Step 2.** [Generate $S(1)$ to $S(n)$]
 5. for all $k \in V$
 6. $S(k) \leftarrow \emptyset, T \leftarrow \emptyset$ \$Comment: T,G are temporary sets\$
 7. for each arc e leading from node u to node k (denoted by $u \rightarrow k$)
 8. $G \leftarrow \emptyset$
 9. for each pair $(B, R) \in S(u)$ corresponding to a path from s to u
 10. do if $R + r(u, k) \leq R(k)$ then $G \leftarrow G \cup \{(B+b(u, k), R+r(u, k))\}$
 11. Endfor
 12. $T \leftarrow \text{merge}(T, G)$; during merging eliminate the dominated pairs
 13. endfor
 14. $S(k) \leftarrow T$
 15. end for
16. **Step 3.** Find a pair with max B in $S(t)$. Use backtracking to find the optimal path.

Fig. 1. The exact pseudo-polynomial DP algorithm

Lemma 1. The complexity of DP is $O(mUB)$ where the *upper bound* UB is the total benefit of the max-benefit route from s to t .

Proof. Since the benefits are integer and we discard dominated pairs, there is at most UB pairs in $S(k)$ for any k . Further, constructing each G in Step 2 lines 8-11 requires $O(UB)$ elementary operations. Indeed, in line 12, merging the sorted sets G and T , as well as discarding all the dominated pairs, are done in linear time (in the number of pairs, which is at most UB). In Step 2 we have a loop that goes over all the arcs, so in total we have $O(m)$ iterations of lines 8-12. Thus, Algorithm 1 has the complexity of $O(m UB)$ in total. \square

3.2 General Description of the FPTAS

Our general approach for constructing a FPTAS basically follows the so-called *interval partitioning* scheme first suggested by Sahni [20] for the min-cost knapsack problem and further developed by Gens and Levner [12, 13]. The main reason why we have used this approach is that the algorithmic scheme for partitioning routes is similar to that for the knapsack and it yields the desirable results. We suggest the improved scheme that consists of three main stages:

We construct the FPTAS for the considered problem in three stages.

Stage A: Find a lower bound LB and an upper bound UB such that $UB/LB \leq n$.

Stage B: Improve the UB/LB ratio to $UB/LB \leq 2$ using algorithms *BOUNDS* of complexity $O(1)$ described below.

Stage C: Having obtained an improved bound ratio, find an ϵ -approximate solution using the ϵ -approximation algorithm *AA(LB, UB, ϵ)*.

We implement this scheme by designing a FPTAS for PN1. Similar FPTAS algorithms for other routing problems can be derived along the same lines.

3.3 Stage A: Finding Preliminary Lower and Upper Bounds for PN1

A directed path $p = (s, i_1, \dots, i_h = t)$ from node s to node t (denoted as the s - t path) is called *feasible* if all its sub-paths, from s to any k , satisfy (1), that is,

$$r(p, k) \equiv \sum_{(i,j) \in p(k)} r_{ij} \leq R_k, \quad k = 1, \dots, h.$$

A forthcoming sub-procedure that we will use is the following *reaching algorithm* that propagates shortest distances from each (topologically ordered) node to higher-indexed nodes. This algorithm works as follows.

Set label $d(s) = 0$ and labels $d(i) = \infty$ for any other node i . Then examine the nodes in topological order. For each node w , scan all the arcs in the adjacency set $A(w)$, which contains the arcs emanating out from node w . If for any arc (i, j) from $A(w)$ we find that: (1) $d(j) > d(i) + r_{ij}$, and (2) $d(i) + r_{ij} \leq R_j$, then set $d(j) = d(i) + r_{ij}$. Notice that condition (2) distinguishes our algorithm from standard reaching algorithms known in the literature (see, e.g., [21, 22]). When we have examined all the nodes once in this order, any finite label $d(i)$ equals the minimum distance from s to i . If the final label $d(i) = \infty$ at a node i , this means that there is no feasible route to this node.

In what follows, along with notation (i, j) we will use symbol e to denote a current arc of G . For a fixed arc e_q ($1 \leq q \leq m$), define the *constrained graph* G_q as a graph that compulsorily contains an arc of benefit b_{e_q} . This graph is needed in order to find a shortest (with respect to delays) path in G from s to t which is forced to pass through the $e_q = (i', j')$. Let us denote this path by $p^*(i', j')$. In order to find it, we can use the reaching algorithm that finds the minimum-delay sub-path from s to i' in G , adds arc (i', j') to the obtained sub-path and then finds for the shortest path from j' to t through the corresponding arcs of G . We will call the arc (i, j) *right* if the obtained path $p^*(i, j)$ is feasible.

Now we can describe Stage A. It contains the following steps.

Step A1. Find h , the number of hops in the longest (with respect to hops) s - t path in graph G . For this aim, we use the standard algorithm for finding the longest s - t path in the acyclic directed graph G (see, e.g. [21, 22]).

Step A2. Order the arcs of G according to decreasing distinct benefit values b_e :

$$b_1 > b_2 > \dots > b_M, \text{ where } M \leq m. \quad (2)$$

Step A3. In the order defined by (2), define the constrained graphs G_q . Find a maximum index q^* such that graph G_{q^*} has no feasible path but G_{q^*+1} has. (Graph G_q must have a feasible path for at least one q otherwise the problem instance has no feasible solution). We have: $LB = b_{q^*+1}$ and $UB = hb_{q^*+1}$.

The validity of the algorithm is verified by the following

Lemma 2. Let b_{q^*+1} be the maximum benefit among all the right arcs of G , and e_{q^*+1} the corresponding arc. Then $b_{q^*+1} \leq b^* \leq hb_{q^*+1}$.

Proof. Since graph G_{q^*+1} has a feasible path passing through e_{q^*+1} and its total benefit is not smaller than b_{q^*+1} , then the total maximal benefit b^* must be larger or equal to this value: $b^* \geq b_{q^*+1}$. On the other hand, the optimal p^* has at most h arcs, each having benefit at most b_{q^*+1} , thus the optimal path benefit is at most hb_{q^*+1} . So, $b_{q^*+1} \leq b^* \leq nb_{q^*+1}$. \square

Lemma 3. The complexity of Stage A for PN1 is $O(m^2)$ time.

Proof. Step A1 runs in $O(m+n)$ time (see [21, 22]. Step A2 requires $O(m \log n)$ for sorting the arcs ; Step A3 requires $O(m^2)$ time in order to find shortest paths in at most m constrained graphs. In total, Stage A requires $O(m^2)$ time. \square

3.4 Stage B: Finding Improved Bounds for PN1

This stage has two building blocks: a test procedure and BOUNDS.

3.4.1 Test procedure

Stage B contains two building blocks: $\text{Test}(v, \varepsilon)$ and procedure BOUNDS.

$\text{Test}(v, \varepsilon)$ is a parametric dynamic-programming type algorithm has the following property: Given positive parameters v and ε , it either reports that the maximum possible benefit $b^* \geq v$, or, otherwise, reports that that $b^* \leq v(1+\varepsilon)$.

Actually, $\text{Test}(v, \varepsilon)$ will be repeatedly applied as a sub-procedure in the algorithm BOUNDS below to narrow the gap between UB and LB until $UB/LB \leq 2$. $\text{Test}(v, \varepsilon)$ differs from many similar testing procedures known in the literature. Namely, to the best of our knowledge, all the earlier ε -tests for the constrained routing problems were based on the rounding and scaling operations (see, e.g., [14-18]) whereas our test is based on partitioning of the data.

Associated with each path p is a pair (B, R) , where R is the delay $R(p)$ and, correspondingly, $B = B(p)$, the path benefit. We deal with sets $S(k)$ ($k = 1, \dots, n$) of pairs (B, R) , defined above in the DP, arranged in increasing order of R values so that every pair in $S(k)$ corresponds to a path from s to a node k . As in DP we delete all the dominated pair in all $S(k)$.

Given a δ , if there are two pairs in $S(k)$, $(B1, R1)$ and $(B2, R2)$, such that $0 \leq B2 - B1 \leq \delta$, then the pairs are called δ -close. We use the operation called discarding δ -close from set $S(k)$, which means the following:

(a) Let v be a given parameter satisfying $LB \leq v \leq UB$. In each $S(k)$, partition the interval $[0, v]$ into $\lceil n/\varepsilon \rceil$ equal subintervals of size no greater than $\delta = \varepsilon v/n$; one more interval is $[v, UB]$

(b) if more than one pair from $S(k)$ falls into any one of the above subintervals, then discard all such δ -close pairs, leaving only one representative pair in each subinterval, namely the pair with the smallest (in this subinterval) R-coordinate.

Notice that in interval $[v, UB]$ we leave only one representative pair, even if this interval is greater than δ . Algorithm Test(v, ϵ) is presented in Fig. 2.

Lemma 4. The complexity of Test(v, ϵ) is $O(mn/\epsilon)$.

Proof: Since in Test(v, ϵ) we partition interval $[0, v]$ into $\lceil n/\epsilon \rceil$ subintervals, we have $O(n/\epsilon)$ representatives within each set $S(k)$. Since we are going over all m arcs, the algorithm has to spend $O(mn/\epsilon)$ time in total. \square

Algorithm 2. Test(v, ϵ)

Input. $\{(b(k, j), r(k, j) \mid (k, j) \in E\}; R(k), v, \epsilon, \delta = \epsilon v/n$

Step 1. [Initialization].

Set $S(0,0) = \{(0,0)\}$,

Step 2. [Generate $S(1)$ to $S(n)$]

for all $k \in V$

$S(k) \leftarrow \emptyset, T \leftarrow \emptyset$ [Remark: T, G are temporary sets]

for each arc e leading from node u to node k (denoted by $u \rightarrow k$)

$G \leftarrow \emptyset$

for each pair $(B, R) \in S(u)$ corresponding to a path from s to u

do if $R + r(u, k) \leq R(k)$ then $G \leftarrow G \cup \{(B + b(u, k), R + r(u, k))\}$

Endfor

$T \leftarrow \text{merge}(T, G)$; during merging eliminate the dominated and δ -close pairs

endfor

$S(k) \leftarrow T$

end for

Step 3. Find a path in $S(t)$, such that $B \geq v$.

If such a path is found in $S(t)$, return $b^* \geq v$.

If such a path cannot be found in $S(t)$ return $b^* \leq v(1+\epsilon)$.

Fig. 2. The testing procedure Test(v, ϵ)

3.4.2 The Narrowing Procedure BOUNDS

A narrowing procedure in this section originates from the procedure suggested by Ergun et al. [18] for solving the restricted shortest path problem with one constraint. In spite of the fact that we are solving a different problem, we use the key idea presented by Ergun et al.. Namely, when we run Test(v, ϵ) we choose ϵ to be a function of UB/LB changing from iteration to iteration. We use, for the reader's convenience, the following notation. To distinguish the allowable error (ϵ) in the FPTAS and an iteratively changing error in the testing procedure, we denote the latter by θ ; then, Algorithm 3 in Section 3.4.1 from now on will be named Test(v, θ). The idea is that when UB and LB are far from each other, we choose a large θ ; when UB and LB get closer, we choose a smaller θ . More precisely, just as in Ergun et al. (2002), at each iteration of Test(v, θ), we set $\theta = \sqrt{UB/LB} - 1$ whereas a new v value at each iteration takes $v = \sqrt{LB \cdot UB / (1+\theta)}$.

Although **BOUNDS** for PN1 is similar to the corresponding narrowing algorithm of Ergun et al. [18], we present it in Fig. 3 for the sake of completeness.

Algorithm 3. BOUNDS

```

Input: LB and UB such that UB/LB ≤ n.
Output: LB and UB such that UB/LB ≤ 2
If UB/LB ≤ 2, Goto 7
Set θ = √{UB/LB} - 1
Set v ← √{LB · UB / (1 + θ)}
Run Test(v, θ)
If Test(v, θ) returns b* ≥ v set LB ← v else set UB ← v(1 + θ)
Go to 1
End

```

Fig. 3. Algorithm **BOUNDS**

The complexity of BOUNDS is $O(mn/\varepsilon)$. The proof is just the same as that of Theorem 6 in Ergun et al. [18] and is skipped here. -

3.5 Stage C: The ε -Approximation Algorithm

Associated with each path p is a pair (B, R) , where B is the path benefit $b(p)$ and, correspondingly, $R = R(p)$, the path resource consumption. We deal again with sets $S(k)$ of pairs (B, R) as in the above algorithms DP and Test. As in DP and $\text{Test}(v, \varepsilon)$ we delete all the dominated pairs in all $S(k)$. In order to restore an ε -approximation path itself corresponding to a pair (B, R) , we use a standard backtracking defining a predecessor node to each node in the path.

We use the operation called *discarding δ -close* from set $S(k)$, which means the following:

- (a) In each $S(k)$, partition the interval $[0, UB]$ into $\lceil (UB/LB)(n/\varepsilon) \rceil$ equal subintervals of size no greater than $\delta = \varepsilon LB/n$;
- (b) If more than one pair from $S(k)$ falls into any one of the above subintervals, then discard all such δ -close pairs, leaving only one representative pair in each subinterval, namely the pair with the smallest (in this subinterval) R -coordinate.

The corresponding algorithm is presented in Fig. 4.

Theorem 1. The complexity of AA(LB, UB, ε) is $O(mn/\varepsilon)$. The complexity of the entire three-stage FPTAS for PN1 is $O(mn/\varepsilon + m^2)$.

Proof. Since the sub-interval length is $\delta = \varepsilon LB/n$, we have $O(n(UB/LB)(1/\varepsilon))$ sub-intervals in interval $[0, UB]$. Since there is at most one representative pair in each sub-interval, we have in total $O(n(UB/LB)(1/\varepsilon))$ pairs in any of the sets G , T , and $S(k)$, for any k . Further, constructing each G in lines 10-12 in Fig. 4 requires $O(n(UB/LB)(1/\varepsilon))$

elementary operations. Merging the sorted sets G and T as well as discarding all the dominated and δ -close pairs in line 12 are done in linear (in the number of pairs) time. Since G and T contain at most $n(UB/LB)(1/\varepsilon)$ pairs each, the operations in line 13 are done in $O(n(UB/LB)(1/\varepsilon))$. In lines 6-16 we go over all the arcs, and in total lines 9-12 are operated in $O(m)$ time. Thus, we have $O(mn(UB/LB)(1/\varepsilon))$ elementary operations in total. Since the input UB and LB of AA(LB, UB, ε) satisfies $UB/LB \leq 2$, its complexity is $O(mn/\varepsilon)$. Taking the complexity of finding bound ratio into account, the suggested FPTAS for PH1 runs in $O(mn/\varepsilon + m^2)$ time. \square

It is worth noticing that the suggested three-stage FPTAS not only able to solve PN1 but can be also applied for solving the single-constrained problem P1 (see its formulation in Section 2), which is a special case of PN1. In this case, Step 1 can be implemented in $O(mn)$ time rather than in $O(m^2)$, which improves the total complexity to $O(mn/\varepsilon)$. Thus, the present algorithm improves the earlier known FPTASs for P1 developed by Hassin [14] by a factor of n^2/h , Camponagaro & Shima's [19] by a factor of $\log \log (b^-/b)n/h$ and Elalouf et al.'s [7] by an additive term $\log \log n$.

Algorithm 4. The ε -approximation algorithm AA (LB, UB, ε)

1. Input. $\{(b(k, j), r(k, j) \mid (k, j) \in E\}; R(k), v, \varepsilon, \delta = \varepsilon v/n$
2. Output: Path with benefit app such that $app \geq OPT(1-\varepsilon)$
3. **Step 1.** [Initialization].
4. Set $S(0,0) = \{(0,0)\}$,
5. **Step 2.** [Generate $S(1)$ to $S(n)$]
 6. for all $k \in V$
 7. $S(k) \leftarrow \emptyset, T \leftarrow \emptyset$ 'Remark: T, G are temporary sets'
 8. for each arc e leading from node u to node k (denoted by $u \rightarrow k$)
 9. $G \leftarrow \emptyset$
 10. for each pair $(B, R) \in S(u)$ corresponding to a path from s to u
 11. do if $R + r(u, k) \leq R(k)$ then $G \leftarrow G \cup \{(B+b(u, k), R+r(u, k))\}$
 12. Endfor
 13. $T \leftarrow \text{merge}(T, G)$; during merging eliminate the dominated and δ -close pairs
 14. endfor
 15. $S(k) \leftarrow T$
 16. end for
17. **Step 3.** Find the pair with max B in $S(t)$
18. Use backtracking to find the corresponding path; its benefit is at least $(1-\varepsilon)b^*$.

Fig. 4. The ε -approximation algorithm AA (LB, UB, ε)

4 Discussion and Concluding Remarks

We show that the mobile-agent itinerary problem can be cast as the resource-constrained longest path problem that is amenable to fast and practical algorithms. We derive a new FPTAS algorithm and present an acceleration technique that modifies

and improves on several FPTAS algorithms. The complexity of our algorithm only depends polynomially on the graph size (i.e., m and n) and $1/\epsilon$, and does not depend on the magnitudes of the input data.

As mentioned, there are several obvious modifications of the considered constrained longest path problem PN1. Thus, another problem, denoted as PN2, from a mathematical point of view, is a "mirror" one to PN1, in which the maximum is changed to minimum and \leq in the constraint is substituted by \geq .

Problem PN2. Given the costs c_{ij} and information volumes b_{ij} assigned to the links in the network, minimize the total cost of agent route under the constraint that the total volume of the gathered information in each subsequent node k in a route is not less than a given threshold, B_k . This problem can be cast as follows:

$$\text{Minimize } c(p) \equiv \sum_{(i,j) \in p} c_{ij}, \text{ s.t. } b(p, k) \equiv \sum_{(i,j) \in p(k)} b_{ij} \geq B_k, k = 1, \dots, h.$$

The third problem is formally "complementary" to PN1:

Problem PN3. Given the data volume (e.g., the number of useful pages) b_{ij} and information value m_{ij} (e.g., the value in monetary form) assigned to each link in the network, maximize the total number of pages collected by the agent, subject to the condition that the total value of the gathered information in each node k is not less than a given threshold M_k .

$$\text{Maximize } b(p) \equiv \sum_{(i,j) \in p} b_{ij}, \text{ s.t. } m(p, k) \equiv \sum_{(i,j) \in p(k)} m_{ij} \geq M_k, k = 1, \dots, h.$$

The next problem, formally speaking, is a "mirror" problem to PN3:

Problem PN4. Given costs c_{ij} and expected travel times t_{ij} assigned to each link in a network, minimize the total expected travel time for completing the task under the constraint that the total route cost in node k does not exceed a given budget C_k .

$$\text{Minimize } t(p) \equiv \sum_{(i,j) \in p} t_{ij}, \text{ s.t. } c(p, k) \equiv \sum_{(i,j) \in p(k)} c_{ij} \leq C_k, k = 1, \dots, h.$$

We believe that the general FPTAS technique described in this paper can be applied to solve the above and even more complicated multi-path, multi-constraint agent routing problems in distributed systems. We believe that the complexity of the algorithms presented in this work can be further improved. While in the present paper we look separately for a best route for each individual agent, a prospective direction for future research is to find a best schedule for several cooperating agents. Another research direction is to carry out extensive computational experiments to assess the performance of our algorithms in comparison with earlier known algorithms and to test the possibility of applying them in practice.

Acknowledgments. The authors wish to thank Prof. Magnus M. Halldorsson for his helpful suggestions that helped improve the paper.

References

1. Guilfoyle, C., Warner, E.: Intelligent Agents: New Revolution in Software, p. 214. Ovum Ltd. Publisher, London (1994)

2. Torsun, I.S.: Foundations of Intelligent Knowledge-Based Systems, p. 507. Academic Press, London (1995)
3. Jennings, N.R., Wooldridge, M.J.: Applications of Intelligent Agents. In: Jennings, N.R., Wooldridge, M.J. (eds.) Agent Technology: Foundations, Applications and Markets, pp. 3–28. Springer, Heidelberg (1998)
4. Shen, W., Norrie, D.H., Barthes, J.-P.: Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing, p. 386. Taylor and Francis, London (2001)
5. White, J.E.: Telescript Technology: The Foundation for the Electronic Marketplace, White Paper. General Magic, Inc., USA (1994)
6. Tsichritzis, D.: Objectworld, Office Automation. Springer, Heidelberg (1985)
7. Elalouf, A., Levner, E., Cheng, T.C.E.: Efficient Routing of Mobile Agents for Agent-based Integrated Enterprise Management: A General Acceleration Technique. LNBP, vol. 88, pp. 1–20. Springer, Berlin (2011)
8. Papaioannou, T.: Using Mobile Agents to Improve the Alignment between Manufacturing and its IT Support Systems. In: Robotics and Autonomous Systems. Elsevier (1999)
9. Peng, Y., Finin, T., Labrou, Y., Chu, B., Long, J., Tolone, X., Boughannam, A.: A Multi-Agent System for Enterprise Integration. In: Proceedings of PAAM 1998, London, UK, pp. 155–169 (1998)
10. Qi, H., Iyengar, S.S., Chakrabarty, K.: Multi-Resolution Data Integration Using Mobile Agents in Distributed Sensor Networks. IEEE Trans. Systems, Man, and Cybernetics Part C: Applications and Rev. 31(3), 383–391 (2001)
11. Wu, Q., Rao, N.S.V., Barhen, J., Iyengar, S.S., Vaishnavi, V.K., Qi, H., Chakrabarty, K.: On Computing Mobile Agent Routes for Data Fusion in Distributed Sensor Networks. IEEE Transactions on Knowledge and Data Engineering 16(6), 740–753 (2004)
12. Gens, G.V., Levner, E.V.: Fast Approximation Algorithms for Job Sequencing with Deadlines. Discrete Applied Mathematics 3, 313–318 (1981)
13. Gens, G.V., Levner, E.V.: Fast Approximation Algorithms for Knapsack Type Problems. LNCIS, vol. 23. Springer, Berlin (1980)
14. Hassin, R.: Approximation Schemes for the Restricted Shortest Path Problem. Mathematics of Operations Research 17(1), 36–42 (1992)
15. Goel, A., Ramakrishnan, K.G., Kataria, D., Logothetis, D.: Efficient Computation of Delay-sensitive Routes from One Source to All Destinations. In: IEEE Infocom 2001, pp. 854–858. IEEE Press (2001)
16. Xue, G., Sen, A., Zhang, W., Tang, J., Thulasiraman, K.: Finding a Path Subject to Many Additive QoS Constraints. IEEE Transactions on Networking 15, 201–211 (2007)
17. Lorenz, D.H., Raz, D.: A Simple Efficient Approximation Scheme for the Restricted Shortest Path problem. Operations Research Letters 28(5), 213–219 (2001)
18. Ergun, F., Sinha, R., Zhang, L.: An Improved FPTAS for Restricted Shortest Path. Information Processing Letters 83(5), 287–291 (2002)
19. Camponogara, E., Shima, R.B.: Mobile Agent Routing with Time Constraints: A Resource Constrained Longest-Path Approach. Journal of Universal Computer Science 16(3), 372–401 (2010)
20. Sahni, S.: Algorithms for Scheduling Independent Tasks. Journal of the ACM 23(1), 116–127 (1976)
21. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, ch. 24.2. MIT Press (2001)
22. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows. Prentice Hall, New Jersey (1993)

Collaborative Redundant Agents: Modeling the Dependences in the Diversity of the Agents' Errors

Laura Zavala¹, Michael Huhns², and Angélica García-Vega³

¹ Computer Science, University of Maryland Baltimore County, Baltimore, MD, USA
rzavala@umbc.edu

² Computer Science, University of South Carolina, Columbia, SC, USA
huhns@sc.edu

³ Facultad de Física e Inteligencia Artificial, Universidad Veracruzana, Xalapa, Ver., México
angegarcia@uv.mx

Abstract. As computing becomes pervasive, there are increasing opportunities for building collaborative multiagent systems that make use of multiple sources of knowledge and functionality for validation and reliability improvement purposes. However, there is no established method to combine the agents' contributions synergistically. Independence is usually assumed when integrating contributions from different sources. In this paper, we present a domain-independent model for representing dependences among agents. We discuss the influence that dependence-based confidence determination might have on the results provided by a group of collaborative agents. We show that it is theoretically possible to obtain higher accuracy than that obtained under the assumption of independence among the agents. We empirically evaluate the effectiveness of a collaborative multiagent system in the presence of dependences among the agents, and to analyze the effects of incorrect confidence integration assumptions.

Keywords: Multiagent Systems, Collaboration, Data Fusion.

1 Introduction

With the increasing availability of sources of information and services, and the involvement of people and organizations as contributors, there are also increased needs and opportunities for the use of collaborating multiagent systems that make use of multiple sources of knowledge for validation and improvement of accuracy and reliability. Potential application examples are situational aware systems [18], sensor networks [12], and semantic sensor web [15]. A common task in these systems is the integration of contributions, which is not straightforward. Common techniques for integrating contributions are based on agent reputation models [7] and the assumption of independence among the agents, such as majority voting and weighted majority voting techniques [11,14]. The dependences among agents are usually not considered despite their crucial role. Consider, for example, a group of three weather expert agents who have the task of collectively producing a weather forecast. Each agent has a reputation (probability of being accurate) of 0.7 which has been asserted by looking at their past performance. Suppose that the particular task is to predict whether or not it is going to rain. How can the contributions of these agents be combined to achieve the most accurate possible

result and outperform any expert individual performance? Clearly, this is possible only if the agents compensate for the mistakes, limitations, and vulnerabilities of each other. Suppose, in our example, we get a positive result (“it is going to rain”) from the three agents, what confidence should we give to the integrated result (“it is going to rain”)? Common approaches used when combining contributions are based on the assumption of independence among the contributions and either use an ad hoc technique that makes use of the agent’s reputations to determine the confidence, or do not determine a confidence for the integrated contributions at all (select the “best” contribution by some rule like majority). Assuming independence of contributions, the best way to maximize the probability of an accurate team prediction is to use Bayes rule to determine the confidence of each contribution. In our example, there is only one contribution: “it is going to rain”, and the confidence assigned using Bayes rule would be 0.927. However, if all the agents employed the same program to determine rain likelihood, they would fail or succeed identically and the confidence in the integrated result should remain at 0.7. On the other hand, if we have a guarantee that the agents never fail together, the confidence in the integrated result should be 1. There is a confidence uncertainty inherent when integrating the agents’ contributions, which, in our example, sets an interval from 0.7 to 0.927 for the confidence in the result, with any additional knowledge of the dependence relation among the agents reducing this confidence interval.

We have proposed[22] the use of multiagent-based redundancy as a basis for robust software and Web services development. In this paper, we focus on the issue of integrating agents’ contributions. Particularly, we study the influence that dependence-based confidence determination might have on the results provided by a group of collaborative agents. In order to achieve that, we have developed a model that allows the representation of the full spectrum of potential dependences between pairs of collaborative agents by considering the space of their coincident errors which can be totally coincident, independent and non-coincident. Because of the impossibility to manage a model representing all potential dependences, the model allows to capture a simplified version using a linear structure where only adjacent nodes are directly dependent on each other and conditional independence is assumed between all nonadjacent nodes—a reasonable simplification that at worst, understates the potential error of assuming full contribution independence. Using the model, we can empirically evaluate the effectiveness of a collaborative multiagent system in the presence of known dependences among the agents, as well as analyze the effects of incorrect confidence integration assumptions.

In the next section we discuss related work. In section 3 we present our formal model for the representation of dependences among agents. Section 4 show the experiments conducted, which show how the model can be used to measure the sensitivity of complex collaborative multiagent systems to incorrect assumptions of independence among the agents. In section 5 we present our conclusions and discuss our plans for the integration of the work presented in this paper with trust and reputation networks.

2 Related Work

Voting algorithms have been used to provide an error masking capability in a wide range of commercial and research applications. There is a large compendium of software

systems implementations and experiments reporting some degree of reliability improvement [11,14,22,19,13]. However, all these works usually make use of simple voting strategies and do not address the issue of dependence, relying strongly on the assumption of independence among the versions. A functional classification is used in [9] to provide a taxonomy of the voting algorithms reported in the literature.

In [1] the authors address the issue of dependence and provide a model for analyzing the effect and propagation of potentially incorrect confidence-integration assumptions in a complex MAS with both concurrent and sequential processing in binary scenarios (two possible outputs). Our model is based on [1] but it does not include the sequential case (chains of decisions) and thus, the results are not affected by previous computations. By concentrating in one-time integration cases we are able to observe better the effect of independence assumptions. Furthermore, we have extended the confidence range to include the cases in which it is possible to obtain higher accuracy than that obtained under the assumption of independence among the agents. Specifically, we have extended the confidence range to include non-coincidence. In [1] the confidence, say of a system of 3 agents with [.8, .8, .8] reliabilities is anywhere between .8 (when they fail identically) and .985 (when their errors are independent). Our model includes one more case for when their errors are mutually exclusive (non-coincident), so that makes the confidence range (in some cases) wider, since the upper bound can reach 1. Finally, our model allows the representation of cases with more than two possible outputs.

There have been several works [16,17,2] in the field of multiagent systems that deal modeling dependencies among agents. Specifically interdependencies among different agents' goals and actions where an agent is said to be dependent on another if the latter can help / prevent him to achieve one of his goals. This type of dependency relations allow an agent to know which of his goals are achievable and which of his plans are feasible (or not) at any moment. In this way, an agent may dynamically choose a goal to pursue and a plan to achieve it, being sure that every skill needed to accomplish the selected plan is available in the system. Our work, however, is focused on another type of dependencies among agents, namely, dependences in the context of the diversity in the agents' errors. Furthermore, the target application of our approach is to collaborative multiagent systems where agents combine their contributions to achieve the most accurate possible result and outperform any expert individual performance.

3 A Model of Dependences for Collaborative Agents

Our model can be applied to collaborative multiagent systems in which all the agents in the system have functionally equivalent capabilities and always do their best to provide accurate contributions for a particular task. They cooperate in order to increase the overall outcome of the system and are not concerned with their personal payoffs. In other words, all of the agents share the same goals or are trying to maximize a social utility instead of an individual, personal utility. The goal is for the system to exhibit better average performance than that of any single agent in the system. More specifically, we make the following assumptions:

- the agents provide equivalent functionality
- the agents' contributions are not always correct

- the probability of an agent's contributions being accurate is known
- agents can be trusted to always do their best to provide accurate contributions
- the agents' contributions have identical semantics, and therefore can be integrated trivially

To analyze the spectrum of dependence between a pair of collaborative agents, we consider the space of their coincident errors. The errors can be: totally coincident (maximally correlated), in which case the agents always fail together if their accuracies are the same, or every time the more accurate agent fails if their accuracies vary; independent, where the probability of coincident errors is given by the product of the agent's individual probabilities of error; non-coincident, in which case the agents never fail together. Let $P(A_i)$ denote the accuracy of agent A_i and $1 - P(A_i)$ its probability of error. There are four different possible outcomes for each pair of contributions provided by different agents. Assume, for example, that we have two agents A_1 and A_2 with $P(A_1) = .9$ and $P(A_2) = .6$. Table 1 shows the four different possible outcomes and provides the number of occurrences out of 100 tests for the three different cases of dependence among A_1 and A_2 .

Table 1. The four different possible outcomes for each pair of contributions provided by agents A_1 and A_2 , and the number of occurrences, out of 100, for each outcome for the three different cases of dependence among the agents. C=Coincident, I=Independent, and NC=Non-Coincident

$P(A_1) = .90$	$P(A_2) = .60$	C	I	NC
fails	fails	10	4	0
fails	succeeds	0	6	10
succeeds	fails	30	36	40
succeeds	succeeds	60	54	50

Figure 1 shows this idea in a graphical representation. We can see what happens as the individual accuracies and the difference between them change. As agent accuracies increase, the contributions naturally become more correlated and the difference between the coincident, independent, and non-coincident extremes becomes smaller. When $P(A_i) = 1.0$ there is no difference. Therefore, if our agents are close to perfect, incorrect dependences assumptions will have less impact in the final result.

A set of decisions made by agents is concurrent if the decisions are made without knowledge of any of the other decisions in the set [1]. Figure 2a illustrates the concurrent decisions of n agents related to node W , which is the state of the world that the agents are trying to match. In our model, W can take on any number of values, though for simplicity we will illustrate it for two values, 1 and 0. In our initial weather agents example, this abstraction could represent $W = \{rain, no-rain\}$. Given a set of n agents as in Figure 2a, there are 2^n subsets or groupings that can be made out of the n agents, and therefore, potential dependences between them. Consequently, a full model representing all the potential dependences between the agents, becomes unmanageable as the number of agents grows. To keep the number of dependences manageable, we simplify the model to the linear structure assumed in [1], where only adjacent nodes are directly dependent on each other and conditional independence is assumed between all

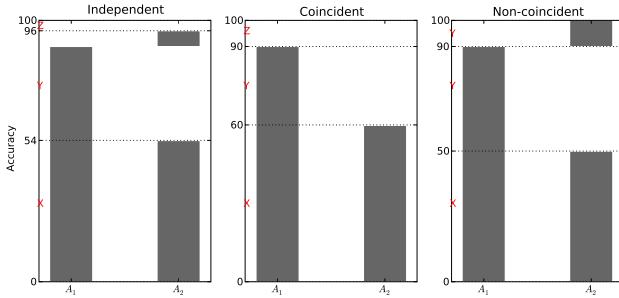


Fig. 1. Dependence relations of two agents. The black bars indicate cases where the agent's result is correct. The percentage of black versus white is the accuracy of the agent. Region X represents where both agents' results are correct; region Y is where their results disagree, and region Z represents where both agents' results are wrong.

nonadjacent nodes—a reasonable simplification that at worst, understates the potential error of assuming full contribution independence. Figure 2b shows the subset of dependences represented in our model. For each pair of nodes with a dependence relation (adjacent nodes) we introduce two nodes, a C node and a NC node, to represent coincident and non-coincident dependence information, respectively, between A_i and A_{i-1} . For simplicity of notation, we will use C_i to refer to the node $C_{i-1 \rightarrow i}$ and NC_i to refer to the node $NC_{i-1 \rightarrow i}$. When both nodes are *false* ($C_i = \text{false}$ and $NC_i = \text{false}$), the errors in A_i and A_{i-1} are independent. In other words, A_i is conditionally independent of A_{i-1} given W , $P(A_i|W) = P(A_i|W, A_{i-1})$.

When $C_i = \text{true}$, the errors between A_i and A_{i-1} are maximally correlated with one another (as in the coincident case in Figure 1). When $NC_i = \text{true}$, the errors between A_i and A_{i-1} are totally exclusive (as in the non-coincident case in Figure 1).

We can assign C_i and NC_i a value between 0 and 1 to represent the range of dependence from fully coincident ($P(C_i = \text{true}) = 1$), to conditionally independent ($P(C_i = \text{true}) = 0$) and ($P(NC_i = \text{true}) = 0$), to non-coincident ($P(NC_i = \text{true}) = 1$).

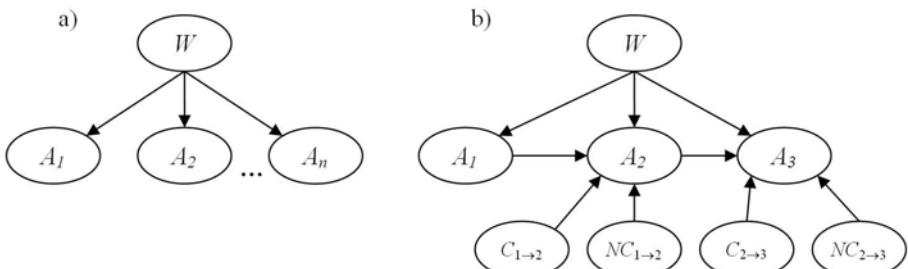


Fig. 2. a) Concurrent decisions of n agents related to node W ; b) The subset of dependences represented in our model

Tables 2 to 4 show the conditional probability tables (CPTs) for agent A_2 in the fully coincident, conditionally independent, and non-coincident cases respectively. For intermediate values of $P(C_i = \text{true})$ or $P(NC_i = \text{true})$ (only one can have an intermediate value, while the other must have a 0 value), $P(A_i|W, A_{i-1})$ is obtained by combining the mixed distributions defined by either one or the other. That is: $P(A_i|W, A_{i-1}) = P(C_i = \text{true})P(A_i|W, A_{i-1}, C_i = \text{true}) + P(C_i = \text{false})P(A_i|W, A_{i-1})$, when $0 < P(C_i = \text{true}) < 1$ or $P(A_i|W, A_{i-1}) = P(NC_i = \text{true})P(A_i|W, A_{i-1}, NC_i = \text{true}) + P(NC_i = \text{false})P(A_i|W, A_{i-1})$, when $0 < P(NC_i = \text{true}) < 1$.

Table 2. Conditional probability table for agent A_2 under the fully coincident model. The errors between A_2 and A_1 are maximally correlated with one another (as in the coincident case in Figure 1).

A_1	W	C_2	1
1	1	1	$\min(\frac{P(A_2)}{P(A_1)}, 1)$
0	0	1	$1 - \min(\frac{P(A_2)}{P(A_1)}, 1)$
1	0	1	$\min(\frac{1-P(A_1)}{1-P(A_2)}, 1)$
0	1	1	$1 - \min(\frac{1-P(A_1)}{1-P(A_2)}, 1)$

Table 3. Conditional probability table for agent A_2 under the independent model. The errors in A_2 and A_1 are independent. In other words, A_2 is conditionally independent of A_1 given W .

A_1	W	0	1
-	0	$P(A_2)$	$1 - P(A_2)$
-	1	$1 - P(A_2)$	$P(A_2)$

Table 4. Conditional probability table for agent A_2 under the non-coincident model. The errors between A_2 and A_1 are totally exclusive (as in the non-coincident case in Figure 1).

A_1	W	NC_2	1
1	1	1	$\frac{P(A_2) - (1 - P(A_1))}{P(A_1)}$
0	0	1	$1 - \frac{P(A_2) - (1 - P(A_1))}{P(A_1)}$
1	0	1	0
0	1	1	1

4 Experiments and Analyses

Our model can be used to measure the sensitivity of complex collaborative multiagent systems to incorrect assumptions of independence among the agents when integrating their contributions. In this section we demonstrate the use of our model with homogeneous and heterogeneous multiagent systems of different sizes and under different scenarios given by real versus assumed dependence relations between the agents.

To determine the accuracy of a particular system configuration, we determine the confidence of the integrated result for each combination of agents' contributions A_i

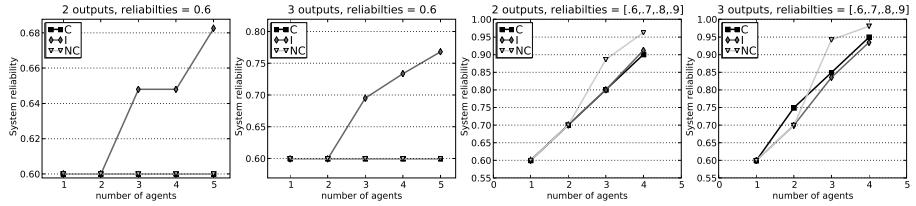


Fig. 3. Reliability of the system for fully coincident (C), independent (I), and non-coincident (NC) agents—homogeneous and heterogeneous cases with 2 and 3 outputs each

in the model. Assuming the system answer is the value of W that maximizes $P(W|A)$, the confidence of an integrated result is then $\max_W P(W|A)$. Since we want to evaluate the sensitivity of a system to particular incorrectly assumed dependence relation, we measure the expected belief that our answer is correct by taking a weighted average over all possible instances of the observed data, $\sum_A P(A) \times \max_W P(W|A)$ and compare it to the expected belief obtained under the assumed dependence relations, $\sum_A P(A) \times \max_W P(W|A, P(C_i), P(NC_i))$.

For simplicity of notation refer to $P(C = \text{true})$ as $(P(C))$ and $P(NC = \text{true})$ as $(P(NC))$. Figure 3 shows the accuracy of homogeneous and heterogeneous systems for fully coincident, independent, and non-coincident agents with 2 and 3 outputs. For homogeneous systems, there is no difference, in terms of accuracy, between fully coincident and non-coincident. The difference in these cases becomes clear when looking at heterogeneous systems.

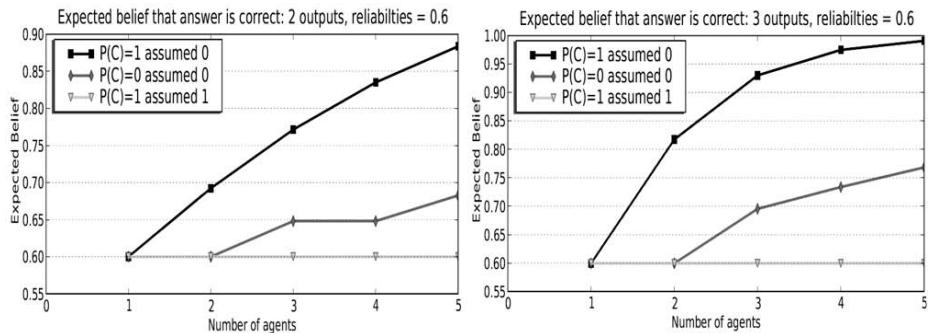


Fig. 4. The effect of assuming agents are independent when they are fully coincident (maximally correlated)

The sensitivity of multiagent systems to incorrect assumptions of independence among the agents is shown in figures 4 to 7. Figure 4 shows the effect of assuming agents are independent when they are fully coincident, for homogeneous systems with

two and three outputs. $P(C) = 0$, *assumed0* shows the expected belief for the case when the agents are conditionally independent which is the usual assumption when applying Bayes rule. But if we make that assumption and the agents are not conditionally independent, the expected belief is overestimated, at worst, as shown by $P(C) = 1$, *assumed0*, which is the case when the agents are fully coincident.

Figures 5 to 7 show the sensitivity of multiagent systems to incorrect assumptions of independence among the agents for heterogeneous systems with two and three outputs. As Figure 2b shows, the model implies a direction of the dependence relations. If the system is homogeneous (agents have the same reliabilities), the direction does not matter. But that is not the case for heterogeneous systems. Therefore, we include two cases in figures 5 to 7. The first case is when more reliable agents depend on less reliable agents. The second case is when less reliable agents depend on more reliable agents.

Figure 5 shows the effect of different dependence assumptions when agents are fully coincident. Specifically, assuming agents are either independent or non-coincident when they are fully coincident. The plots on the top row show the case when more reliable agents depend on less reliable agents. The plot for two outputs shows how when agents are fully coincident, adding more reliable agents improves the expected belief, but it can never be greater than that of the most reliable agent at any particular system configuration (number of agents). The plots on the second row show the case when less reliable agents depend on more reliable agents. The expected belief does not improve adding less reliable agents, and again, it can never be greater than that of the most reliable agent. Finally, the expected belief is underestimated when assuming independence and non-coincidence.

Figure 6 shows the effect of different dependence assumptions when agents are independent. Specifically, assuming agents are either coincident or non-coincident when they are independent. The plots on the top row show the case when more reliable agents depend on less reliable agents. When agents are independent, adding more reliable agents improves the expected belief. Contrary to the case when agents are dependent (Figure 5), the expected belief becomes greater than that of the most reliable agent at a certain system configuration (number of agents). The improvement is bigger for systems with a larger number of outputs. Finally, the expected belief is overestimated when assuming coincidence and underestimated when assuming non-coincidence.

Figure 7 shows the effect of different dependence assumptions when agents are non-coincident. Specifically, assuming agents are either coincident or independent when they are non-coincident. The plots on the top row show the case when more reliable agents depend on less reliable agents. When agents are non-coincident, adding more reliable agents improves the expected belief. As in the independent case (Figure 6), the expected belief becomes greater than that of the most reliable agent at a certain system configuration (number of agents), but the improvement is even greater. The improvement is also bigger for systems with a larger number of outputs. Finally, the expected belief is overestimated when assuming independence and coincidence.

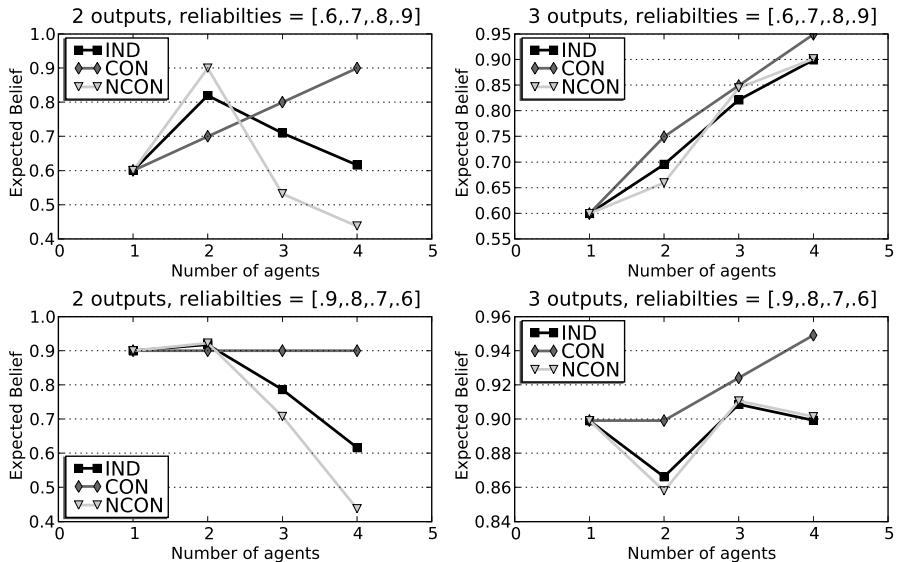


Fig. 5. The effects of assuming agents are either independent (IND) or non-coincident (NCON) when they are fully coincident (CON). When agents are coincident, the expected belief can never be greater than that of the most reliable agent. The expected belief is underestimated when assuming independence and non-coincidence.

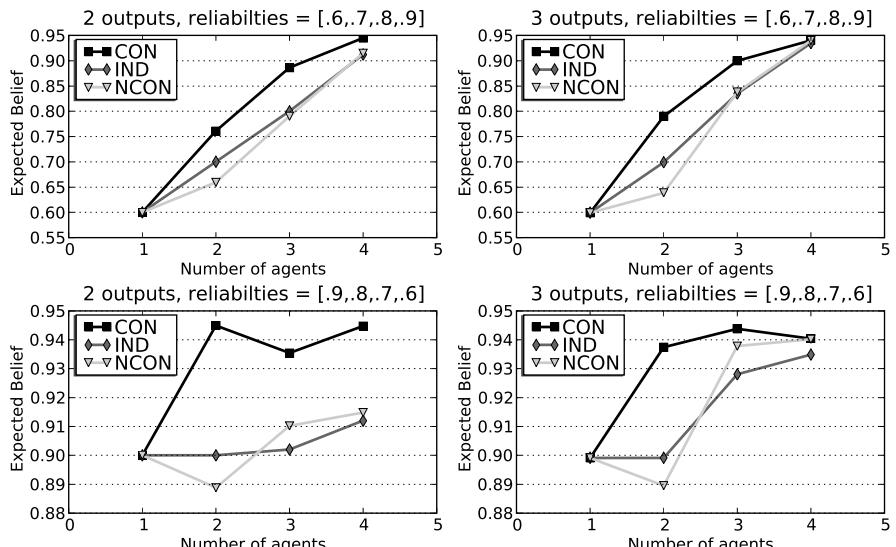


Fig. 6. The effects of assuming agents are either coincident (CON) or independent (IND) when they are independent (IND). When agents are independent, the expected belief becomes greater than that of the most reliable agent at a certain system configuration (number of agents). The expected belief is overestimated when assuming coincidence and underestimated when assuming non-coincidence.

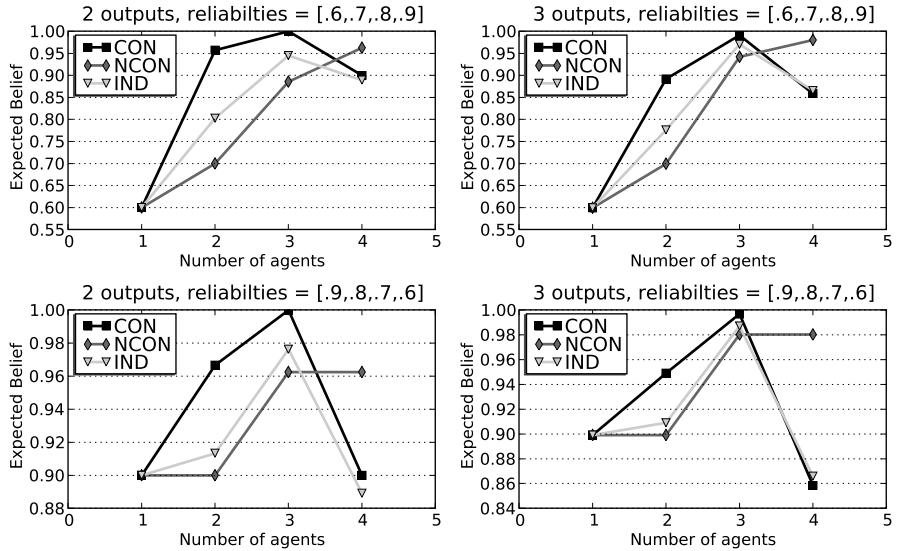


Fig. 7. The effects of assuming agents are either coincident (CON) or independent (IND) when they are non-coincident (NCON)

5 Conclusions and Future Work

We presented a domain independent model for the study of the effect that incorrect assumptions about the dependences among collaborative agents have on the overall system accuracy estimation. We showed results of experiments using our model with homogeneous and heterogeneous multiagent systems of different sizes and under different scenarios given by real versus assumed dependences relation between the agents.

We showed that the accuracy of a collaborative multiagent system varies according to the diversity of the errors of the agents. We also showed that the lower bound on the accuracy is given by the case where the agents errors are maximally correlated; and that the upper bound is not given by the conditional independence case, but by the case where agents errors are non-coincident (never fail together), usually 1.

We did not specify how to obtain the degree of dependence among agents, which can be a difficult task and varies with the application. In software fault tolerance, for example, testing as well as static programs analyses techniques have been used in trying to determine potential dependences among programs[4,20,10,21]. A challenge, as we mentioned earlier, is that the number of potential dependences grows exponentially with the number of agents. Also, multiagent systems are typically open and dynamic, which makes it difficult to measure dependences among the agents. A future work is the study of dynamic and adaptive strategies to estimate and model dependences among the agents, including alternative formalisms , e.g. Dempster-Shafer theory.

Finally, we have discussed our plans to use our work in the area of trust networks and online reputation systems. The dependences in reputation systems can be obtained from the rating similarities between pairs of users.

5.1 Modeling and Using Dependences in Trust and Reputation Networks

We are interested in investigating the relation of our work with trust networks and online reputation systems. In this section we briefly discuss our insights on these tasks which we have planned as a future work.

The benefits of linking our work with trust networks are in both directions, i.e. using trust networks as an instrument in the dependences model, and using the dependences model as an instrument in trust networks. In the first case, a trust network could be used in assigning dependency probabilities among entities (e.g. the same contribution by a group of friends is likely to have some underlying dependences —similarities in the process of generating the contribution). In this case we can say that we use the trust network to navigate between the lower and upper (confidence) bounds generated by the model. In the second case, the dependences model could be used to assist existing trust propagation algorithms [6] in predicting trust between any two nodes in the network.

We plan to use our work on modeling dependences among entities when integrating contributions as a baseline to develop a Bayesian inference-based recommendation approach for online reputation systems[8] such as Flixster and Epinions. The dependences correspond to rating similarities between a pair of users, which can be measured by a set of conditional probabilities derived from their mutual rating history. Integrating contributions corresponds to calculating a rating score for an item based on the ratings (contributions) of the users. Suppose for example, in the electronics domain, a user is considering buying a certain product and wants to know what the general opinion is about it. Other user's rating of the product can be combined as an averaged score or can be better combined using likely dependences among the users that have rated the product to obtain a score that accounts for similarities among users. Further, the dependences of the user who is querying the product with those who have rated the item can also be factored into the result so that the obtained score is biased towards the user's preferences. In the first case we would be calculating an overall (not averaged) score that reflects the reputation of the product in the community, and in the second case we simply give more weight to some users' contributions.

References

1. Becker, R., Corkill, D.: Determining confidence when integrating contributions from multiple agents. In: Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2007), pp. 449–456. The International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS) (May 2007)
2. Conte, R., Castelfranchi, C.: Simulating multi-agent interdependencies. a two-way approach to the micro-macro link. In: Social Science Microsimulation [Dagstuhl Seminar, May 1995], pp. 394–415. Springer, London (1996), <http://portal.acm.org/citation.cfm?id=647378.724484>

3. Corkill, D.D.: Collaborating software: Blackboard and multi-agent systems & the future. In: Proceedings of the International Lisp Conference, New York (October 2003), <http://mas.cs.umass.edu/paper/265>
4. Dai, Y., Xie, M., Poh, K., Ng, S.: A model for correlated failures in n-version programming. *IIE Transactions* 36(12), 1183–1192 (2004)
5. Grosz, B.J.: Aaai-94 presidential address: Collaborative systems. *AI Magazine* 17(2), 67–85 (1996)
6. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, pp. 403–412. ACM, New York (2004), <http://doi.acm.org/10.1145/988672.988727>
7. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 13(2), 119–154 (2006)
8. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. *Decision Support Systems* 43, 618–644 (2007), <http://portal.acm.org/citation.cfm?id=1225318.1225716>
9. Latif-Shabgahi, G., Bass, J.M., Bennett, S.: A taxonomy for software voting algorithms used in safety-critical systems. *IEEE Transactions on Reliability* 53(3), 319–328 (2004)
10. Littlewood, B., Miller, D.R.: A conceptual model of the effect of diverse methodologies on coincident failures in multi-version software. In: *Fehlertolerierende Rechensysteme/Fault-Tolerant Computing Systems. Internationale GI/ITG/GMA-Fachtagung*, vol. 3, pp. 263–272. Springer, London (1987)
11. Looker, N., Munro, M., Xu, J.: Increasing web service dependability through consensus voting. In: COMPSAC 2005: Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC 2005), vol. 2, pp. 66–69. IEEE Computer Society, Washington, DC, USA (2005)
12. Mitchell, H.: *Multi-Sensor Data Fusion*. Springer, Heidelberg (2007)
13. Parker, J.R.: Multiple sensors, voting methods, and target value analysis. In: Kadar, I. (ed.) Proc. SPIE, Signal Processing, Sensor Fusion, and Target Recognition VIII, Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, vol. 3720, pp. 330–335 (July 1999)
14. Parker, J.: Voting methods for multiple autonomous agents. In: Proceedings of the Third Australian and New Zealand Conference on Intelligent Information Systems, ANZIIS 1995, pp. 128–133. IEEE Computer Society, Perth (1995), <citeseeer.ist.psu.edu/339155.html>
15. Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. *IEEE Internet Computing* 12(4), 78–83 (2008)
16. Sichman, J.S., Conte, R.: Multi-agent dependence by dependence graphs. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, AAMAS 2002, pp. 483–490. ACM, New York (2002), <http://doi.acm.org/10.1145/544741.544855>
17. Sichman, J.S.: Depint: Dependence-based coalition formation in an open multi-agent scenario. *J. Artificial Societies and Social Simulation* 1(2) (1998)
18. Smiley, M.: Carry small, live large: Situational awareness. White paper, Intel Corporation: Intel Developer Forum (August 2008), http://electronics.wesrch.com/Paper/paper_details.php?id=EL11TZQLMTGDE&%paper_type=pdf
19. Townend, P., Groth, P., Xu, J.: A provenance-aware weighted fault tolerance scheme for service-based applications. In: ISORC 2005: Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, pp. 258–266. IEEE Computer Society, Washington, DC, USA (2005)

20. Vouk, M., McAllister, D., Eckhardt, D., Kim, K.: An empirical evaluation of consensus voting and consensus recovery block reliability in the presence of failure correlation. *Journal of Computer and Software Engineering* 4, 367–388 (1993)
21. Zavala, L., Huhns, M.N.: Analysis of coincident failing ensembles in multi-version systems. In: *Proceedings of the 19th IEEE International Symposium on Software Reliability Engineering: Dependable Software Engineering Workshop*. IEEE Press, Los Alamitos (2008)
22. Zavala Gutierrez, R.L., Huhns, M.N.: On building robust web service-based applications. In: Cavedon, L., Maamar, Z., Martin, D., Benatallah, B. (eds.) *Extending Web Services Technologies: The Use of Multi-Agent Approaches*, ch. 14, pp. 293–310. Kluwer Academic Publishing, New York (2004)

Strategy Patterns Prediction Model (SPPM)

Aram B. González and Jorge A. Ramírez Uresti

Departamento de Tecnologías de Información y Computación, Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Estado de México
Estado de México, México
{a00965707, juresti}@itesm.mx

Abstract. Multi-agent systems are broadly known for being able to simulate real-life situations which require the interaction and cooperation of individuals. Opponent modeling can be used along with multi-agent systems to model complex situations such as competitions like soccer games. In this paper, a model for predicting opponent moves is presented. The model is based around an offline step (learning phase) and an online one (execution phase). The offline step is the one that gets and analyses previous experiences while the online step is the one that uses the data generated by offline analysis to predict opponent moves. This model is illustrated by an experiment with the RoboCup 2D Soccer Simulator.

Keywords: Case Based Reasoning, Multi-agent Systems, Opponent Modeling.

1 Introduction

Agents can be defined according to Wooldridge [11] as an autonomous entity in an environment with the capacity of taking its own actions in order to achieve a goal. Also, multi-agent systems take these agents in order to cooperate and achieve a common goal that cannot be completed without the help of other agents.

Multi-agent systems are broadly known for being able to simulate real-life situations which require the interaction and cooperation of individuals. These systems are really good in modeling situations where different autonomous individuals need to interact with each other and their environment in order to accomplish a certain goal.

Due to the multi-agent systems' nature, a common application is to use them to represent a competitive environment in which two teams play against each other in order to accomplish a goal that directly interferes with the other team's objective. An example of this type of environments is the soccer game. A soccer game features two teams composed of eleven players each where the fundamental objective is to score more goals than the opponent. Using agents to represent each player is a natural way to model these kinds of environments since most players tend to have similar capacities and in this case, only the goalkeeper has to attend different rules because it is the only one who can grab the ball with its hands.

In competitive multi-agent systems as in human competitions like a soccer game and in general any kind of game between two or more entities, knowing how our op-

ponent is going to behave and what action is going to make based on the actual world status can be really important in accomplishing our agents' goals. Knowing opponent moves is really desirable because it is easier to anticipate any of their moves and therefore it's also easier to adapt and counter their actions.

In general, the result of predicting the behavior and movements of other agents and storing them in such a way that it is useful for making predictions is known as Opponent Modeling. Opponent modeling does not specify a single technique to achieve its only goal, so the algorithms and methods used for doing so are open for the researchers to choose. It can be implemented in most games that involve two or more players competing against each other. Some research [3, 8] take into account this type of modeling getting some relevant improvements in their results, but in order to create a good model and knowing that it takes a lot of information to do so, both decided to take a sub-domain of their original environment so that the number of possible moves that the opponent can take gets seriously reduced.

For opponent modeling being useful it must provide information so our agents can anticipate most, if not all possible foes', movements in an acceptable manner, even when facing opponents that have never been faced or studied before. To achieve this, the knowledge base containing data corresponding to rivals must be in some way vast and adaptable enough to recognize situations similar to the ones kept in it.

Knowing this, there have been some experiments where opponent modeling is taken to some extreme cases when the information about the rival isn't enough [7]. Parker et al [7] decided to test two cases, in the first one they act as if the opponent is going to always take the best course of action and react to that and on the other hand, in the second case, they take the total opposite and they react without taking opponent in consideration at all. Being the case that the tests were done under an incomplete information game, reacting as if the opponent knows and takes the best choice did not get good results because it assumes something wrong, because of the nature of the environment of the test (Kriegspiel game), the opponent does not really know what the best option is so he must make a decision on what he perceives. As unpredictably as it may look, taking no consideration of the opponent (making a random choice) ended in better results because it had a bigger number of chances of matching the rival action.

Creating a good opponent model is not a trivial task and doing all the process can take a large amount of time but most important it can take a lot of data to represent as many cases as possible. This causes that creating a functional opponent model that is based specifically on the actual rival without any previous knowledge becomes a difficult task and in a really dynamic environment, such as soccer, it becomes almost impossible because of the little number of interactions that can be generalized into a real model of the entire team including its strategies [10].

Taking this into consideration, most opponent modeling applications take a singular approach only modeling the agents and not taking into consideration the complete strategy that they use. So there is a way to know what an agent is going to do as an autonomous entity but, how it is going to interact with its own team is completely ignored.

Opponent modeling is a really good way to interact with agents that have a goal that interferes (many times it is exactly the opposite) with our goal. It lets you anticipate your rival choices and try to act based on them, but there are certain times when it is impossible to create a trustful model or to put it simple, our actual opponent does not fit in our model so you need to have something like a contingency plan [6]. This contingency plan according to McCracken and Bowling [6] can be a really generic opponent model that takes the most basic info (or in some cases nothing at all) to react. This is like a backup plan that gets triggered when everything else fails and does not really interfere with the rest of the opponent model.

Opponent modeling is not exclusive for multi-agent systems, intelligent systems or even computer science; in fact it is a way humans tend to act when they are participating in any kind of competition. Taking soccer as an example, the coach studies his rival team before a match. He performs this analysis based on videos and previous games against that specific team. Some coaches base their whole strategy on the information they have and the previous results a team has gotten using that strategy against that adversary. As we can see, the human need to predict the foes way to act, is the origin of opponent modeling.

Having in mind the origins of opponent modeling it was decided to test our Strategy Patterns Prediction Model (SPPM) in a soccer-like environment. Therefore, the RoboCup 2D simulator was chosen as the concrete application to test it.

In soccer all the strategies are based in the ball position and possession since it is the most determinant factor on the field. Being able to determine a ball position over time in such way that it creates a possible route allows a team to react in such way that it can stop the opponent from completing their goal. A set of strategy patterns can be formed just by following the balls position because it dictates where the opponent players need to be located to realize their plays.

Using opponent modeling in this kind of environment can give the team the following advantages:

- The player can know the style of play of the adversary. This lets it know the adversary preferred way to act and the zones it tends to follow during playtime.
- If something like a confidence model were to be implemented, knowing how the rival acts would give it a good boost. This is something really common in human play.
- Players can take advantage of the opponent's weaknesses so they can elaborate some plays that really hurt the opponent teams' strategy.
- If the other team does not have an evident weakness it can be taken into account in the model and try to act as in a normal situation or take into account an emergency case.

Some approaches similar to the one presented in this research have been done like the one presented by Laviers et al. [5] but the environment in where it is tested and the objective is quite different. The environment used in Laviers et al. [5] is football where players and coaches have more time to take into account the opponent's moves due to the rules of that game, also the research focuses in improving only offensive plays.

This research is based on opponent modeling on multi-agent systems on dynamic environments and it is focused on the defensive actions of the team. It is accomplished following a complete cycle that is going to be discussed in the rest of this document. The remainder of this document is organized as follows. Section 2 describes the environment where the tests took place. The gathering of information and the creation of the knowledge base is discussed in Section 3. In Section 4, the online (in-game) features of the model are discussed (including case retrieval, case differentiation and possible actions to take). Section 5 shows the results obtained by this method and finally in Section 6 conclusions and further work is described.

2 RoboCup 2D Soccer Simulator

RoboCup [9] is an international event that reunites academic institutions to expose new researches in Artificial Intelligence, Multi-Agent Systems, Robotics, etc. Most of these researches base their implementations and results in one of the many competitions presented in RoboCup. The competitions include robotic and simulated versions of rescue situations and soccer.

Giving the nature of this research, it was decided to use the 2D Soccer Simulator to implement and test the model. It was decided not to use the robotic version because of the extra complexity that involves trying to move a real robot, the 3D Soccer Simulator was not chosen because it includes an extra degree of freedom (dimension) that had to be taken into account for the model also raising the number of possible actions any player can take.

A game in the simulator is divided by two half times composed of 3000 steps each. The players can perform only one action by each step except for sending messages, in this case they can send as many as they can until the time of the current step ends.

The RoboCup 2D Soccer Simulator takes most of the human soccer rules making it as close to the human game as possible. It also adds noise to some elements of the game like the motion model, the visual model, and the communications model of each player so that it resembles human play. The field is a rectangular one with dimensions of 70m x 110m. The RoboCup 2D Soccer Simulator field meets the requirements of a regular human soccer field.

The agents are modeled with similar capabilities and all can perform the same actions that include: dash, kick, say, turn and turn neck except the goalkeepers that can also catch the ball with their hands. The players have a given amount of stamina that decreases every time that they make an action that is not the say action. Noise is present in the vision of the player so that the data it perceives varies depending on the distance to the object.

The communications model has also some restrictions, while the players can send as many as they can each step, they can only receive one message (sometimes this message can be incomplete) of a random member of the team and one random message from the opposing team. Also the messages cannot be longer than 10 characters.

To test this research the restrictions involving the message length and messages per cycle in the communications model were eliminated in order to simulate a perception model that lets the agents have reliable information to use in the prediction model.

The goal of the tests done in this research was achieved by executing entire matches between different teams. The scores of the matches were in some cases irrelevant to the results of the test but in general they provide a way to explain what happens with the prediction.

3 Knowledge Base Creation

For SPPM, in order to create a useful opponent model it is needed to take into account previous experiences. In this case, records and logs of past games were used. These logs are automatically generated by the RoboCup 2D Soccer Simulator each time a game is executed and it's saved in a RCG file.

The RCG files can be converted to XML files where all the info of the corresponding game is shown including the server parameters, player parameters, game status and player actions.

The prediction model intends to forecast the ball position when it is in the adversary's possession so not all the information contained in the XML is useful for this purpose. In order to reduce the time needed to create the knowledge base, the unnecessary information inside the XML files is completely removed. This leaves only the data corresponding to the players' actions, players' positions, ball position and game status.

The knowledge base takes only into account the game patterns that have a minimum duration of 10 steps (this is because of the time it takes to make a defensive action corresponding to the pattern detected). There are no more restrictions besides the restriction previously mentioned.

When the knowledge base is created, a search tree is also generated in order to compare the actual game info to the patterns stored in the knowledge base. The generated tree is sorted and takes the ball position as its first and most important discriminant factor. The complete pattern is composed of the positions of the ball and the positions of each of the enemy team's players.

To reduce the complexity of the opponent modeling it was decided that the field must be divided into zones. This division allows the system to be tolerant to the noise generated by the environment (RoboCup 2D Soccer Simulator).

While dividing the field into zones has been done before the presented work [1, 2], it was not found a related work on optimizing the field division based on any criteria so we had to create a division that would serve the SPPM's purpose.

The division was made in such way so that the size of each one of the blocks generated is large enough to reduce system complexity and small enough to keep the prediction relevant. The division's size decision was made based on the fact that having a division consisting of small zones would give us too many combinations for search resulting in none advantages of creating a division at all. Creating big division zones ends up giving us a small search space allowing to reduce the time employed

looking inside the search tree for possible solutions but it also affects the prediction's precision and therefore its usefulness.

This resulted in the soccer field being divided in medium-sized zones that allowed us to generate a grid consisting of 60 zones which is enough to keep the prediction relevant and the search tree in a reasonable size. The final division is shown in figure 1. The field division also takes advantage of the flags provided by the RoboCup 2D Soccer Simulator which serve as reference points for the agents inside the game.

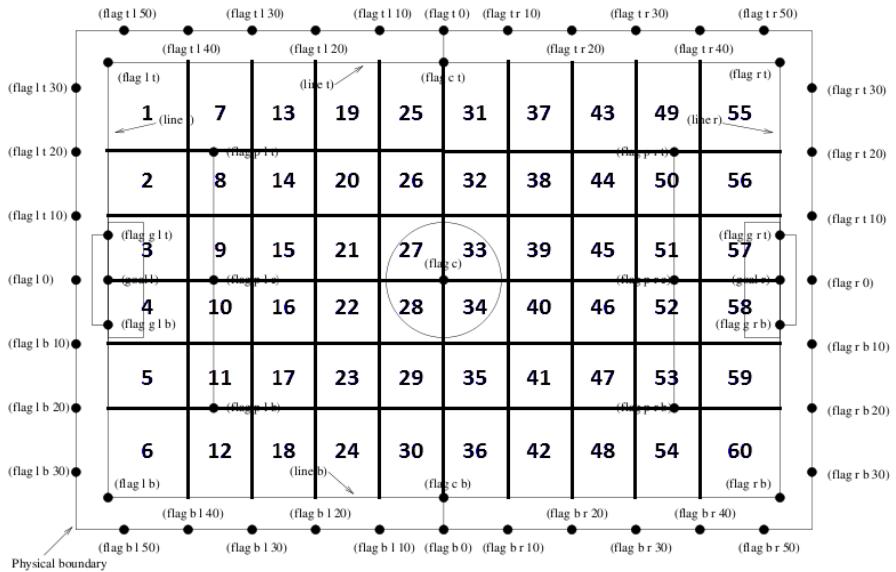


Fig. 1. Field divided into zones to remove complexity and improve noise toleration into the prediction system. The original field image can be found in the RoboCup website [9].

4 In-Game Features

In order to make use of the knowledge base created in the offline mode of the model, SPPM needs to use and test this data into new games. The complete process that must be followed includes: getting the actual game status information (opponent players' and ball positions), look for similar patterns inside the knowledge base and return the possible zones where the ball will be according to the cases stored in the knowledge base with the potential cover zones that will let the team respond to the rival.

To get the complete pattern and, being the fact that this is a multi-agent system, it is needed to obtain the partial information that each of the agents knows, due to this situation some communication restrictions were eliminated as mentioned in the previous section.

Even with the modified communication parameters, trying to make a distributed decision process was not possible since it required a full negotiation cycle involving the agents resulting in time wasted in just trying to coordinate who is going to take the

decisions because of the system allowing the agents to listen teammates messages only a cycle after they have been sent.

The negotiation process was tested and the minimum cycles required to reach a consensus between agents is about 6 cycles. Having the constant time overhead each time a prediction is meant to be done extends the time the entire process requires and since RoboCup 2D Soccer Simulator is an entirely dynamic environment, the time consumed in deciding the team's leader is un-viable because the prediction may no longer coincide with the actual field state.

In order to reduce the time spent on communication issues it was decided to follow a centralized approach for decision process. In human soccer, the goalkeeper normally has a complete vision of the field and also is the player that most likely has the fewest interactions with the ball so we decided based on the characteristics previously mentioned that it should be the agent who receives all the data and to make all the decisions.

Once all the data is received, it needs to be cleaned and consolidated into a single pattern so that it can be used into the search tree. Once the result is taken from the search tree we get all the possible patterns ID taking into account only the best matches. The criteria to decide those matches are based in the distance between the actual field status and the one contained in the pattern, giving more importance to strategies that involve more players. This is called the similarity measure.

This process can be compared and is based on case based reasoning (CBR). CBR uses human like thinking in order to react to actual circumstances based on previous experiences. It has been openly used in this domain [1, 2].

Having all the play patterns that coincide with the actual in-game status, the zones where the ball was during those plays can be taken from the knowledge base and for the prediction a sample is taken each 5 steps (again this is determined in this case by the distance a player can travel). With this information the probabilities of the ball being in a certain zone can be deducted and this is shown in figure 2.

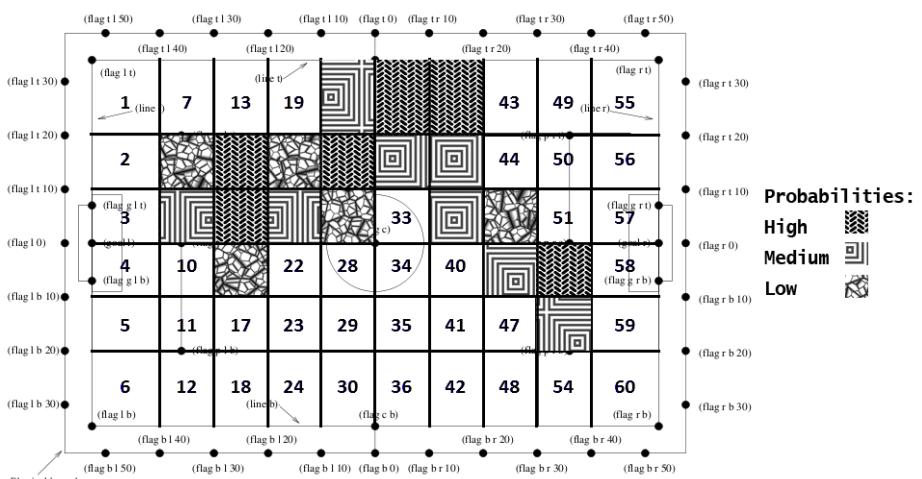


Fig. 2. Zones that indicate the ball's probability of being in each zone in a specific time

Having the zones at each time 5 period interval step, we can determine which ones need to be covered by our players (the ones with most possibilities of the ball being there) and the ones that can be ignored. Being this a soccer game, the areas that need to be covered are the ones that surround the predicted zones and that are between the ball position and the team's goal box.

The next step is to decide whether it is a viable option to send the players to the zones or if it is too late or even if the prediction is wrong and there is no point to cover them. Another issue to take into account is that even if the entire search process takes about 1 or 2 steps to be completed, there are cases when it takes about 12 to 15 steps to end the entire process. In those cases there needs to be a condition in which the process stops if it has taken too much time or if the ball is close to the goal box – these situations affect the goalkeeper's ability to react to an attack, so it may stop performing the decision process and focus in defending its own goal box.

After analyzing all the combinations of possible situations mentioned in the previous paragraph that can be present during a game, it was decided that the following are the only ones that can really affect the team and in particular the goalkeeper's individual goals: catch the ball, react to opponents approaching the goal line, clear the ball from the penalty area, etc.

If the goalkeeper determines that it is viable to make a defensive action, it communicates it to the rest of the team with the time and zones that need to be covered so that the players that are closest to those zones go to them and try to recover the ball.

If any of the agents perceive that the ball or the play is not going where the prediction said it would, then an alert message is sent to prevent covering zones that have almost no probabilities to have the ball inside.

The logs generated from the test files can be analyzed and included inside the knowledge base, this lets the system evolve and get more information about different and new situations. This feature allows our SPPM to respond better in next matches and, in some way, evolve across time.

5 Results Obtained

The tests made during this research involve three types: test with our team against teams that were previously analyzed and included inside the knowledge base, tests against teams not included in the knowledge base and games where our team was not involved at all.

The first test was intended to prove the system reliability against opponents' movement of already known teams. The second one does the same but with other teams and situations that are unknown for the system. The third set is made to test it in general with different teams and not using the one developed for this research.

Twenty-seven analyses were done to get the results that will be presented here where 12 analyses were done over previously analyzed teams, 3 over not previously analyzed teams and 12 over games that did not involve the team developed for this research.

Overall our SPPM achieved to get a prediction of the ball position with a precision over the 80% in an acceptable range defined by being in a distance equivalent to one zone far from the real ball position. This lets the team define coverage zones along time so that the adversary team can be stopped and the ball recovered. An example of the pattern created with the zones is shown in figure 3.

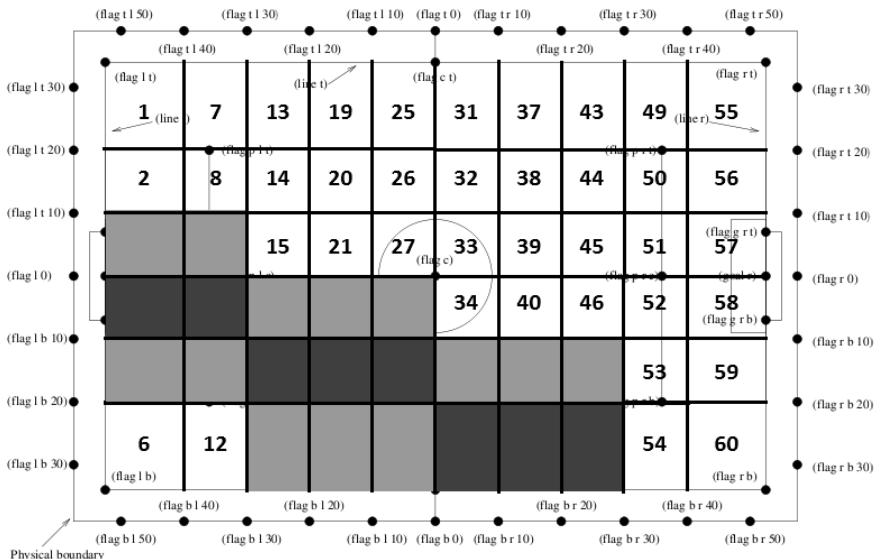


Fig. 3. Pattern created by coverage zones returned by the prediction model

The results involving the distance is shown in figure 4. According to the results, taking into consideration the mean distances in X and in Y separately is the best way to get the actual position of the ball.

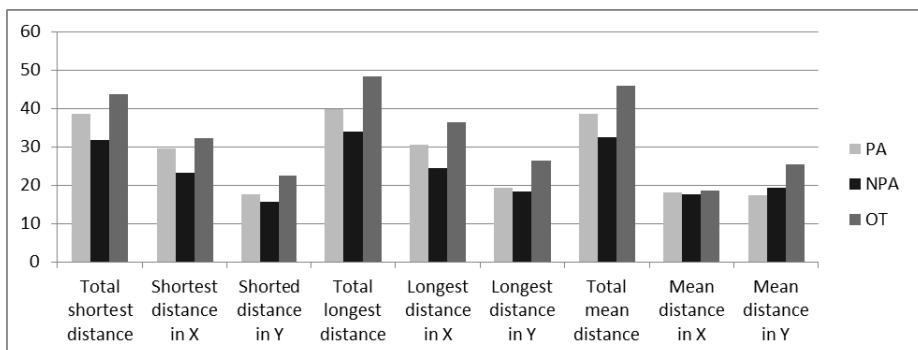


Fig. 4. Distances got in the results for Previously Analyzed teams (PA), Not Previously Analyzed teams (NPA) and Other Teams (OT) that not involve the one developed for this research

Given the results generated by analyzing the distances, the predictions obtained during the course of this research were accommodated in their respective group based on the mean distance during the whole play between the prognosticated zones and the real ones.

The figure 5 shows the percentage of the results and the group that they belong to, this graphic only takes into account the results in mean distance in X and Y because of the results previously generated. It is shown that most of the time (more than 80%); the zones predicted are close enough to the real ones to make a defensive action that lets the team try to recover the ball.

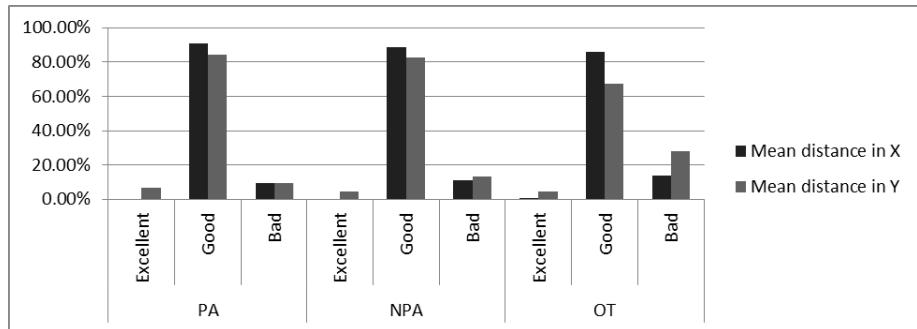


Fig. 5. Percentage of usefulness got in the results for Previously Analyzed teams (PA), Not Previously Analyzed teams (NPA) and Other Teams (OT) that not involve the one developed for this research

Analyzing the results and the circumstances presented inside the games that were shown in the figures 4 and 5 the following circumstances were observed:

- There are some teams less susceptible to fall into the model predictions.
- Some predictions do not end in the zone that was supposed to be in because in some cases the opponent changed the route when a member of our team got close. In other cases the opponent directly shot to goal during the circumstance described and sometimes our team managed to get the ball effectively ruining the prediction.
- Some teams change their tactics depending on the score and the time left to end the game.

Even though the considerations described before, it is considered that the predictions have a very good accuracy rate and are done in an efficient time so that the agents can react to it in real time.

6 Conclusions and Further Work

In this work a model for opponent strategies modeling (SPPM) is presented. SPPM obtains information from RoboCup 2D Soccer Simulation log files, converts this in-

formation into a knowledge base containing strategy patterns used by opponents. With these strategy patterns a search tree is generated to index all possible cases in such a way that any search of the tree is fast enough to be useful in real-time. When in play, the opponent formation of players and ball position is detected, analyzed for possible matches in our knowledge base, a probability of the ball position in the future states is generated, and a final decision on how to defend against this possible strategy is made. All this process is done fast enough so a useful real-time decision is made.

The importance of the creation of the knowledge base that supports the model is also discussed as well as the actions that take place inside the environment that the model is used. The model was tested in the RoboCup 2D Soccer Simulator so that it is proved in a dynamical environment which also has an opponent who takes its own decisions and follows its own course of action. The model discussed in this work gives an accuracy of around 80% in the tests. Taking into consideration the dynamic nature of the environment in which it takes place, it can be said it is a really good percentage.

The model here presented can be used in other domains that do not require having a competition or anything like that, just something to predict and a clear pattern to follow to do it. It also needs to have a way to compare the cases inside the knowledge base and the ones presented during its execution.

Improvements and future work includes:

- Incorporation of weights in the learning phase so we can give priority to certain rivals (possibly the opponent we are about to play against).
- Improving the field zone division criteria so that it allows getting the optimal zone size.
- Online learning phase to complement the offline one so that the team can adapt to what the opponent strategies more precisely.
- Use a different machine learning method instead of case-based reasoning so that if the knowledge base greatly increases its size it does not take a considerable amount of time to do the search process.
- Extend the SPPM to environments that do not involve a competitive scheme but a cooperative one.

Acknowledgments. This work was done under the support of CONACYT. The RoboCup 2D Soccer Simulator base team used for some of this work tests was developed with the help of Gianfranco Arroyo Orrico, José Alberto Ugalde del Rosal and José Iván Ferrer Ruiz.

References

1. Arias, M.: Definición de una Arquitectura para la Generación y Selección de Estrategias en Equipos de Fútbol Robótico. Tesis para la Maestría en Ciencias Computacionales, Instituto Técnológico y de Estudios Superiores de Monterrey, México (2008)
2. Berger, R., Herfert, D.: AT Humboldt Team Description 2009. In: RoboCup International Symposium 2009. RoboCup, Craz Austria (2009)

3. Del Giudice, A., Gmytrasiewicz, P.: Towards Strategic Kriegspiel Play with Opponent Modeling. In: AAAI Spring Symposia. AAAI, Stanford (2007)
4. Kuhlmann, G., Knox, W., Stone, P.: Know Thine Enemy: A Champion RoboCup Coach Agent. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence, pp. 1463–1468. AAAI, Massachusetts (2006)
5. Laviers, K., Sukthankar, G., Molineaux, M., Aha, D.W.: Improving Offensive Performance Through Opponent Modeling. In: Proceedings of the Fifth Artificial Intelligence for Interactive Digital Entertainment Conference, pp. 58–63. AAAI, Stanford (2009)
6. McCracken, P., Bowling, M.: Safe Strategies for Agent Modelling in Games. In: AAAI 2004 Fall Symposia. AAAI, Virginia (2004)
7. Parker, A., Nau, D., Subrahmanian, V.: Overconfidence or Paranoia? Search in Imperfect-Information Games. In: The Twenty-First National Conference on Artificial Intelligence, pp. 1045–1050. AAAI, Boston (2006)
8. Richards, M., Amir, E.: Opponent Modeling in Scrabble. In: Twentieth International Joint Conference on Artificial Intelligence, pp. 1482–1487. IJCAI, Hyderabad (2007)
9. RoboCup (n.d.). Retrieved from RoboCup, <http://www.robocup.org/>
10. Stone, P., Riley, P., Veloso, M.: Defining and Using Ideal Teammate and Opponent Agent Models. In: Proceedings of the Twelfth Annual Conference on Innovative Applications of Artificial Intelligence. AAAI, Austin (2000)
11. Wooldridge, M.: An Introduction to MultiAgent Systems. Bell & Bain, Glasglow (2009)

Fuzzy Case-Based Reasoning for Managing Strategic and Tactical Reasoning in StarCraft

Pedro Cadena and Leonardo Garrido

Tecnológico de Monterrey, Campus Monterrey

Av. Eugenio Garza Sada 2501

Monterrey, Nuevo León. México

pedro.kdna@gmail.com, leonardo.garrido@items.mx

Abstract. We present the combination of Fuzzy sets and Case-Based Reasoning (FCBR) to deal with strategic and tactical management in the real-time strategy environment of StarCraft. Case-based reasoning is a problem solving AI approach that uses past experience to deal with actual problems. Fuzzy set theory is used in case representation to provide a characterization of imprecise and uncertain information. The results revealed that our system can successfully reason about strategies and tactics, defeating the built-in AI of StarCraft. The principal conclusion was that FCBR can reason with abstract information and a large space of actions. Moreover, the resulting system shows its potential to incorporate human knowledge and can effectively adapt to varying conditions of the map.

Keywords: Case-based reasoning, fuzzy sets, tactical management, real-time strategy.

1 Introduction

Nowadays the term Artificial Intelligence (AI) has become popular. The main purpose of artificial intelligence is the creation of intelligent machines and intelligent agents [13]. An important field that contributes to the development of AI is the video game context. One genre of video games that has increasing importance in the AI field is the real-time strategy (RTS) type. Real-time strategy games is a genre of games that do not progress incrementally in turns, the evolution of the game is in continuous time. RTS games offer a large variety of fundamental AI research problems [3]. Decision making under uncertainty, spatial and temporal reasoning, opponent modeling and collaboration are some problems proposed by RTS games to the AI community.

Real-time games provide a challenging domain for game playing agents [8]. The complexity in RTS games increases with the large space of objects and actions presented in the environment. Moreover, the actions in an RTS game can occur at different scale levels. Strategic actions make decisions about what to construct, tactical actions involves decision to deploy group of units across the map and micro-management about individual unit actions. Additionally, players

must deal with multiple, simultaneous, real-time tasks. During game, players may be collecting resources at the same time that decide what to construct while managing several battles. Finally, incomplete information is enforced by RTS games in form of *the fog of war*. It refers to the lack of vision and information on areas of the map that have not been explored by friendly units, or areas that were explored but have since been abandoned.

Different research works have been proposed to develop learning agents that play in RTS games at different levels. Some of these approaches include Case-Based Reasoning (CBR) [1]. CBR is an approach to problem solving and learning that is able to utilize specific knowledge, of previously experienced problem situations. It is inspired by the human capability to solve problems by generalizing over previous observations in a restricted problem domain.

In this paper, we introduce the combination of Fuzzy sets and Case-Based Reasoning (FCBR) for managing tactical reasoning in the RTS game StarCraft. Additionally, we implement CBR for managing strategic reasoning. To deal with the vast space of actions that a match of StarCraft proposes, we divide the task of play a complete match in four categories: resource management, strategic management, tactical management and micro management. Strategic management is handled by an agent that uses CBR for strategy selection. Tactical management is performed by an agent that combines fuzzy sets and case-based reasoning for tactics selection. We use the BWAPI Standard Add-on Library (BWSAL) to develop resource management [5]. Finally, we implemented a hard-coding agent to improve micro organization. The resulting system shows its potential defeating the built-in AI of StarCraft with a rate of 60%.

2 Related Work

Several approaches have been proposed to deal with the large number of objects and actions involved in RTS games in order to make better and novel opponents. In the field of CBR, many researches have published different approaches using CBR to deal with the complexity of RTS games.

Case-based goal formulation is a technique for formulating new goals for an agent using a library of examples [15]. This technique is implemented in a StarCraft playing agent called *EISBot*. The agent consists of two components: a goal formulation component that performs strategy selection using a CBR approach and a reactive planner that handles second to second actions in the game. The proposed presented in this paper uses a similar CBR component to manage strategy during gameplay. The difference is that our system is not based on goals. Instead, our case library stores cases composed of features that represent the state of the game and the actions executed in that state. Furthermore, our system incorporates a fuzzy CBR component to deal with tactical actions in the game.

Case-based planning and execution approach [10] involves learning behaviors from expert demonstrations, and uses the learned behaviors inside a case-based planning system to reuse them for new situations. The process is divided in

two main stages: behavior acquisition and execution. CBR is used to maintain a current partial plan tree. The main difference with our approach is that they use CBR to construct a plan tree based on goals and behaviors while we use CBR to execute strategic and tactical reasoning without building a general plan. Moreover, this system needs to annotate the traces of the game with the goals the player was pursuing while our system automatically extracts cases from *replays*.

Case-based Tactician (CAT) is a case-based system that learns to select proper tactics to use at each state during gameplay with random opponents in the real-time strategy game of Wargus [2]. CAT employs three sources of domain knowledge: a state lattice, a set of tactics for each state and the use of cases for mapping game situations to tactics and their performance. The differences with our approach are that we combine fuzzy sets and case-based reasoning to deal with tactical management and that we also add CBR for strategic management.

Other approaches applied to RTS games that do not rely on CBR include the use of algorithms based on human perceptions and finite state machines [16], dynamic scripting [12], neuro-evolution methodologies [9], among others.

3 Strategic Reasoning

Strategy involves high-level decisions about which type of buildings and units to produce. Several factors affect the strategy during game. It is common to select a general strategy at the beginning of the game and modify this plan according with the circumstances, such as the discoverer of new enemy technology.

To deal with strategic reasoning we propose a Case-Based Reasoning (CBR) agent that handles the information and decisions about what kind of structures and units to produce depending on the state of the game.

3.1 Strategy Selection

With the help of StarCraft players, we selected the strategy for our agent. When playing against built-in AI a good strategy is to lead massive attacks against the enemy. Our strategy relies on the production of two types of combat units: *Zealots* and *Dragoons*.

Our strategy is to train groups of *Zealots* and *Dragoons* and then to send groups to attack enemy regions. Units are produced continuously so we can attack with a good frequency. The strategy also involves training *Observers* that are used for deep-space exploration and are also used as detectors. Observers support our combat units to detect hidden enemy units and to explore enemy regions.

3.2 Case Representation for Strategies

Cases are composed of a description of the problem and its solution. The description of the problem is the representation of the state of the game. The solution

part is the building and training actions executed in such state of the game. Hence, we can see our cases composed of two sections, the state of the game and the strategic orders to execute. In Figure 1, we can see the representation of a typical case that composes our case base for strategic reasoning.

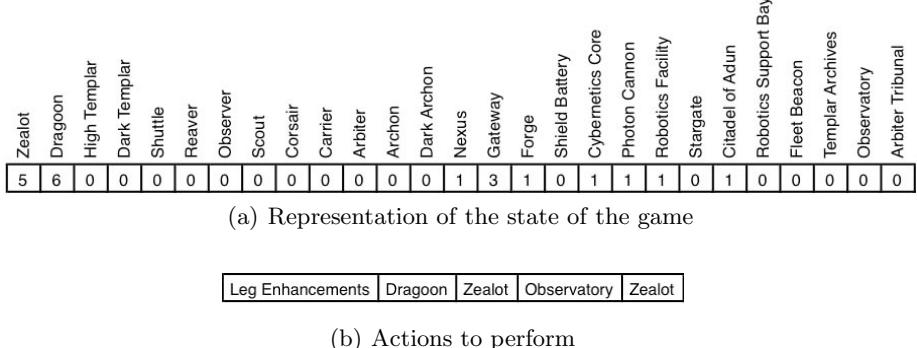


Fig. 1. Case representation for strategic reasoning

We represent the state of the game as a vector of 27 elements. Each element of the vector represents a feature of the game. Our features are each unit type of the game, including buildings and combat units. The first 13 features denote combat unit types; the other 14 features denote building unit types. The value of each element represents the number of units, of that type, produced since the start of the game. So, for example in figure 1(a), we can see in the first cell of the vector, that five units of type *Zealot* have been produced since the beginning of the game.

The solution part of the case consists of the list of actions to be performed by the agent. Actions represent the type of unit that must be constructed or trained. Actions also involve the kind of upgrades and tech development to complete. In figure 1(b) we can see that the actions of that case correspond to upgrade the *Leg Enhancement*, to train a *Dragoon* and a *Zealot*, to construct an *Observatory* and to train a *Zealot* again. We empirically defined a length of five actions for every case. This means that our planning window has a length of five actions. Our agent interacts with BWSAL library to perform the CBR cycle and execute the actions.

Figure 2 illustrates the case representation for strategies in StarCraft. The top left side of the figure shows a partial state of the game as a vector of numerical values that represent the number of units constructed since the beginning of the game. In this example, there exist four units of type *Zealot* and five units of type *Dragoon*. In this figure, the actions that correspond to the solution of the case were to build a *Gateway*, *Citadel of Adum*, *Dragoon*, *Zealot* and *Robotics Facility*.



Fig. 2. Case representation for strategies in the game

3.3 Methodology to Build the Case Base

We build our case with the information of one *replay* where a human player battles against the built-in AI of StarCraft, following the strategy of massive attacks described before. The case base is saved in text file format where each line represents a case. This file is added to the StarCraft folder and is loaded by our agent using a coded method. Following we describe the methodology to build the case base:

1. An experienced human player plays a match against the built-in AI of the game.
2. The game is saved in a format that enables the *replay* of the game.
3. The replay is analyzed using the Brood War Application Programming Interface (BWAPI) [4].
4. The features that represent the state of the game and the strategic actions are captured.
5. The information is saved in form of cases and is added to the case base.

3.4 Matching Treat

Our strategic agent creates a new case with the actual state of the game. Then, the agent compares the new case with every case inside the case base and selects the case that is the most similar to the new case. To compare the similarity between the new case and the cases of the case base, we use the Euclidean Distance. We have two states: one represents the state of the new case and the other represents the state of the case extracted from the case base. Each state is described as a vector of integers that represent the number of different type of units that have been produced since the beginning of the match. If we define the two states as $\mathbf{p} = (p_1, p_2, \dots, p_n)$ and $\mathbf{q} = (q_1, q_2, \dots, q_n)$ then the Euclidean Distance from \mathbf{p} to \mathbf{q} or form \mathbf{q} to \mathbf{p} is given by equation 1.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (1)$$

This distance denotes the similarity between cases. Cases with shorter Euclidean Distances are more similar than cases with larger Euclidean Distances.

4 Tactical Reasoning

We refer to tactics as intermediate decisions about how to deploy groups of units across the map and execute attack orders. The movement of military units takes importance in game to discover new regions, to cover terrain and to send units to attack. Tactics leads the armies and can vary along the game according with the objectives.

To deal with tactical management we propose the combination of fuzzy sets and case-based reasoning into an agent that handles the information and decisions to deploy units across the map. In this paper, the combination of fuzzy sets and case-based reasoning is called Fuzzy Case-Based Reasoning (FCBR). The union of fuzzy theory and case-based reasoning simplifies the process of knowledge representation and enables the knowledge acquisition practice using a case base [14]. The use of linguistic variables to represent the features of the cases reduces the large space of actions and objects that an RTS game proposes. Therefore, using a FCBR approach, we can deal with the vast space of actions presented in an RTS game and incorporate human knowledge in the reasoning process. Our tactical agent is based on CBR approach and uses a fuzzy representation of cases. This technique allows dealing with abstract and incomplete information. Furthermore, an abstract representation of the state of the game is more like human thinking. Human players tend to think in a form that is fluid or approximate rather than fixed and exact, fuzzy theory deals with such kind of reasoning.

4.1 Abstraction of the Space

To accomplish tactical reasoning and to deal with the large amount of actions and information that an RTS game like StarCraft proposes, we divide the state of the game into regions. Using the BWTA library, an analysis of the map can be done. This analysis computes the regions, chokepoints and base locations [6].

Our tactical agent uses regions to perform tactical reasoning. Instead of having a game state that includes information about the entire map, we have a game state for every region of the map. Therefore, each region has its own fuzzy state description. Individual regions are independent of other regions. This means that we have a state description of the game for each single region of the map. Hence, a typical case represents the state of some region and the actions executed in such state for such region.

4.2 Fuzzy Case Representation

Analogously to the strategic representation of cases, tactical cases are composed of a description of the problem and its solution. The description of the problem is the representation of the state of some region. The solution part involves the tactical actions executed over the units of such region to move them across the map.

In Figure 3, we can see a partial representation of a typical case that composes our case base for tactical reasoning. We represent the state of a region as a vector of 31 elements. Each element of the vector represents a feature of the region. Each feature has a range of values characterized by linguistic variables. For example, in figure 3(a) the first element of the vector represents the feature of *area*, the *area* is characterized by a set of linguistic variables (small, medium, big, etc.), that represent the geographical size of the region.

Area	Chokepoints	Military Presence	Combat Intensity	Lost Units	Corsair	Dark Templar	Dark Archon	Zealot	Dragoon	High templar	Archon	Shuttle	Scout	Arbiter	Carrier	Reaver	Observer
big	one	friendly	none	none	none	none	none	few	few	none	none	none	none	none	none	none	none

(a) Fuzzy representation of the state of a region

AttackUnit	Protoss Zealot	Protoss Dark Templar	AttackUnit
------------	----------------	----------------------	------------

(b) Actions to perform

Fig. 3. Case representation for tactical reasoning

The first two elements of the vector denote the geographical features of *area* and *chokepoints*. The next three elements represent the military features of *military presence*, *combat intensity* and *lost units*. The next 13 features represent the number of friend combat units in the region and the last 13 elements of the vector (not shown in the figure) denote the number of enemy combat units in the region. The *numeric state* feature is used in the last 26 elements of the vector to characterize the number of units in the region for every type of combat unit.

The solution part of the case is composed of the list of tactical actions to be performed by the agent. Tactical actions are issued to friend combat units in the region. Such actions involve move or attack-move other regions, attack enemy units in the same region or attack enemy units that are in other regions. Actions are composed of the order name, the type of unit that executes the action, the type of target unit and the description of the target region. For example, in figure 3(b) the case contains the action of *AttackUnit*, the type of unit that must execute that action is *Zealot* and the target unit type is *Dark Templar*.

Figure 4 illustrates the case representation for tactics in StarCraft. The top left side of the figure shows a partial state of the game as a vector of linguistic variables that represent the fuzzy features of the state of the region. In this example, the region is big and it has one path that communicates with other regions. Moreover, there are many units of type *Zealot*.



Fig. 4. Case representation for tactics in the game

4.3 Fuzzy Sets and Case Base

A fuzzy set is a collection of elements, which possess a specific property of the set and which were qualified in the set using a qualification algorithm [11]. We used trapezoidal membership functions to represent the fuzzy sets that characterize the state of the regions in our cases.

Figure 5 shows an example of the feature of area used in our case representation. For the feature of area we have small, medium, big, very big and immense membership functions. The horizontal axis represents the area of the region in pixels and the vertical axis represents the membership degree. All linguistic variables are defined in a similar way.

Analogous to strategic reasoning, our tactical agent incorporates human knowledge for tactical reasoning in its case base. The information to build the tactical case base was acquired following the same methodology as for strategies. Our case base for tactical management is composed of 147 cases.

We have two case bases, one for strategic management and one for tactical management.

4.4 Fuzzy Matching Method

Our tactical agent creates a new case with the actual state of the region. Then, the agent compares the new case with every case inside the case base (tactical case base) and selects the case that is the most similar to the new case. Our case base has cases of all regions.

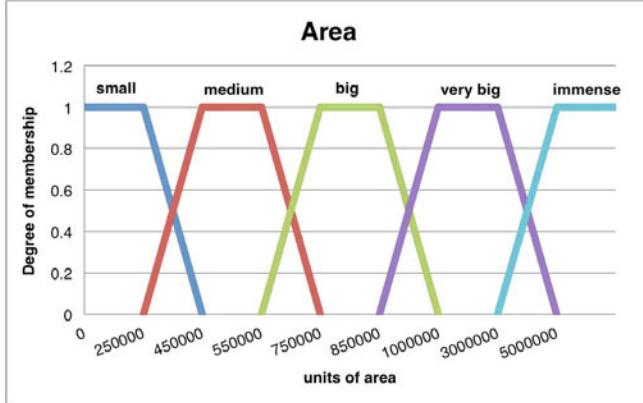


Fig. 5. Membership functions for *area*

To compare the similarity between the new case and the cases of the case base we use the intersection area between membership functions [7]. Assuming that A_1 is the area associated with one membership function, A_2 is the area associated with the second membership function and OA is the overlapping area, then the similarity between the two attributes is defined as:

$$S_{IA} = \min\left(\frac{OA}{A_1}, \frac{OA}{A_2}\right) \quad (2)$$

Using the intersection area we obtain the overall similarity between two cases using a weighted Euclidean Distance. We have two fuzzy states; one represent the state of the new case and the other represents the state of the case extracted from the case base. Each state is described as a vector of linguistic variables that represent fuzzy sets and membership functions. If we define the two states as $\mathbf{p} = (p_1, p_2, \dots, p_n)$ and $\mathbf{q} = (q_1, q_2, \dots, q_n)$ then the weighted Euclidean Distance from \mathbf{p} to \mathbf{q} or form \mathbf{q} to \mathbf{p} is given by equation 3.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n w_i (1 - S_{IA}(p_i, q_i))^2} \quad (3)$$

Where w_i is a weight attached to the i -th variable and $S_{IA}(p_i, q_i)$ is the intersection area between the membership functions of the i -th components of the states of the cases. This distance denotes the similarity between cases. Cases with shorter weighted Euclidean Distances are more similar than cases with larger weighted Euclidean Distances.

5 Experiments and Results

We implemented our approach in the RTS game of StarCraft. We focused on strategic and tactical management building a case-based reasoning agent to accomplish strategic reasoning and building a fuzzy case-based reasoning agent to

accomplish tactical reasoning. The development of these agents and the addition of other ones resulted in a bot (a software program that imitates the behavior of a human) that plays against built-in AI enemies in StarCraft. The resulting bot is called *FCBR-Bot*.

We developed our experiments in the map of Python. This is a four players map and has four possible start positions for players. We performed our experiments in a *Protoss* vs *Protoss* version of the game where our bot and the built-in AI of the game engaged in battle. Furthermore, all experiments were performed using incomplete information by means of *the fog of war*.

We divided the experiments in three parts according with the start location of our bot and the enemy. In the first experiment our bot and its opponent begin in a selected start position of the map. These positions are selected to resemble the start positions of the players in the *replay* where we extracted the information. In the second experiment our bot begins in a selected start position while its opponent has a random start location. In the third experiment both our bot and the opponent can start in random locations of the map. For each configuration we performed 100 experiments, it means 100 matches, where our bot played against the built-in AI of the game. The results of the games are shown in table 1.

Table 1. Game Stats for *FCBR-Bot* vs Built-in AI

Map Positions	Win Games	Lose Games	Tie Games
Selected vs Selected	60%	28%	12%
Selected vs Random	58%	25%	17%
Random vs Random	61%	23%	16%

Analyzing the results we observe that the implementation of our approach was victorious in approximately 60% of the played games while the opponent only won approximately 25% of the total games, getting ties in the rest of the matches. This means that a case-based reasoning and a fuzzy case-based reasoning approach that enables strategic and tactical reasoning can successfully defeat built-in AI opponents in StarCraft independently of the start locations of both players. Therefore, our system can deal with abstract information and a large space of actions and can effectively adapt to varying conditions of the map presented in terrain differences of every region.

Figure 6 shows a comparison between *EISBot* [15] and *FCBR-Bot* results in *Protoss* versus *Protoss* games. The results obtained by our system in the three configurations mentioned before are called FCBR1 for selected vs selected initial positions, FCBR2 for selected vs random initial positions and FCBR3 for random vs random initial positions.

Playing the *Protoss* faction, *EISBot* was victorious 35% of the times, playing against the *Protoss* built-in AI of StarCraft on the map of Python. Comparing this results with the results obtained by our approach it can be stated that

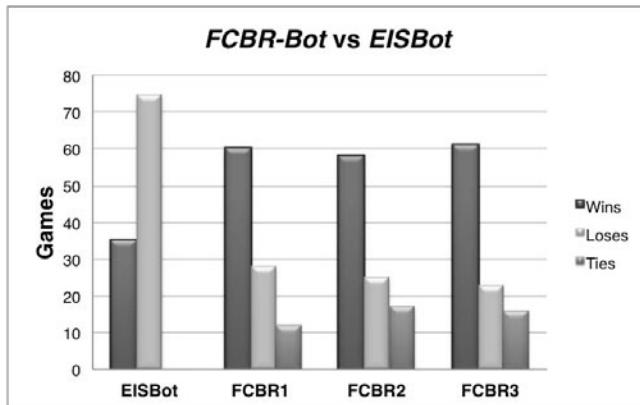


Fig. 6. Comparison between *FCBR-Bot* and *EISBot* in Protoss vs Protoss games

our system performs better than *EISBot* on the same map winning in 60% of the matches in *Protoss* versus *Protoss* games. In comparison with *EISBot*, our system incorporates fuzzy case-based reasoning to deal with tactical management and uses other additions to play complete games of StarCraft.

6 Conclusions and Future Work

In this paper, we have presented the use of fuzzy sets and case-based reasoning approach for dealing with strategic and tactical reasoning in the real-time strategy game of StarCraft. We showed the implementation of case-based reasoning to deal with strategic management. Moreover, we described the implementation of fuzzy case-based reasoning to deal with tactical management. We described the implementation of our approach in StarCraft.

There are several research directions for future work. The first direction is to evaluate the potential of our approach in transfer learning tasks, such as playing all three races in StarCraft. Actually, our system plays *Protoss* versus *Protoss* games. Additionally, as a second research direction, we would like to systematically explore how the knowledge learnt in a set of maps can be applied to a different set of maps. The third direction of research involves to experiment adding a case retention module in our system that retains automatically all the behaviors that had successful results while playing, and also annotating all the cases in the case base with their rate of success and failure allowing the system to learn from experience. The last direction of future work is to merge the improvements mentioned before and implement a system to contend in the AIIDE StarCraft AI Competition.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications* 7(1), 39–59 (1994)
2. Aha, D., Molineaux, M., Ponsen, M.: Learning to Win: Case-Based Plan Selection in a Real-Time Strategy Game. In: Muñoz-Ávila, H., Ricci, F. (eds.) ICCBR 2005. LNCS (LNAI), vol. 3620, pp. 5–20. Springer, Heidelberg (2005)
3. Buro, M., Furtak, T.: RTS games as test-bed for real-time AI research. In: Proceedings of the 7th Joint Conference on Information Science (JCIS 2003), pp. 481–484. Citeseer (2003)
4. BWAPI: Brood war application programming interface (bwapi) (2011), <http://code.google.com/p/bwapi/> (retrieved on April 2011)
5. BWSAL: The bwapi standard add-on library (bwsal) (2011), <http://code.google.com/p/bwsal/> (retrieved on April 2011)
6. BWTA: Broodwar terrain analyzer (bwta) (2011), <http://code.google.com/p/bwta/> (retrieved on April 2011)
7. Dvir, G., Langholz, G., Schneider, M.: Matching attributes in a fuzzy case based reasoning. In: 18th International Conference of the North American Fuzzy Information Processing Society, NAFIPS, pp. 33–36. IEEE (1999)
8. McCoy, J., Mateas, M.: An integrated agent for playing real-time strategy games. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 1313–1318 (2008)
9. Olesen, J., Yannakakis, G., Hallam, J.: Real-time challenge balance in an rts game using rtneat. In: IEEE Symposium On Computational Intelligence and Games, CIG 2008, pp. 87–94. IEEE (2009)
10. Ontañón, S., Mishra, K., Sugandh, N., Ram, A.: Case-Based Planning and Execution for Real-Time Strategy Games. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 164–178. Springer, Heidelberg (2007)
11. Piegat, A.: A new definition of the fuzzy set. *Int. J. Appl. Math. Comput. Sci.* 15(1), 125–140 (2005)
12. Ponsen, M., Spronck, P.: Improving adaptive game ai with evolutionary learning. In: Proceedings of Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004), pp. 389–396. University of Wolverhampton, England (2004)
13. Russell, S., Norvig, P.: Artificial intelligence: a modern approach. Prentice-hall (2010)
14. Sahar, N., Kazuhiko, S., Hirotsugu, M.: Utilizing fuzzy-cbr methodology as an artificial intelligent engine in generating scenario case for plant virtual reality risk simulation. *European Journal of Scientific Research* 41(1), 57–71 (2010)
15. Weber, B., Mateas, M., Jhala, A.: Case-based goal formulation. In: Proceedings of the AAAI Workshop on Goal-Driven Autonomy (2010)
16. Wintermute, S., Xu, J., Laird, J.: Sorts: A human-level approach to real-time strategy ai. *Ann Arbor* 1001, 48109–2121 (2007)

Variable and Value Ordering Decision Matrix Hyper-heuristics: A Local Improvement Approach

José Carlos Ortiz-Bayliss¹, Hugo Terashima-Marín¹, Ender Özcan²,
Andrew J. Parkes², and Santiago Enrique Conant-Pablos¹

¹ Tecnológico de Monterrey, Campus Monterrey
Monterrey, Mexico, 64849

jcobayliss@gmail.com, {terashima,sconant}@itesm.mx

² University of Nottingham, Jubilee Campus
Nottingham, United Kingdom, NG8 1BB
ender.ozcan@nottingham.ac.uk, ajp@cs.nott.ac.uk

Abstract. Constraint Satisfaction Problems (CSP) represent an important topic of study because of their many applications in different areas of artificial intelligence and operational research. When solving a CSP, the order in which the variables are selected to be instantiated and the order of the corresponding values to be tried affect the complexity of the search. Hyper-heuristics are flexible methods that provide generality when solving different problems and, within CSP, they can be used to determine the next variable and value to try. They select from a set of low-level heuristics and decide which one to apply at each decision point according to the problem state. This study explores a hyper-heuristic model for variable and value ordering within CSP based on a decision matrix hyper-heuristic that is constructed by going into a local improvement method that changes small portions of the matrix. The results suggest that the approach is able to combine the strengths of different low-level heuristics to perform well on a wide range of instances and compensate for their weaknesses on specific instances.

Keywords: Constraint Satisfaction, Hyper-heuristics, Variable and Value Ordering.

1 Introduction

A Constraint Satisfaction Problem (CSP) is defined as follows: given a set of n variables with its respective domain of possible values m and a set of constraints, each one involving a subset of variables, find all possible n -tuples such that each n -tuple is an instantiation of the n variables satisfying all the constraints. For this research we have only considered those CSP in which the domains are discrete, finite sets and the constraints involve only one or two variables (binary constraints). Various studies have been developed to randomly generate instances of binary CSP (see for example [15]), and those studies have shown that random

binary CSP have very interesting properties which make them an important topic of study. The relevance of studying CSP lies on the fact that they are an important technique used to find a solution for many artificial intelligence problems [11].

Binary CSP present two important properties that are used in this research: the constraint density (p_1) and the constraint tightness (p_2). The constraint density is a measure of the proportion of constraints within the instance; the closer the value of p_1 to 1, the larger the number of constraints in the instance. The constraint tightness (p_2) represents a proportion of the conflicts within the constraints. A conflict is a pair of values $\langle x, y \rangle$ that is not allowed for two variables at the same time. The higher the number of conflicts, the more unlikely an instance has a solution.

Stated as a classic search problem, a CSP is usually solved using a Depth First Search (DFS) where every variable represents a node in the tree. Every time a variable is instantiated, the constraints in which that variable is involved must be checked to verify that none of them is violated. When an assignation violates one or more constraints, the instantiation must be undone, and another value must be considered for that variable. If there are not any values available, the value of the previous instantiated variable must be changed. This technique is known as backtracking. There are some improvements to this basic search method which try to reduce the number of revisions of the constraints (consistency checks) like constraint propagation and backjumping. With constraint propagation the idea is to propagate the effect of one instantiation to the rest of the variables due to the constraints among them. Thus, every time a variable is instantiated, the values of the other variables that are not allowed because of the current instantiation are removed. Because of this, only allowed values for the variables remain. Backjumping is another powerful technique for retracting and modifying the value of a previously instantiated variable but is different to backtracking because backjumping can go back more than one level at the time when a backward movement is needed.

The selection of the next variable to instantiate and its respective value affects the search complexity and represents an opportunity to optimize the search. Every CSP contains characteristics that could make it more suitable to a certain heuristic. If these characteristics can be identified, then the problems would be solved more efficiently. Some heuristics for variable and value ordering exist, but none of them has been able to behave well for all instances. As we can observe, the selection of the right heuristic for the current instance is not trivial.

Hyper-heuristics are a modern approach to take advantage of the selective application of the low-level heuristics based on the current problem features. Even though the idea of combining multiple heuristics goes back to 1960s [8] the term hyper-heuristic was first introduced by Denzinger et al. [7] in 1997. Surveys on hyper-heuristic methodologies can be found in [5] and [3]. Hyper-heuristics can be divided into two main classes: those which select from existing heuristics and the ones that generate new heuristics. A more detailed description about the classification of hyper-heuristics can be found in [5] and [4]. As

representative studies on the hyper-heuristic methodologies that generate new heuristics, we can cite Fukunaga, who uses genetic programming as an automated heuristic discovery system for the SAT problem [9]. Conversely, most of the hyper-heuristics that select from existing low-level heuristics are based on a similar iterative framework in which the search is performed in two successive stages: heuristic selection and move acceptance. One of the first attempts to systematically map CSP to algorithms and heuristics according to the features of the problems was presented by Tsang and Kwan [18], in 1993. In that study, the authors presented a survey of algorithms and heuristics for solving CSP and they proposed a relation between the formulation of the CSP and the most adequate solving method for that formulation. More recently, Ortiz-Bayliss et al. [13] developed a study about heuristics for variable ordering within CSP and a way to exploit their different behaviours to construct hyper-heuristics by using a static decision matrix to select the heuristic to apply given the current state of the problem. More studies about hyper-heuristics applied to CSP include the work done by Terashima-Marín et al. [17], who proposed an evolutionary framework to generate hyper-heuristics for variable ordering in CSP; and the research developed by Bittle and Fox [1] who presented a hyper-heuristic approach for variable and value ordering for CSP based on a symbolic cognitive architecture augmented with case based reasoning as the machine learning mechanism for their hyper-heuristics. Ortiz-Bayliss et al [14] recently presented a study where they represent variable ordering hyper-heuristics as integer matrices and a genetic algorithm is used to evolve the structure of the matrices in order to generate hyper-heuristics.

Our model produces hyper-heuristics represented as matrices of integers by going through a local improvement process. Each matrix represents a rule for the application of the ordering heuristics based on the values of p_1 and p_2 of the instance at hand.

This paper is organized as follows: Section 2 describes in detail the solution model developed for this research. The experiments and main results are shown in Section 3. Finally, Section 4 presents the conclusions and future work.

2 Solution Approach

In this section we present the solution model proposed in this research. We also include a brief description of the variable and value ordering heuristics and the CSP instances used in this investigation. Finally, the decision matrix hyper-heuristic and the local improvement process are described.

2.1 Variable and Value Ordering

Many researchers have proved the importance of the order of the variables and its impact in the cost of the solution search [15]. The search space grows exponentially with respect to the number of variables, and so does the time for finding the optimal ordering. Once a variable has been selected to be instantiated we need to decide which value, among all the feasible ones, will be used for

that instantiation. This ordering also has relevance to the search, because it also affects the complexity of the search.

Various heuristic and approximate approaches have been proposed that find good solutions for some instances of the problem. However, it has not been possible to find a reliable method to solve well all instances of CSP. In this study, we have included two variable ordering heuristics and two value ordering heuristics. Max-Conflicts (MXC) and Saturation Degree (SD) [2] are the variable ordering heuristics and we have included Min-Conflicts (MINC) [12] and Max-Conflicts (MXC) as value ordering heuristics.

A solution for any given CSP is constructed selecting one variable at a time based on one of the two variable ordering heuristics used in this investigation: MXC and SD. Each one of these heuristics reorders the variables to be instantiated dynamically at each step during the construction process. Later, a value must be selected and assigned to the chosen variable considering the constraints and using MINC or MXC as value ordering heuristic depending on the instance features. A CSP solver that makes use of constraint propagation and backjumping was implemented for developing this research. The ordering heuristics used in this investigation are briefly explained in the following lines.

MXC is a very simple and fast heuristic, and the main idea is to select the variable which values are involved in the larger number of conflicts among the constraints in the instance. MXC can be used both for variable ordering and for value ordering. When used for variable ordering, the instantiation will produce a subproblem that minimises the number of conflicts among the variables left to instantiate. When applied to value ordering, the principle of selecting the ‘most conflictive’ value remains, but now the heuristic only chooses values from the selected variable. The other variables are not considered because they have not been selected for instantiation.

SD has been more frequently used for graph colouring, but it is possible to adapt it for being applied to the variable ordering problem in general CSP. The degree of a node is defined as the number of nodes adjacent to it. The saturation degree of a node X_i is the number of different colours to which X_i is adjacent [2]. In graph colouring, one implicit constraint is that two adjacent nodes cannot be assigned the same colour. Thus, all connected instantiated nodes must have different colors. In our research, we define the saturation degree of a node X_i as the number of adjacent nodes to X_i that have already been instantiated. In this way, our implementation of SD takes advantage of the topology of the constraint graph to select the most restricted variable given the progress in the search.

MINC is one simple and commonly used value ordering heuristic in CSP. When using MINC, the heuristic prefers the value (from the selected variable) that is involved in the minimum number of conflicts [12]. MINC will try to leave the maximum flexibility for subsequent variable assignments.

By combining the variable and value ordering heuristics, we obtain four combinations: MXC/MINC, SD/MINC, MXC/MXC and SD/MXC. From this point on we will refer to each one of these combinations as ordering heuristics. In this

way, each heuristic describes one way to order the variables and, at the same time, a way to order the values from the variables.

2.2 CSP Instances and the Problem State Representation

The binary CSP instances used for the experiments in this research are randomly generated in two stages. In the first stage, a constraint graph G with n nodes is randomly constructed and then, in the second stage, the incompatibility graph C is formed by randomly selecting a set of edges (incompatible pairs of values) for each edge (constraint) in G . More details on the framework for problem instance generation can be found in [16]. The parameter p_1 determines how many constraints exist in a CSP instance and it is called constraint density, whereas p_2 determines how restrictive the constraints are and it is called constraint tightness. In this model, there should be exactly $p_1 n(n - 1)/2$ constraints (rounded to the nearest integer), and for each pair of constrained variables, the number of inconsistent pairs of values should be exactly $m^2 p_2$ (where m is the uniform domain size of the variables). Every time a variable is assigned a new value and the infeasible values are removed from domains of the remaining uninstantiated variables, the values of p_1 and p_2 change and a sub-problem with new features appears. This is the reason why we decided to use the constraint density and tightness to represent the problem state and guide the selection of the low-level heuristics. Our idea is that these two features can be used to describe a CSP instance and to create a relation between instances and heuristics.

There is a relation between the structure of CSP and the difficulty of solving them with search algorithms [6]. Specifically, the median search cost of search algorithms for CSP exhibits a sharp peak as the values of p_1 and p_2 change. This peak coincides with the transition from under-constrained to over-constrained instances (region known as transition phase), often manifested as an abrupt change in the probability that an instance has a solution. Inside this region, the most difficult soluble problems and the most difficult insoluble problems co-exist [16].

The collection of instances used for training includes 130 different hard instances, divided into two sets: 30 for Set A and 100 for Set B. Both sets A and B are composed by distinct instances generated with the parameters ($n = 20, m = 10, p_1 = 0.6, p_2 = 0.33$), which correspond to hard instances [6]. An additional Set C is used only for testing purposes. For Set C we generated 405 instances, with $n = 20, m = 10, p_1 = 0.6$ and p_2 in the range $[0, 1]$ with increments of 0.0125. The complete collection of instances used in this research includes both easy instances (outside the transition phase) and hard instances (inside the transition phase).

2.3 The Decision Matrix Hyper-heuristic and the Local Improvement Approach

The decision matrix hyper-heuristic is formed by a matrix of integers, where the rows are used to represent the constraint density (p_1) and the columns the

constraint tightness (p_2). There are other features that can be used to describe a CSP instance (see for example: k and ρ in [10]) but p_1 and p_2 have been usually applied to for this purpose. The value of each element represents the heuristic to apply when the instance features correspond to the values of p_1 and p_2 coded in the axes. Each cell of the matrix corresponds to one decision point, the points where the hyper-heuristic decides which heuristic to apply to continue the search. Each cell contains a value from 0 to 3, which is used to specify the ordering heuristic to be used when the instance features match the ones at that decision point. Because the matrix presents increments of 0.10 on both p_1 and p_2 axes (and these parameters lie in the range (0, 1]), the resulting matrices contain 10×10 cells. We tried different resolutions and sizes for the matrices and we observed that increasing the resolution of the matrix does not necessary provide a better performance. After preliminary studies we found that the best size of the matrices (in the number of cells per rows and columns) lies in the range [10, 20]. We observed that small decision matrices do not contain enough decision points to produce a significant change of heuristics during the search. Thus, the hyper-heuristics obtained are very monotonous. Conversely, than 20 rows and columns produced over-detailed matrices that required larger amounts of time during the improvement process. The results produced by this over-detailed matrices did not show to be worth the additional improvement time. When the hyper-heuristic is created, all the cells are initialized to -1, meaning that the cell has never been used as decision point. When the hyper-heuristic is used to solve an instance, the decision points used during the search are initialized. There are two ways to initialize the decision points: (1) selecting one random ordering heuristic or (2) using a default ordering heuristic. For this research we decided to use the second way to initialize the matrices.

We will try to clarify how the decision matrix hyper-heuristic is used with an example. Imagine we have a hyper-heuristic coded in a 4×4 matrix with increments of 0.25 in p_1 and p_2 . This matrix has the next values on each axis: 0.25, 0.5, 0.75 and 1.0 (the values are uniformly distributed in the range (0, 1]). In this example, we assume that the decision matrix has already been updated via the improvement process. Thus, the values of the cells that were used during the training have already been assigned one of the four possible ordering heuristics. The points that were not visited during the training process do not have one heuristic assigned. In case an unseen instance is presented to the hyper-heuristic and it forces the matrix to visit one of these unassigned cells, the value for that decision point will be determined using the default ordering heuristic. Figure 1 shows the hyper-heuristic discussed. Suppose that an instance P is presented to this hyper-heuristic and the features of instance P are $p_1 = 0.8$ and $p_2 = 0.70$ (during the search, p_1 is estimated as the number of non empty edges in the constraint graph over the maximum possible number of edges and p_2 is obtained by the average of the tightness of all the constraints). In that case, the cell to be accessed in the decision matrix would be (2, 2), which corresponds to the values $p_1 = 0.75$ and $p_2 = 0.75$, due to the resolution of the matrix. In this example, heuristic 0 would be selected as ordering heuristic. If we change the resolution of

the matrix, the number of cells will change along with the values of the constraint density and tightness coded in the axes. Once the ordering heuristic has been used, the original instance P is transformed into P' , which values of p_1 and p_2 may be different from P . If we continue applying the hyper-heuristic, we will be moving through the decision matrix. Every time we get closer to the solution, we also get closer to the origin of the decision matrix. When we finally find a solution to the instance, the cells visited in the matrix allow us to map the search tree for that specific instance.

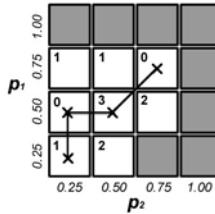


Fig. 1. An example hyper-heuristic

The local improvement approach uses a very simple method to upgrade the matrices and improve the performance of the hyper-heuristics. The method is based on the idea of changing the decision matrix, one cell at the time, and accepting the change only if it improves the performance of the hyper-heuristic on a set of instances (training set). Basically, we are implementing a hill climbing search on the space of heuristics. Every time we change the ordering heuristic corresponding to one decision point, the rest of the search tree from that node may be different. Thus, new points will be activated in the decision matrix and their cells initialized. If we observe the search tree on the space $p_1 \times p_2$, it is clear that, as we get closer to the solution (the deeper nodes in the search tree), we find the smaller the values of p_1 and p_2 . Based on this observation we concluded that the nodes of the tree which are located further from the origin are visited first than the closer ones. We need to keep a record of the number of changes per cell in the matrix, otherwise the improvement method will always change the same decision point. Thus, we proposed an order to change the decision points in the matrix: the further points from the origin that have not achieved the maximum number of changes have priority to be updated. Because the process is not randomized, if we run the process twice with the same parameters we will obtain the same results. The decision points to be updated depend only on the instances solved during the training and the cells visited because of the search. The local improvement process is described in the following lines:

1. Initialize the decision matrix hyper-heuristic.
2. Solve the instances in the training set with the hyper-heuristic and obtain the average consistency checks per instance ($avg_0(HH)$).
3. Update the decision matrix. Only one cell is changed according to the criteria already described.

4. Solve the instances in the training set with the updated hyper-heuristic and obtain the average consistency checks per instance ($avg(HH)$). If $avg(HH) < avg_0(HH)$, make $avg_0(HH) = avg(HH)$ and accept the change. Otherwise, cancel the change and return the decision matrix to the previous configuration.
5. Repeat from step 3 until the maximum number of cycles is reached.

3 Experiments and Results

In order to test the model proposed we developed three experiments. In the first experiment we focus on the generation of hyper-heuristics using a small hard training set (Set A) and later, we test those hyper-heuristics on a larger set of hard unseen instances (Set B) with similar properties than the instances inside the training set. The second experiment tries to obtain new hyper-heuristics using the larger set (Set B) and analysing the results to see whether the training on those specific instances can improve the results of the hyper-heuristics obtained in the first experiment. The third and final experiment tests the performance of the hyper-heuristics trained with Set B on unseen instances from Set C which includes hard and not so hard instances.

3.1 Experiment I

In this experiment we produced four hyper-heuristics, each one using a different default ordering heuristic. The size of the matrices was set to 10 and for each hyper-heuristic the local improvement process ran for 30 cycles. In this experiment, the instances from Set A were used as input for the process. Table 1 presents the average number of consistency checks that any of the four hyper-heuristics uses at the start of the process, at the end of the improvement process and the reduction achieved through the process with respect to the initial hyper-heuristic. One conclusion derived from Table 1 is that hyper-heuristics HH01-04 have the same performance on Set A in terms of consistency checks. It is important to mention that the decision matrices from these hyper-heuristics are not equal, but some decision points share the same ordering heuristics. This means that even with a different combination of heuristics we can achieve the same performance. Because each hyper-heuristic is using a different default ordering heuristic to initialize the decision matrix at the beginning of the process, it is clear to see that the average consistency checks that MXC/MINC requires to solve each instance in Set A is the same that the average obtained by the hyper-heuristic that uses it as default heuristic at the beginning of the process, it is, 15834. Following the same reasoning, to solve each instance in Set A, SD/MINC, MXC/MXC and SD/MXC require 9709, 16032 and 9995 consistency checks, respectively. In all the cases the hyper-heuristics are able to overcome the average consistency checks required by the pure ordering heuristics.

Once we trained hyper-heuristics HH01-04, it is time to test whether they are able to perform well on a wider set of instances with similar features to those

Table 1. Performance of HH01-04 on Set A

HH	Default H	Avg. at start	Avg. at end	Reduction
HH01	MXC/MINC	15834	7947	50%
HH02	SD/MINC	9709	7947	19%
HH03	MXC/MXC	16032	7947	51%
HH04	SD/MXC	9995	7947	21%

used during the training. We used HH01-04 to solve Set B without any additional training. The results presented in Table 2 are still competitive, but not as good as in the previous test. Even though the hyper-heuristics are able to overcome the results of MXC/MINC and MXC/MXC, the hyper-heuristics are not able to beat SD/MINC and SD/MXC. This decrease in the performance is justified because the hyper-heuristics were not trained for those specific instances. Even though no additional training was performed, HH01-04 provide good results when compared to the simple ordering heuristics: the average consistency checks required by HH02 to solve each instance of Set B is 11.57% above the average of the best ordering heuristic (which in this case is SD/MINC with 7989 consistency checks). The results from this test are presented in Table 2.

Table 2. Performance of HH01-04 on Set B

H	Avg(Checks)	HH	Avg(Checks)
MXC/MINC	20308	HH01	8913
SD/MINC	7989	HH02	8914
MXC/MXC	19170	HH03	8913
SD/MXC	8553	HH04	8913

3.2 Experiment II

We have already argued that one possible cause of the decrease in the performance of HH01-04 when tested on Set B is the fact that they were not trained for such set. In an attempt to confirm this idea, we produced four new hyper-heuristics (HH05-08) as described in Experiment I, but this time we used Set B as training set during the process. The results shown in Table 3 confirm the idea that the difference in the performance of HH01-04 with respect to the two sets was not because of the model itself, but the training instances used during the process. Table 3 presents the results of hyper-heuristics HH05-08 on Set B. As we can observe, these hyper-heuristics provide better results than the previous HH01-04. The difference in the performance is notorious, and three of the four hyper-heuristics are able to perform better than each one of the simple ordering heuristics (even though in the case of SD/MINC the difference is not significant). HH07 does not provide results as good as the other hyper-heuristics but it is still better than any of the previous HH01-04.

Table 3. Performance of HH05-08 on Set B

HH	Default H	Avg. at start	Avg. at end	Reduction
HH05	MXC/MINC	20308	7959	61%
HH06	SD/MINC	7989	7959	1%
HH07	MXC/MXC	19170	8856	54%
HH08	SD/MXC	8553	7960	7%

3.3 Experiment III

In the previous experiments we trained and tested our hyper-heuristics using only hard instances. We also need to consider the performance of these hyper-heuristics on other instances with similar features, but not as difficult as the ones presented before. The instances from Set C were solved with HH05-08 and compared against the performance of the corresponding default ordering heuristic (Figure 2).

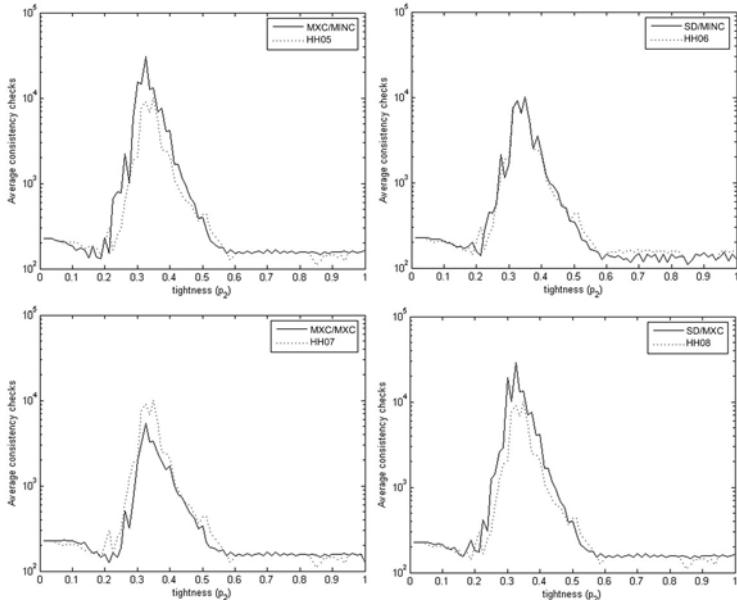


Fig. 2. Performance of HH05-08 and their corresponding default ordering heuristic on Set C

In Figure 2 we can observe that HH05 and HH08 clearly outperform their respective default ordering heuristics. Specially at the peak located at the transition phase, each of these two heuristics dominate the ordering heuristics. In the case of HH06 the performance is not very clear. Both the behaviour of HH06 and SD/MINC describe a very similar curve for values of $p_2 < 0.6$. For larger values of p_2 , HH06 is dominated by SD/MINC. Finally, without any additional

training, HH07 was not able to compete with MXC/MXC. This ordering heuristic looks very robust for the instances contained in Set C. It is important to recall that the instances in Set C were never seen before by the hyper-heuristics and also, they have features different to those used during the training.

4 Conclusions

The solution model presented in this paper produces good quality hyper-heuristics which are able to compete with the simple ordering heuristics on the different sets used in this investigation.

The hyper-heuristics produced for one specific set reduce their performance when applied to unseen instances with similar features but they are still competitive. Even though the hyper-heuristics presented achieved promising results we still need to work more in order to obtain better quality hyper-heuristics that can be applied to a wider range of instances without having to retrain them.

As future work we have considered the inclusion of new ordering heuristics and new ways to test the performance of the hyper-heuristics, not only the average consistency checks on the sets. We also know it is a common practice to use randomly generated instances in CSP studies but we are aware that it is not enough to test our approach. We want to test our hyper-heuristics on real instances taken from existing CSP repositories.

Acknowledgments. This research was supported in part by ITESM under the Research Chair CAT-144 and the CONACYT Project under grant 99695.

References

1. Bittle, S.A., Fox, M.S.: Learning and using hyper-heuristics for variable and value ordering in constraint satisfaction problems. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, GECCO 2009, pp. 2209–2212. ACM, New York (2009)
2. Brelaz, D.: New methods to colour the vertices of a graph. Communications of the ACM 22 (1979)
3. Burke, E., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Rong, Q.: A survey of hyper-heuristics. Tech. Rep. NOTTCS-TR-SUB-0906241418-2747, School of Computer Science, University of Nottingham (2009)
4. Burke, E., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Woodward, J.R.: A classification of hyper-heuristic approaches. Tech. Rep. NOTTCS-TR-SUB-0907061259-5808, School of Computer Science, University of Nottingham (2009)
5. Chakhlevitch, K., Cowling, P.: Hyperheuristics: Recent developments. In: Cotta, C., Sevaux, M., Sørensen, K. (eds.) Adaptive and Multilevel Metaheuristics. SCI, vol. 136, pp. 3–29. Springer, Heidelberg (2008)
6. Cheeseman, P., Kanefsky, B., Taylor, W.M.: Where the really hard problems are. In: Proceedings of IJCAI 1991, pp. 331–337 (1991)

7. Denzinger, J., Fuchs, M., Fuchs, M., Informatik, F.F., Munchen, T.: High performance atp systems by combining several ai methods. In: Proc. Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 1997, pp. 102–107. Morgan Kaufmann (1997)
8. Fisher, H., Thompson, G.L.: Probabilistic learning combinations of local job-shop scheduling rules. In: Factory Scheduling Conference. Carnegie Institute of Technology (1961)
9. Fukunaga, A.S.: Automated discovery of local search heuristics for satisfiability testing. *Evolutionary Computation* 16, 31–61 (2008)
10. Gent, I.P., MacIntyre, E., Prosser, P., Smith, B.M., Walsh, T.: An Empirical Study of Dynamic Variable Ordering Heuristics for the Constraint Satisfaction Problem. In: Freuder, E.C. (ed.) CP 1996. LNCS, vol. 1118, pp. 179–193. Springer, Heidelberg (1996)
11. Mackworth, A.K.: Consistency in networks of relations. *Artificial Intelligence* 8(1), 99–118 (1977)
12. Minton, S., Johnston, M.D., Phillips, A., Laird, P.: Minimizing conflicts: A heuristic repair method for csp and scheduling problems. *Artificial Intelligence* 58, 161–205 (1992)
13. Ortiz-Bayliss, J.C., Terashima-Marín, H., Özcan, E., Parkes, A.J.: Mapping the performance of heuristics for constraint satisfaction. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (July 2010)
14. Ortiz-Bayliss, J.C., Terashima-Marín, H., Özcan, E., Parkes, A.J.: On the idea of evolving decision matrix hyper-heuristics for solving constraint satisfaction problems. In: Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO 2011, pp. 255–256. ACM, New York (2011)
15. Prosser, P.: Binary constraint satisfaction problems: Some are harder than others. In: Proceedings of the European Conference in Artificial Intelligence, Amsterdam, Holland, pp. 95–99 (1994)
16. Smith, B.M.: Locating the phase transition in binary constraint satisfaction problems. *Artificial Intelligence* 81, 155–181 (1996)
17. Terashima-Marín, H., Ortiz-Bayliss, J.C., Ross, P., Valenzuela-Rendón, M.: Hyper-heuristics for the dynamic variable ordering in constraint satisfaction problems. In: GECCO 2008: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation. ACM (2008)
18. Tsang, E., Kwan, A.: Mapping constraint satisfaction problems to algorithms and heuristics. Tech. Rep. CSM-198, Department of Computer Sciences, University of Essex (1993)

Improving the Performance of Heuristic Algorithms Based on Causal Inference

Marcela Quiroz Castellanos¹, Laura Cruz Reyes¹, José Torres-Jiménez²,
Claudia Gómez Santillán¹, Mario César López Locés¹, Jesús Eduardo Carrillo Ibarra¹,
and Guadalupe Castilla Valdez¹

¹ Instituto Tecnológico de Ciudad Madero, México

lcruzr@prodigy.net.mx,

{qc.marcela,mario.cesar2509,jesus.carrillo1987}@gmail.com,
{cggs71,jgvaldez50}@hotmail.com

² CINVESTAV-TAMAULIPAS, México

jnj@cinvestav.mx

Abstract. Causal inference can be used to construct models that explain the performance of heuristic algorithms for NP-hard problems. In this paper, we show the application of causal inference to the algorithmic optimization process through an experimental analysis to assess the impact of the parameters that control the behavior of a heuristic algorithm. As a case study we present an analysis of the main parameters of one state of the art procedure for the Bin Packing Problem (BPP). The studies confirm the importance of the application of causal reasoning as a guide for improving the performance of the algorithms.

Keywords: weight annealing, bin packing problem, causal inference, parameter adjustment, tuning, performance evaluation.

1 Introduction

Causal inference has led to explanations of important events in areas such as biology, education, medicine, among others. A detailed analysis of the information contained in the problems, covering a period of time and characterized by indices, leads to explanations of facts and behaviors [1]. One of the main challenges in the analysis of heuristic algorithms is the identification of strategies and conditions that favor its performance. Causal inference can be used to build models that explain the performance of algorithms for hard combinatorial problems, however little causal modeling work has been carried out to explain the behavior of heuristic algorithms [2, 3, 4, 5].

The importance of causal inference is that it allows understanding the inherent relationships between a set of factors by means of a causal model that summarizes the underlying structure of the data under study. Some of the main contributions provided by the study of the causal relations of the factors that impact the algorithm performance are: the creation of policies on how to deal with a situation [6, 7], the development of theories of explanation of the performance [3, 4, 5] and the definition of guidelines to improve the performance [2, 5].

In this work, we present a study of the configuration of the main parameters that define the behavior of one heuristic algorithm for the bin packing problem (BPP). BPP is considered NP-hard [8, 9] and consists in packing a set of items of different sizes in the minor number of fixed size bins without violating the capacity of any bin. BPP has an extensive number of industrial and logistic applications and frequently happens as a sub-problem in several practical problems [10]. We show the application of causal inference to obtain a broad perspective of the parameters that directly affect the overall performance of one high performance procedure named WABP (Weight Annealing for Bin Packing) [11]. For this study we evaluated the performance of the algorithm with different types of standard problems considered challengers that result difficult for it. The algorithm was executed with different configurations of its main parameters in order to make a causal analysis of the obtained results and understand the impact of each parameter on the final performance. The PC learning algorithm of the TETRAD tool [12] was applied for automatically learning a causal model from experiments results.

Relevant parameter tuning methods have been proposed, however, most of them just look for the best configuration that meets some success criterion (for instance, beats some actual benchmark) and do not provide any knowledge about the relations between parameters and performance [13]. We propose a scientific methodology that is concerned with gaining insights into an algorithm through controlled experimentation that aims to study the effects of parameters on algorithm behavior (performance) by means of a causal study.

This paper continues in Section 2 with an introduction of causal inference. Section 3 presents some important concepts of causal models. Section 4 includes a brief description of an algorithm for constructing a model out of experimental data. Next, Section 5 presents a case study that exemplifies the application of causal modeling in the analysis of the impact of the main control parameters in one state of the art algorithm. Finally, Section 6 presents conclusions and suggests future work.

2 Learning Explanations through Causal Inference

Causal inference combines mathematical, probabilistic and philosophical concepts. The mathematical part deals with the directed acyclic graphs (DAGs) that allow to represent causal models. The probabilistic part focuses on conditional independence of variables: two events are conditionally independent, given a third event, if the occurrence of one does not alter the probability of occurrence of the other whenever it is known the third [14]. The philosophical part of the causal inference involves the causality between variables [15].

The term causality indicates how the world responds to an intervention; it describes the relation between causes and events. If an event A is the cause of an event B three conditions have to be fulfilled: 1) that A happens before B , 2) that B happens always than A happen, and 3) A and B are close in space and time. By repeatedly observing the fulfillment of these conditions it can be generalize that since until now B has always happen after A , in the future will happen the same. This can establishes a law.

To further claim that the relationship is causal is to assert that if we were to intervene to make A happen, then consequently B will happen.

Causal inference goes beyond most of the statistical analysis, because their goal is not only likely to infer assumptions or probabilities under static conditions, but also the dynamics of assumptions under changing conditions, such as changes induced by external interventions. This distinction implies that association and causality do not mix: "correlation does not imply causation". Therefore, while an associative concept is any relationship that can be defined in terms of a joint distribution of observed variables, a causal concept is any relationship that cannot be defined only from the distribution. Associational assumptions, even untested, are testable in principle, given sufficiently large sample and sufficiently fine measurements. Causal assumptions, in contrast, cannot be verified, unless one resorts to experimental control [16].

3 Causal Models

This subsection will briefly introduce qualitative causal reasoning and causal models [1, 16, 17]. Causal reasoning is related to the conditions under which the study is performed as well as a specified set of variables (called the causal system). For a given effect variable, a causal assignment is an assignment of a particular value to each variable in the causal system besides the effect. A response structure for an effect is a description of what value the effect variable takes on in every possible causal assignment.

Two causal assignments $C1$ and $C2$ are a test pair of causal assignments for a possible cause variable x if and only if they are identical except for the values assigned to variable x .

In a system of variables S , x is a direct cause of y relative to S if and only if there is at least one test pair of causal assignments for x in the response structure for y across which y differs. Informally, if we can intervene to change x while holding everything else fixed, and still produce a change in y , even the chances of y , then x is a direct cause of y .

A causal model is a generalized representation of knowledge obtained by finding dependencies that imply cause and effect relationships. Causality requires identifying the direct relationship between events or variables. The most common representation of a causal model is through a directed acyclic graph (DAG) that shows the causal influences between the variables of the system and helps to estimate the total and partial effects resulting from the manipulation of a variable. A causal graph includes a set of variables, and a set of directed edges that connect pairs of these variables. The edges are "directed" because causation is asymmetric and has a direction. If one variable x is a direct cause of another variable y in some causal system S , then we include an arrow, or directed edge, from x to y in the causal graph that represents S .

The scientific search for causal knowledge can be viewed as a struggle against the problem of alternative theories. Faced with limited knowledge and evidence, many causal theories might explain the evidence and be consistent with our background knowledge. We can reduce the number of alternative theories by strengthening our

background knowledge; for example, we might assume that variables cannot cause (the values of) variables that were measured earlier. The most powerful technique for eliminating alternative causal theories is an experiment in which we can control or intervene on the values of variables.

4 Learning Causal Models from Data

The algorithms used in the literature for the discovery of explanations in data sets are based on the calculation of correlation and independence to find possible causal relationships. For example, if two variables are statistically independent, we can affirm that between them there is no causal relationship. Learning a causal model presents a high degree of complexity due to the number of possible structures. However, prior knowledge of a set of independent relationships between variables can simplify the calculations. There are efficient learning algorithms that use prior knowledge and some properties based on heuristic rules to construct DAGs that allow to make causal conclusions [1, 18].

The PC algorithm is the most common algorithm for constructing a model out of experimental data [1, 2]. It starts by constructing an undirected graph by finding direct relations. These have the property that they cannot become probabilistically independent upon conditioning on some other set of vertices. After that, the algorithm tries to direct the relations by looking for causal independences and by applying heuristics. If the algorithm cannot decide upon the direction, it returns a bidirectional arrow or an undirected arrow. The TETRAD tool [12] supports discrete and continuous variables, but for continuous variables it assumes linear dependencies and multivariate normality. Furthermore, domain knowledge can be inserted before executing learning algorithms, like known independencies.

5 Causal Analysis of Parameters in WABP Algorithm

In order to show the application of causal inference in the analysis of performance of algorithms, we made a study of the principal parameters of a heuristic algorithm for the bin packing problem called WABP [11]. WABP algorithm makes use of the concept of weight annealing to expand and speed up the neighborhood search by creating distortions in different parts of the search space.

5.1 Weight Annealing Algorithm for the Bin Packing Problem

WABP is a very simple algorithm, illustrated in the flowchart in the Figure 1, which given the parameters K , $nloop$, T and $Tred$, and an instance defined by the values of C , the bin capacity, t_j , the size of the items, n , the number of items and LB , a lower bound of the optimal solution of the given instance, starts with a solution generated by the well-known deterministic heuristic for BPP: first fit decreasing [19], which is then improved by the weight annealing strategy. The algorithm iterates for a fixed number

of cycles where the filling of containers, as well as the weights of the objects are distorted by the function $calcWeights(T, k)$, to support moves that allows escape from local optima, caring to not obtain an infeasible solution. The size of the distortion is controlled by a temperature parameter that is decreased by $reduceTemp(T, Tred)$ until the distorted solution matches the real solution, given by LB . Finally, the algorithm returns the obtained solution.

The variables considered in the causal analysis are: the scaling parameter K , which controls the amount of size distortion for each object; $nloop$, the number of iterations to be performed; the temperature parameter T , which controls the amount by which a single weight can be varied and the accepted moves in the search space; and $Tred$, the percentage in which T is reduced in each iteration of the algorithm.

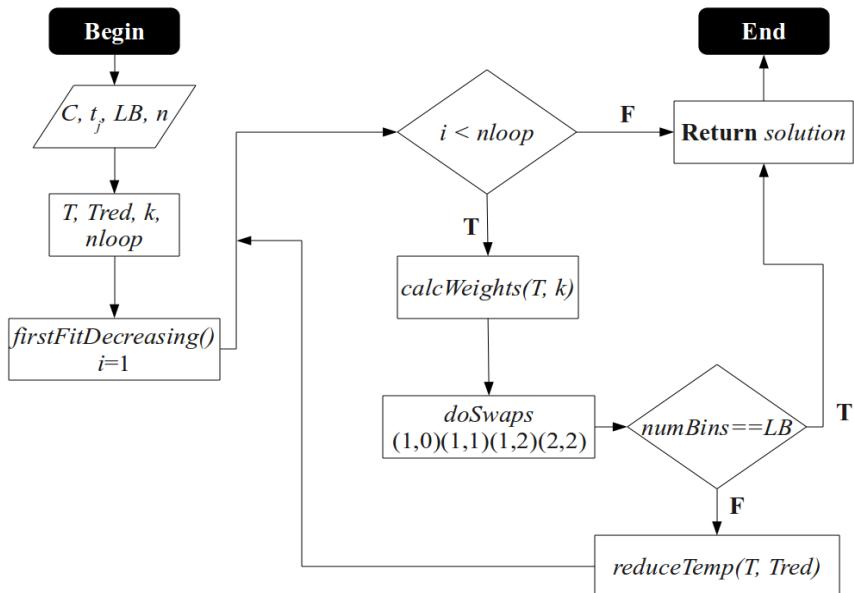


Fig. 1. The WABP procedure

5.2 Experimental Design

The compiling suite g++ in its version 4.4.3 was used to implement the WABP algorithm in object code. The performance of the algorithm was evaluated with different types of standard problems considered challengers. The trial cases had been selected by different authors to compare the effectiveness of its strategies with the one of other algorithms of the state of the art. Such benchmark instances can be found on recognized websites [20, 21, 22, 23]. We take some instances in every class for which WABP could not find the optimal solutions. Table 1 presents the characteristics and the number of instances considered in every set to evaluate the performance of the algorithm.

Table 1. Characteristics and number of cases taken of every set of BPP instances

Set	No. of inst.	No. of items	Bin capacity
Uniform	4	120, 250, 500	150
Triplet	80	60, 120, 249, 501	1000
DataSet1	11	100, 200, 500	100, 120, 150
DataSet2	5	50, 100, 500	1000
DataSet3	2	200	100000
Was1	1	100	1000
Was2	2	120	1000
Gau	17	[57, 239]	10000
Hard28	28	[160, 200]	1000

Four possible values were established for each of the parameters of interest K , $nloop$, T and $Tred$, as shown in Table 2.

Table 2. Values of parameters used in the configurations

Category	K	$nloop$	T	$Tred$
1	0	50	1	0.98
2	0.01	100	5	0.9
3	0.03	150	10	0.85
4	0.08	200	20	0.8

The combinations of the values of the parameters, taking into account all possible causal assignments, give a total of 256 different configurations. For each configuration we evaluated the performance of WABP on all 150 instances. Thus, an experiment consists in resolving all instances of all sets described, with a particular combination of values to the considered parameters. For each experiment the number of instances for which the algorithm finds the optimal solution and the average number of iterations required to find the solutions were recorded. The execution of the experiments was carried out on a Dell PowerEdge 2600 with 4 GB RAM, 70 GB of storage and an Intel Xeon at 3.06 GHz running the Ubuntu 8.04 operating system.

Table 3. Sample of the results obtained by WABP in different configurations

Configuration	K	$nloop$	T	$Tred$	#Optimal	#Iterations
1	0.01	100	1	0.8	23	89
2	0.03	200	10	0.98	36	174
3	0.08	100	5	0.85	31	86
4	0	200	20	0.85	27	176
5	0	50	5	0.98	23	45

We programmed WABP procedure, obtaining similar results to the reported by Fleszar and Charalambous [24], so we agree with the authors in the fact that the

published results by Loh et al. [11] are incorrect. Table 3 contains a sample of the results of 5 of the 256 experimental configurations performed. The second column is the parameter K , the third is $nloop$, the fourth is T , the fifth column is $Tred$, the sixth column is the number of optimal solutions and the last column is the average number of iterations that it took to find the solution of the 150 instances.

Table 4. Range of values in the categorization for the number of optimal solutions

Category	Range
1	[10, 19]
2	[20, 28]
3	[29, 37]
4	[38, 46]

5.3 Learning a Causal Model from the Experimental Data

The results obtained in the experimentation were analyzed with the TETRAD causal analysis tool in the version 4.3.10 [12]. Since the range of the number of optimal solutions was considerably large, it was reduced to four, as shown in Table 4, which was obtained by dividing by four the difference between the smallest and largest number of optimal solutions obtained during the experiment.

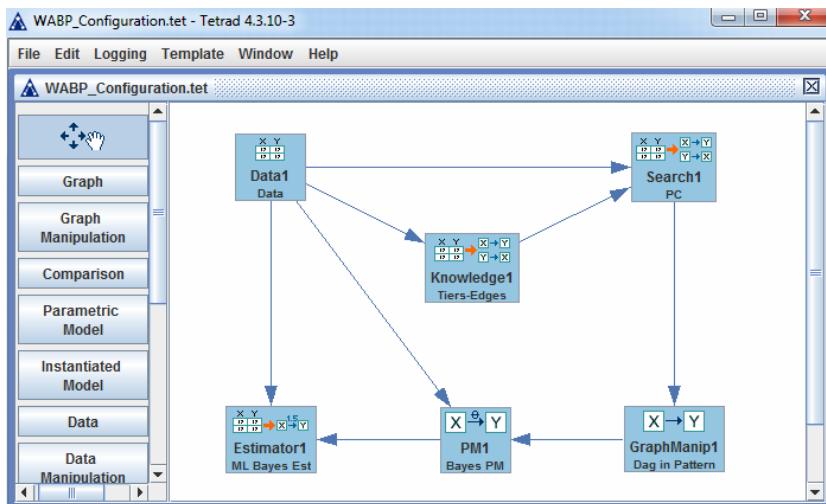


Fig. 2. TETRAD session

Once categorized all data, it was proceeded to create a new session of TETRAD as shown in Figure 2. The module Data1 is used to enter a data table for a future causal analysis. The module Knowledge1 allows to add previous knowledge about the relationships between different variables, we established some limitations in the form of prior knowledge to avoid relationships that may be considered absurd, such as that the

performance of the algorithm is a cause of the values of the parameters. Next, the module Search1 executes the PC algorithm learning a causal graph out of experimental data that describes the relationships between the variables under analysis. The module GraphManip1 permits to obtain the causal graph that can be used in future analysis. The module ParametricModel (PM1) summarizes the different values that define each of the variables. Finally, the module Estimator1, estimates the relationships between each variable and its causes to complement the causal model.

The pattern of relationships obtained through the causal analysis with TETRAD is presented in Figure 3, which clearly shows that the variables that influence directly the number of optimum obtained are K , T and $Tred$. Also shows that the average number of iterations required by the algorithm is caused by the parameter K , T , $Tred$ and $nloop$.

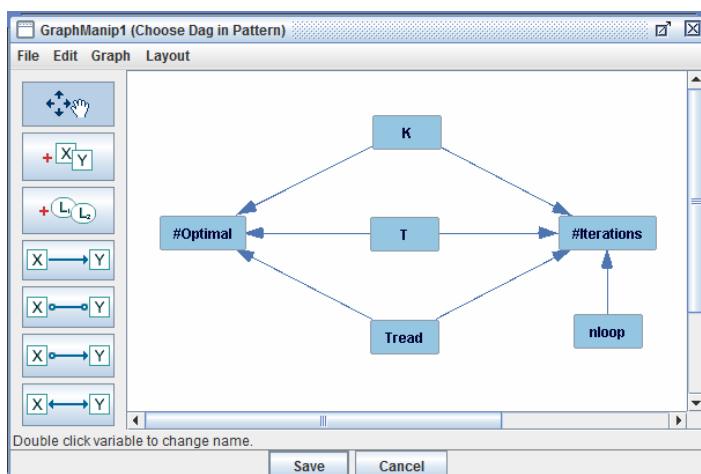


Fig. 3. Causal Graph generated by TETRAD

The complete causal model is obtained in the Estimator1 module through a table that includes the frequency distribution model that summarizes the impact of the parameters in the behavior of the algorithm. Table 5 shows a sample of the frequency distribution model. Obtained results show in detail how the parameters affect the algorithmic performance.

After analyzing the causal model and its corresponding frequency distribution table we can infer the following conclusions, note that when we refer to a value, we are talking about the categories defined in Table 2:

- When $K = 1$, which means that there is not distortion in the size of the elements, the values assigned to T , the initial temperature, and $Tred$, the percentage in which temperature is reduced, do not influence the number of optimal solutions obtained by the algorithm.
- For the parameter $Tred$ that decreases the value of the initial temperature T , if the value of the latter is low and the parameter K has medium values (2 and 3),

$Tred$ does not affect the number of optimal obtained, doing so only when the percentage of distortion is high ($K = 4$).

- The best results are obtained when T and $Tred$ are low, which means low initial temperatures decreasing very slowly, this is because it is more likely that the algorithm allows good solutions, exploring further regions of the search space.
- The worst results occur when the value of K and T are high and $Tred$ is low. It can be seen that with a high temperature and a disturbance equally high, solutions that are more likely to worsen the current solution are accepted, even when the decreasing speed of the temperature is low.
- The configuration that is capable of obtaining the highest number of optimal solutions is the one that include a medium degree of disturbance, and low levels of temperature and percentage of temperature reduction ($K = 0.03$, $T = 1$ and $Tred = 0.98$), this allows to explore a larger search space, accepting particularly those solutions that represent a real improvement in quality.

Table 5. A sample of the frequency distribution model

<i>K</i>	<i>T</i>	<i>Tread</i>	#Optimal=1	#Optimal=2	#Optimal=3	#Optimal=4
1	1	1	0.0000	1.0000	0.0000	0.0000
1	4	4	0.0000	1.0000	0.0000	0.0000
2	1	1	0.0000	0.5000	0.5000	0.0000
2	1	2	0.0000	1.0000	0.0000	0.0000
2	1	4	0.0000	1.0000	0.0000	0.0000
2	2	1	0.0000	0.2500	0.5000	0.2500
2	2	3	0.0000	0.2500	0.7500	0.0000
3	1	1	0.0000	0.0000	0.2500	0.7500
3	4	1	0.2500	0.2500	0.5000	0.0000
4	1	1	0.0000	0.2500	0.5000	0.2500
4	4	4	0.0000	0.0000	1.0000	0.0000

Table 6. Values of the parameters and number of optimal solutions found for WABP

<i>K</i>	<i>T</i>	<i>Tred</i>	#Optimal
0	1, 5, 10, 20	0.98, 0.9, 0.85, 0.8	[20, 28]
0.01	1, 5, 10, 20	0.8	[20, 28]
0.01, 0.3	1, 5	0.9, 0.85, 0.8	[20, 28]
0.01, 0.03, 0.08	1, 5	0.98	[20, 46]
0.08	10, 20	0.9, 0.85, 0.8	[20, 37]
0.08	10, 20	0.98	[10, 37]

The analysis of the frequency relative table of the causal model shows that the parameter that has the highest impact in the performance of the algorithm is the amount of size distortion for each object (K), followed by the parameter that regulate the temperature in each iteration $Tred$. Table 6 summarizes the real values of each parameter that have an influence in the number of optimal solutions found. WABP shows the

best quality in the solutions when the value of $Tred$ is 0.98, but these cases occur only when the values of K are different from 0 (only when the problem space is distorted) and the initial temperature has small values. It is interesting to note that when K and T present their maximum values and $Tred = 0.98$ the algorithm shows the worst performance, observing how all the parameters interact to cause the final results of the algorithm.

When we executed WABP with the configuration purpose by the Loh et al. [11] the algorithm obtains the optimal solution for only 27 of the 150 instances. With the best configuration we obtain in our experimental design we outperform these results finding 46 optimal solutions with means an improvement of 70.37% in the effectiveness of the algorithm. Table 7 details the number of optimal solutions obtained for the original and the new configuration in every set of instances. We believe that the performance of WABP can still be improved exploring values of the parameters closest to the ones that obtain the best results.

Table 7. Effectiveness of the algorithm WABP with the old and the new configuration

Set	# of inst.	# of optimal with old configuration	# of optimal with new configuration
Uniform	4	1	2
Triplet	80	0	15
DataSet1	11	3	5
DataSet2	5	1	1
DataSet3	2	2	2
Was1	1	1	1
Was2	2	2	2
Gau	17	12	13
Hard28	28	5	5
Total	150	27	46

6 Conclusions and Future Work

This paper shows that the causal analysis is useful in the tuning and improvement of heuristics algorithms, because it allows identifying clearly what are the parameters that impact the final performance and to what extent they do. The utility of causal inference was shown by studying a heuristic algorithm for the bin packing problem.

We presented a case of study where we analyzed the parameters that control the algorithm WABP, obtaining a causal model which showed that the configuration that achieves the highest number of optimal solutions is the one where the amount of size distortion for the objects (K) is medium and the initial temperature (T) and the percentage of reduction in temperature ($Tred$) are low. The selected configuration outperforms the effectiveness of the original algorithm in a 70.37%. It was also noted that the parameter most influential in obtaining optimal solutions is the amount of size distortion for each object (K), followed by the parameters that regulate the temperature

(T and T_{red}). On the other hand it could be seen that the number of possible iterations appear to have little impact on the optimal number.

As future work, in the case of WABP algorithm, we are planning to make a deeper study within the parameter of distortion of the objects in the range of [0.01, 0.05], the initial temperature between the values (0, 1] and the reduction factor temperature in the range (0.90, 1), which seem to be the most promising values to increase the efficiency of the algorithm. It would also be interesting to design a new causal model for WABP that includes, in addition, variables that characterize the instances of BPP in order to obtain more detailed explanations of the algorithmic performance.

In general, it is expected that the work presented in this paper represents a guideline to study the performance of heuristic algorithms through the application of causal inference in other algorithms and optimization problems. Causal inference allows to obtain a deep understanding about algorithmic behavior, this knowledge can be used to improve the performance.

References

1. Spirtes, P., Glymour, C., Scheines, R.: *Causation, Prediction, and Search*, 2nd edn. MIT Press (2001)
2. Lemeire, J.: Learning Causal Models of Multivariate Systems and the Value of it for the Performance Modeling of Computer Programs. PhD thesis. Vrije Universiteit Brussel (2007)
3. Pérez, V.: Modelado Causal del Desempeño de Algoritmos Metaheurísticos en Problemas de Distribución de Objetos. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México (2007)
4. Pérez, J., Cruz, L., Pazos, R., Landero, V., Reyes, G., Zavala, C., Fraire, H., Pérez, V.: A Causal Approach for Explaining Why a Heuristic Algorithm Outperforms Another in Solving an Instance Set of the Bin Packing Problem. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) *Foundations of Intelligent Systems*. LNCS (LNAI), vol. 4994, pp. 591–598. Springer, Heidelberg (2008)
5. Quiroz, M.: Caracterización de Factores de Desempeño de Algoritmos de Solución de BPP. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México (2009)
6. Cruz, L.: Caracterización de Algoritmos Heurísticos Aplicados al Diseño de Bases de Datos Distribuidas. PhD Thesis, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Morelos, México (2004)
7. Álvarez, V.: Modelo para representar la Complejidad del problema y el desempeño de algoritmos. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México (2006)
8. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company (1979); A classic introduction to the field
9. Basse, S.: *Computer Algorithms, Introduction to Design and Analysis*. Editorial Addison-Wesley Publishing Company (1998)
10. Cruz Reyes, L., Nieto-Yáñez, D.M., Rangel-Valdez, N., Herrera Ortiz, J.A., González, B.J., Castilla Valdez, G., Delgado-Orta, J.F.: DiPro: An Algorithm for the Packing in Product Transportation Problems with Multiple Loading and Routing Variants. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) *MICAI 2007*. LNCS (LNAI), vol. 4827, pp. 1078–1088. Springer, Heidelberg (2007)

11. Loh, K., Golden, B., Wasil, E.: Solving the one-dimensional bin packing problem with a weight annealing heuristic. *Computers & Operations Research* 35, 2283–2291 (2008)
12. The TETRAD Project: Causal Models and Statistical Data. TETRAD Homepage: <http://www.phil.cmu.edu/projects/tetrad/>
13. Eiben, A., Smit, S.: Parameter Tuning for Configuring and Analyzing Evolutionary Algorithms. *Swarm and Evolutionary Computation* 1(1), 19–31 (2011)
14. Nadkarni, S., Shenoy, P.: A bayesian network approach to making inferences in causal maps. *European Journal of Operational Research* 128, 479–498 (2001)
15. Vila, M., Sánchez, D., Escobar, L.: Relaciones Causales en Reglas de Asociación. In: XII Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF 2004 (2004)
16. Pearl, J.: Causal inference in statistics: An overview. *Statistics Surveys* 3, 96–146 (2009)
17. Carnegie Mellon University: Causal and Statistical Reasoning. Open Learning Initiative (2011), <http://oli.web.cmu.edu/openlearning/>
18. Kalisch, M., Bühlmann, P.: Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm. *Journal of Machine Learning Research* 8, 613–636 (2007)
19. Johnson, D.S.: Fast algorithms for bin packing. *Journal of Computer and System Sciences* 8(3), 272–314 (1974)
20. Beasley, J.E.: OR-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society* 41(11), 1069–1072 (1990),
<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/binpackinfo.html>
21. Klein, R., Scholl, A.: Bin Packing benchmark data sets,
<http://www.wiwi.uni-jena.de/Entscheidung/binpp/>
22. Euro Especial Interest Group on Cutting and Packing. One Dimensional Cutting and Packing Data Sets,
http://paginas.fe.up.pt/~esicup/tiki-list_file_gallery.php?galleryId=1
23. Cutting and Packing at Dresden University. Benchmark data sets.,
<http://www.math.tu-dresden.de/~capad/cpd-ti.html#pmp>
24. Fleszar, K., Charalambous, C.: Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem. *European Journal of Operational Research* 210(2), 176–184 (2011)

Fuzzified Tree Search in Real Domain Games

Dmitrijs Rutko^{*}

Faculty of Computing, University of Latvia
Raina blvd. 19, Riga, LV-1586, Latvia
dim_rut@inbox.lv

Abstract. Fuzzified game tree search algorithm is based on the idea that the exact game tree evaluation is not required to find the best move. Therefore, pruning techniques may be applied earlier resulting in faster search and greater performance. Applied to an abstract domain, it outperforms the existing ones such as Alpha-Beta, PVS, Negascout, NegaC*, SSS*/ Dual* and MTD(f). In this paper we present experimental results in real domain games, where the proposed algorithm demonstrated 10 percent performance increase over the existing algorithms.

Keywords: game tree search, alpha-beta pruning, fuzzified search algorithm, performance.

1 Introduction

Games have attracted intellectual resources of mankind for ages. For AI researchers the nature of games is a challenging subject for study. Usually it is easy to represent the state of a game, but it is remarkably hard to solve, as like in the real world in many cases it is infeasible to find the optimal solution.

Games are usually represented in their extensive form with the help of a game tree which starts at the initial position and contains all the possible moves from each position. Classical game tree search algorithms such as Minimax and Negamax operate using a complete scan of all the nodes of the game tree and are considered to be too inefficient. The most practical approaches are based on the Alpha-beta pruning technique, which seeks to reduce the number of nodes to be evaluated in the search tree. It is designed to stop the evaluation of a move completely if at least one possibility is found, the one that proves the current move to be worse than the previously examined move. Such moves do not need to be evaluated further.

More advanced extensions and additional improvements of Alpha-Beta are known as PVS, Negascout and NegaC*. Another group of algorithms like SSS* / Dual* and MTD(f) is based on best-first search in a game tree. Potentially, they could be even more efficient, however, they typically require high memory consumption.

Through analyzing and comparing these algorithms it can be seen that in many cases the decision about the best move can be made before the exact game tree minimax value

^{*} This research is supported by the European Social Fund project No. 2009/0138/1DP/1.1.2.1.2/09/IPIA/VIAA/004.

is obtained. The paper aims to introduce a new approach which allows to find the best move faster while visiting less nodes.

The paper is organized as follows: the current status in the game tree search is discussed. Then, the idea that allows to perform game tree search in a manner based on the move that leads to the best result is proposed. Following this, the algorithm structure and implementation details are explained. Thereafter, the experimental setup and empirical results on the search performance obtained in a real domain are shown. The paper is concluded with future research directions.

2 Current Situation in Game Tree Search

Classical game tree search algorithms are based on the Alpha-beta pruning technique. Alpha-beta is a search algorithm which tries to reduce the number of nodes to be evaluated in the search tree by the Minimax algorithm. It completely stops evaluating a move when at least one possibility has been found that proves the move to be worse than the previously examined one. Such moves need not be evaluated further. When applied to a standard minimax tree, it returns the same move as minimax would, but prunes away branches that cannot possibly influence the final decision [11].

The illustration of the Alpha-beta approach is given in **Fig. 1**.

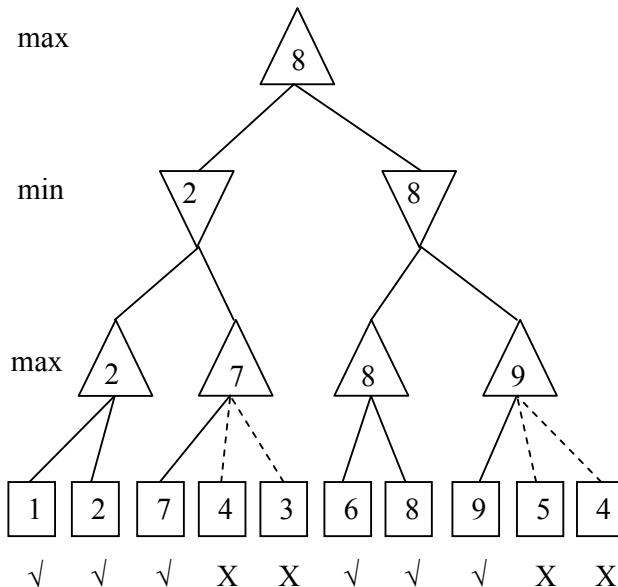


Fig. 1. The Alpha-Beta approach

The game tree in **Fig. 1** has two branches with minimax values 2 and 8 for the left and right sub-trees respectively. In order to find the best move, the Alpha-beta algorithm scans all the sub-trees from left to right and is forced to evaluate almost

each node. The possible cut-offs are depicted with a dashed line (at each step, the previous evaluation is smaller than the value of currently checked node).

When all the nodes are checked, the algorithm compares the top-level sub-trees. The evaluation of the left and the right branches are 2 and 8 respectively; the highest outcome is chosen, and the best move goes to the right sub-tree.

The main benefit of alpha-beta pruning lies in the fact that many branches of the search tree can be eliminated. Thus the search can be limited to a 'more promising' sub-tree, and a deeper search can be performed in the same time.

Another way of reducing search space is using *null window* or *zero window* approach which is naturally a boolean test that checks whether a move produces a better or worse result than a passed value. Many algorithms incorporate this idea [11].

Since the minimax algorithm and its variants are inherently depth-first, a strategy such as *iterative deepening* is usually used in conjunction with alpha-beta so that a reasonably good move can be returned even if the algorithm is interrupted before it has finished execution. Iterative deepening runs repeatedly, increasing the depth limit with each iteration until it reaches d , the depth of the shallowest goal state. Another advantage of using iterative deepening is that searches at shallower depths give move-ordering hints that can help produce cutoffs for higher-depth searches much earlier than would otherwise be possible [11].

More advanced algorithms are the following:

- *PVS* (Principal Variation Search) is an enhancement to Alpha-Beta based on *null* or *zero window* searches of none PV-nodes to prove whether a move is worse or not than the already safe score from the principal variation [1, 10].
- *NegaScout*, which is an Alpha-Beta enhancement. The improvements rely on a Negamax framework and some fail-soft issues concerning the two last plies which did not require any re-searches [3, 4].
- *NegaC** – an idea to turn a Depth-First to a Best-First search like MTD(f) to utilize null window searches of a fail-soft Alpha-Beta routine and to use the bounds that are returned in a bisection scheme [5].
- *SSS** and its counterpart *Dual** are search algorithms which conduct a state space search traversing a game tree in a best-first fashion similar to that of the A* search algorithm and retain global information about the search space. They search fewer nodes than Alpha-Beta in fixed-depth minimax tree search [2].
- *MTD(f)*, the short name for MTD(n, f), which stands for Memory-enhanced Test Driver with node n and value f. MTD is the name of a group of driver-algorithms that search minimax trees using null window alpha-beta with transposition table calls. In order to work, MTD(f) needs the first guess as to where the minimax value will turn out to be. The better this first guess is, the more efficient the algorithm will be, on average, since the better it is, the less passes the repeat-until loop will have to do to converge on the minimax value [6-9].

Transposition tables are another technique which is used to speed up the search of the game tree in computer chess and other computer games. In many games, it is possible to reach a given position, which is called transposition, in more than one way. In general, after two moves there are 4 possible transpositions since either player may swap their move order. So, it is still likely that the program will end up analyzing the same position several times. To avoid this problem transposition tables store previously analyzed positions of the game [11].

3 The Fuzzy Approach

We propose a new approach which is based on the attempt to implement a human way of thinking adapted to logical games. A human player rarely or almost never evaluates a given position precisely. In many cases, the selection process is limited to rejecting less promising nodes and making certain that the selected option is better than others. The important point is that we are not interested in the exact position evaluation but in the node which guarantees the highest outcome.

Let the given problem be explained in details.

We could look at our game tree from a relative perspective like “if this move is better or worse than some value X” (**Fig. 2**). At each level, we identify if a sub-tree satisfies “greater or equal” criteria. So passing search algorithm, for instance, with argument 5, we can obtain the information that the left branch has a value less than 5 and the right branch has a value greater or equal than 5. We do not know exact sub-tree evaluation, but we have found the move, which leads to the best result.

In this case, different cut-offs are possible:

- at max level, if the evaluation is greater (or equal) than the search value;
- at min level, if the evaluation is less than the search value.

In the given example, the reduced nodes are shown with dashed line. Comparing to **Fig. 1** it can be seen that not only more cut-offs are possible, but also pruning occurs at a higher level which results in a better performance.

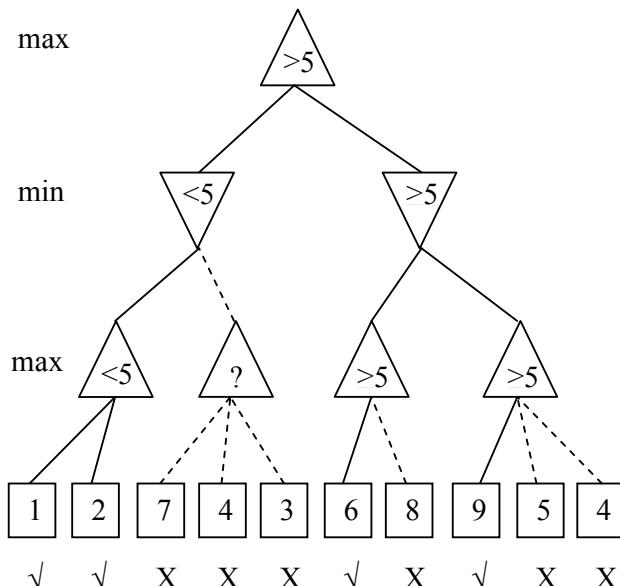


Fig. 2. Fuzzy best node approach

In this approach, the best/worst cases are the same as for alpha-beta pruning: $O(w^{d/2})$ for the best case as only one branch should be checked at cut-off level, and $O(w^d)$ for worst case as all nodes should be checked (w is width, d is depth of the tree). But in the presented approach, cut-offs are more often possible in general.

If we use geometric interpretation and put our sub-tree minimax values on coordinate axis, then our task is to separate/divide branches so that only one branch would have higher value than the test value. **Fig. 3** illustrates our previous example. Alpha-beta window is initially set to leaf node range $\alpha = 0, \beta = 10$; then the following test values are used X_1, X_2, X_3 . If value X_2 is chosen, then the successful separation is obtained after the first iteration as we know that the second sub-tree has higher estimation. If values X_1 or X_3 are chosen, then no separation is possible at this point – both values are on the same side of the test value. In this case, the algorithm continues with reduced alpha-beta search window: 1) $\alpha = X_1$ in the first case; or 2) $\beta = X_3$ in the second.

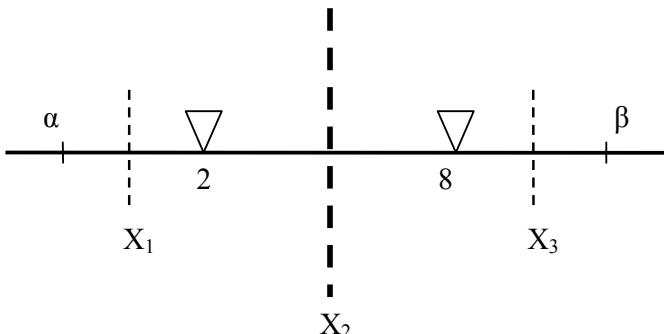


Fig. 3. Geometric interpretation of separation in the fuzzified game tree search

In a game tree with three or more sub-trees, the algorithm workflow remains the same. Our task is to separate sub-trees in a way that only one branch has higher estimation than the test value. However, more cases are possible – 0, 1, 2, 3 branches fall in on one side of separation line. In this case, alpha-beta window is reduced correspondingly and the algorithm proceeds with the next iteration.

Comparing to the existing algorithms such as MTD(f) in order to work, it needs the first guess as to where the minimax value will turn out to be. If you feed MTD(f) the minimax value to start with, it will only do two passes, the bare minimum: one to find an upper bound of value x , and one to find a lower bound of the same value.

In the presented algorithm, it is possible to find the best move with a single iteration and we are not limited to the accurate first guess. For the presented example, any value from interval 3..7 (inclusive) would apply.

4 The Fuzzified Search Algorithm

Best Node Search (BNS) is a new game tree search algorithm based on the idea described in the previous section. The main difference between the classical approach

and the proposed algorithm is that BNS does not require the knowledge of the exact game tree minimax value to select a move. We only need to know which sub-tree has higher estimation. By iteratively performing search attempts the algorithm can obtain information about which branch has higher estimation without knowing the exact value. So less information is required and, as a result, the best move can be found faster – total number of searched nodes is smaller and total algorithm execution time is reduced comparing to the algorithms based on the exact game tree evaluation.

The presented algorithm uses a standard call of Alpha-Beta search with ‘zero window’. The proposed implementation relies on the transposition tables but variation without memory (transposition tables) usage is also possible. While scanning a game tree, the algorithm checks all sub-trees and returns a node which leads to the best result. In general, BNS is expected to be more efficient comparing to the classical algorithms in terms of number of nodes checked as it does not obtain additional information which is not required in many cases – the exact game tree minimax value.

The BNS algorithm is given in **Fig. 4** which makes use of the following functions:

1. NextGuess() – returns the next separation value to be tested by the algorithm;
2. AlphaBeta() – alpha-beta search with Zero Window (Null Window) performs a boolean test whether a move produces a worse or a better score than the passed value.

All sub-trees are tested with separation values (this information is stored in the transposition tables and reused in subsequent iterations). If exactly one branch exceeds test value, then the best node is found. If all branches have smaller estimation, then the number of sub-trees that exceeds separation test value remains the same and beta value is reduced. If several nodes exceed test value, then subtreeCount is updated correspondingly, and alpha value is updated to test value, and algorithm continues with the next iteration. If a single sub-tree that exceeds test value cannot not be found and alpha-beta range is reduced to 1, it means that several sub-trees have the same estimation and we can choose any of them.

```

function BNS(node,  $\alpha$ ,  $\beta$ )
    subtreeCount := number of children of node
    do
        test := NextGuess( $\alpha$ ,  $\beta$ , subtreeCount)
        betterCount := 0
        foreach child of node
            bestVal := -AlphaBeta(child, -test, -(test - 1))
            if bestVal  $\geq$  test
                betterCount := betterCount + 1
                bestNode := child
        update number of sub-trees that exceeds separation test
        value
        update alpha-beta range
    while not(( $\beta - \alpha < 2$ ) or (betterCount = 1))
    return bestNode

```

Fig. 4. The BNS algorithm

One of the main parts of this algorithm is the method `NextGuess(α, β, subtreeCount)` which returns the next value to be checked by the algorithm. In the simplest case, it could be a formula based on linear distribution – alpha-beta range is proportionally divided into sections according to the sub-tree count:

```
NextGuess = α + (β - α) * (subtreeCount - 1) / subtreeCount;
```

where alpha and beta are the lower and the upper bounds of the search window respectively; `subtreeCount` is the number of sub-trees which satisfies the previous test call (the branches that have higher estimation than the test value).

Moreover dynamic adjusting of separation value provided by function `NextGuess` is also possible. If during our search we can conclude that we should update separation value in order to increase probability of successful separation we can apply it. For example if the current node has a higher or equal estimation with test the value, then it is reasonable to increase this test value and check remaining nodes with new updated test value.

However, the best algorithm performance is achieved after its statistical training or analytical game tree evaluation, which gives accurate information about resulting game tree value distributions. As a result, `NextGuess` also becomes more precise.

Some algorithms, such as MTD(f) benefit from accurate “first guess” – as to where the minimax value will turn out to be. The better than first guess is, the more efficient the algorithm will be, on average.

The BNS algorithm can significantly benefit from good separation value as well. The better separation value is, the faster the best node will be found, on average. One approach to improve algorithm performance is to enhance it through statistical training. The main idea is to collect statistical information over series of games in order to analyze and find optimal separation values for the algorithm [12].

Another approach is based on game tree analytical evaluation where probability density functions and cumulative distribution functions are applied in order to calculate the most probabilistic outcome at each level of the game tree [13].

5 Game Setup

For our research we were seeking for a not complex game (like Chess or Go) to be implementable in this study. But on the other hand selected game should not be too simple to be computable completely.

So for our experiments we have chosen “Hey! That's My Fish!” game which is the perfect match for our needs – it is simple but with some subtle strategy behind [14].

“Hey! That's My Fish!” is a 2-4 player board game. The aim is to collect as many fish as you can with your penguins.

Setup: 60 hexes are randomly laid out in 8 rows, alternating between 7 and 8 hexes each - **Fig. 5**. They are all face up so that you can clearly see where the fish clusters are. Each hex has either 1, 2, or 3 fish on it. Each player then places 2-4 penguins (depending on the number of players). They place these one at a time and they must be placed on “1” fish hexes.



Fig. 5. “Hey! That’s My Fish!” game board

- Two Player game: each person has 4 penguins
- Three Player game: each player has 3 penguins
- Four Player game: each player has 2 penguins

Play: On his turn a player moves one of his penguins. He must move it in a straight line, and may move it as little as 1 hex and as many hexes as is legal. Penguins must stop before they reach: the edge of the board; a break in the hexes; or another penguin. After moving his penguin the player then picks up the hex which his penguin started on.

Ending the Game: a player leaves the game when he can't move any more of his penguins. (He takes the last hexes that his penguins are standing on.) When all the players have done moving, the game is over. (Practically, the game actually ends when each player can see that his penguins are each on their own "islands" of ice. Each player then picks up any hexes on these islands which he could reach.) The players then count up all their fish, and the player with the most wins.

However, there is also a fair amount of tactical content in the game. The moves are not entirely obvious. You can be very aggressive in the game, making clever blocking moves and carefully analyzing the vector-based movement [15].

The main strategic element of gameplay is working to isolate other players' penguins, trapping them in small areas; when this happens, and an area is isolated that contains only one penguin, the owner scores all tiles in the isolated area, and the penguin is removed from the play. Maximizing score by getting high-fish hexes is a secondary but important strategic consideration [16].

6 Experimental Results

The entire framework was created for the given game and mentioned algorithms were implemented and tested in real domain. Now we are presenting some additional details on the implementation.

Evaluation Function

To perform game tree search a straight-forward evaluation function was implemented. At each step we count the number of fish consumed so far, so overall evaluation function is the amount of fish obtained by the current player minus the amount of fish obtained by the opponent limited by search depth.

$$\text{Evaluation function} = \text{Fish Amount (player)} - \text{Fish Amount (opponent)}$$

The main advantages of this approach are the following: it is fast, simple, reasonable, and it converges to correct estimation towards the game end. Downsides of this function are that it does not address specific aspects of the game like strategic area isolation and opponent blocking, but we are mainly focused on search algorithm comparison, which gives us a wide area for research.

Nevertheless, the current implementation achieves a reasonable level of play provided by the program. It allows 6 ply search depth in the beginning of the game, 10 ply in mid-game, and reaching search depth of 14 ply in the end-game. For details refer to **Fig. 6**. This search depth generally depends on the decreasing number of moves available during the game. There are approximately 30-50 moves in the initial stage, around 10-30 in the mid-game, and less than 10 moves available in the end-game. The game typically terminates in 40-50 total moves.

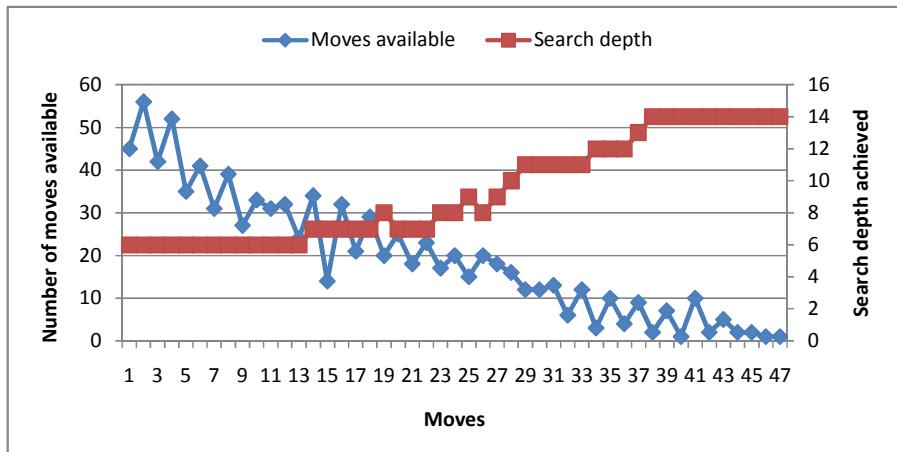


Fig. 6. The number of moves available vs. achieved search depth

Iterative Deepening

Both MTD(f) and BNS algorithms can benefit from good estimation of possible outcome. MTD(f) uses first guess and BNS uses separation value as their initial seeds. Both goals can be efficiently reached with iterative deepening technique - strategy in which a depth-limited search is run repeatedly, increasing the depth limit with each iteration.

At each iteration MTD(f) algorithms returns the exact game tree value, so next time it uses exact values from the previous step (from previous search level). On the other hand BNS algorithms returns lower bound estimation of the game tree, which means that exact value could be higher. Nevertheless, as it can be seen in the charts this prediction is successfully used in consequent searches. Please note that the evaluation function is symmetric in its nature so for odd search depth estimation from previous odd depth should be used, and for even search depth estimation from previous even depth should be used correspondingly.

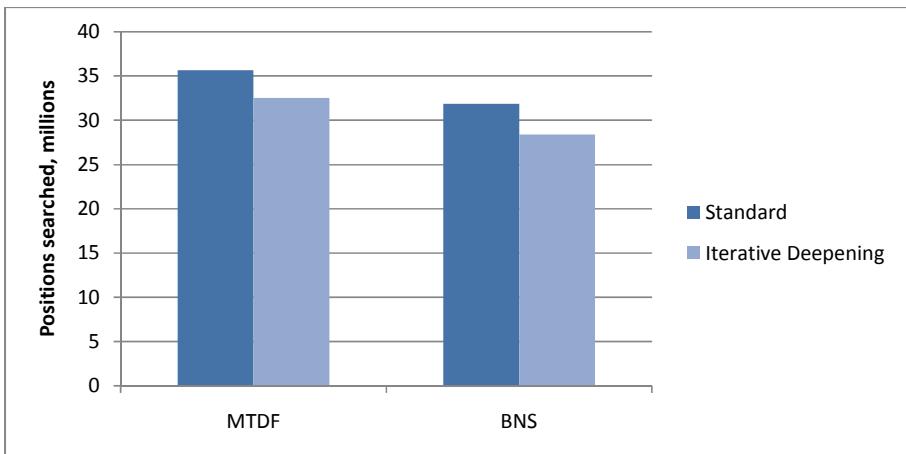


Fig. 7. Performance improvement with Iterative Deepening

Fig. 7 shows the total number of positions searched (leaf nodes visited / evaluation function calls) by both algorithms during the entire game. It could be seen that introducing *iterative deepening* technique brings the overall improvement in 9-11 percent for each algorithm, and exactly these implementations were used in the further experiments.

Transposition Tables

Many algorithms are based on re-iterative search approaches and for example MTD(f), BNS and others can speed up their search by using transposition tables. However in our experiments versions without TT were used, as generally implementation of TT requires high memory consumption, and our main focus was to verify algorithm behavior in real domain games.

Another group of algorithms including SSS* and Dual* are usually efficient with TT only [13]. So they were not used in our experiments.

Algorithm Performance Comparison

These algorithms were implemented and tested in “Hey! That’s My Fish!” game. The results of one particular match are shown in **Fig. 8**. Despite the fact that the initial board position is random each time, different match series produce quite similar results.

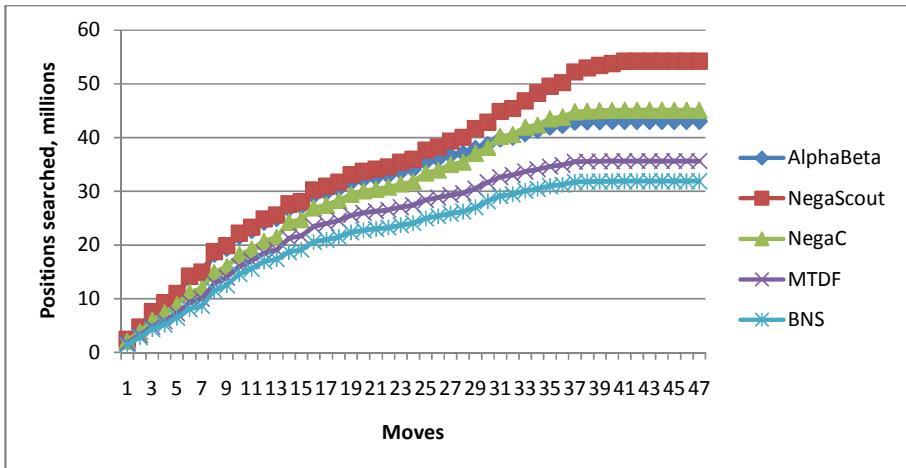


Fig. 8. Number of positions searched within the game

The total number of positions searched (number of evaluation function calls / leaf nodes visited) was collected and analyzed during the game. This figure presents cumulative statistics, meaning that total number of leaf nodes is counted during the game, which increases from the initial stage of the game to the game end. This particular game has 47 moves in total.

Fig. 9 demonstrates the same data but from another perspective. In this chart MTD(f) performance as taken as base point (treated as 100 percent), and all others algorithms are measured as ratio to it.

In these diagrams it can be seen that NegaScout and AlphaBeta algorithms have similar performance during the first half of the game, however towards to the end efficiency of NegaScout is decreased. Similarly NegaC shows better results during initial phases of the game, but again performance also decreased at the end. Apparently the main reason is that NegaScout / NegaC are more effective with high branching factor and less effective when the average number of moves is decreasing. Among these algorithms MTD(f) has best results.

On the other hand, proposed BNS algorithm demonstrates the performance improvement in approximately 15 percent in the first three moves, then goes a bit below 10 percent, and converges to stable 10 percent speed up at the game end.

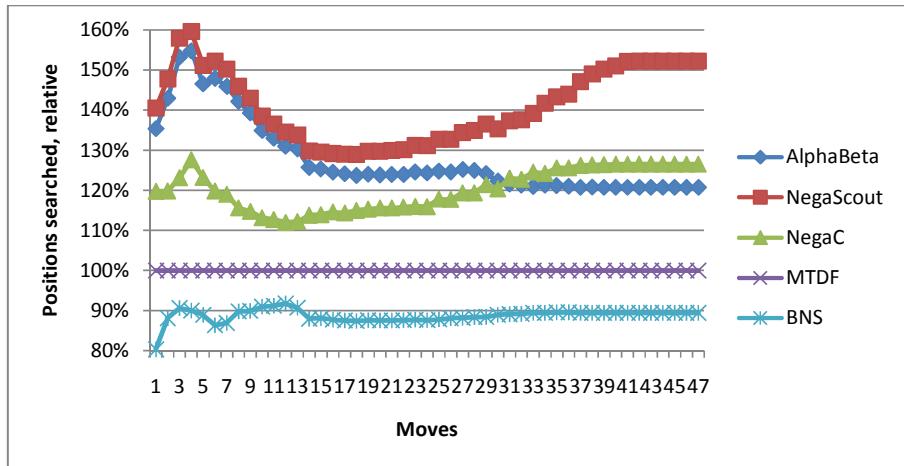
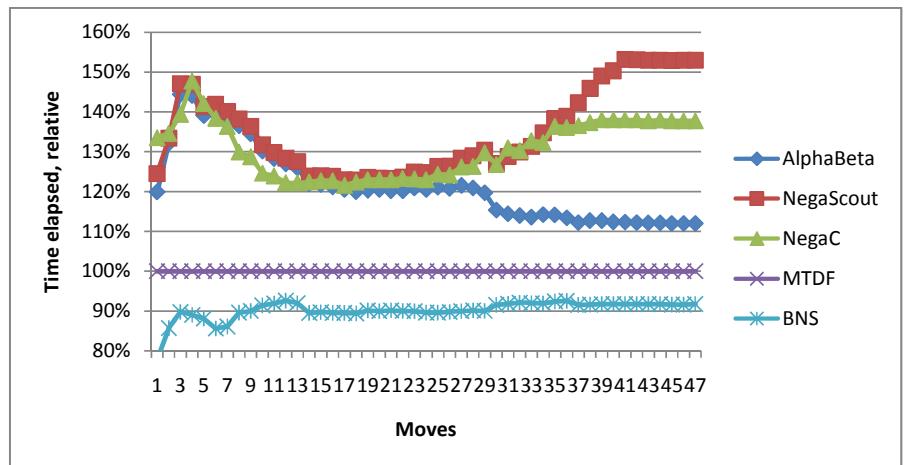
**Fig. 9.** Relative number of positions searched within the game**Fig. 10.** Relative time elapsed within the game

Fig. 10 demonstrates relative time elapsed by algorithms. As it could be seen BNS algorithm practically does not introduce any computational overhead compared to others.

7 Conclusions and Future Work

The main goal of this paper was to show that the proposed BNS algorithm could be also efficient in real domain games. The experiments demonstrate that BNS gives a 10 percent performance improvement over MTD(f) algorithm, which is comparable

with expected results achieved in experiments in abstract domain [13]. It can be concluded that BNS demonstrates good potential and could be used as a general purpose game tree search algorithm.

However additional research might be needed in the following areas:

- The implementation and analysis of transposition tables, which can potentially increase the performance of the algorithms;
- The usage of different knowledge based or heuristic based evaluation functions;
- The implementation of a multi-player game. The existing game provides flexible extension up to 4 players.

The future experiments should also consider analyzing algorithm performance in other games, but we believe that the proposed approach could be successfully applied for any type of game as well.

References

1. Marsland, T.A., Campbell, M.: Parallel Search of Strongly Ordered Game Trees. ACM Comput. Surv. (1982)
2. Pearl, J.: The solution for the branching factor of the alpha-beta pruning algorithm and its optimality. Communications of the ACM (1982)
3. Reinefeld, A.: An Improvement to the Scout Tree-Search Algorithm. ICCA Journal 6(4), 4–14 (1983)
4. Reinefeld, A.: Spielbaum-Suchverfahren. Informatik-Fachbericht, vol. 200. Springer, Heidelberg (1989)
5. Weill, J.C.: The NegaC* search. ICCA Journal (March 1992)
6. Plaat, A., Schaeffer, J., Pijls, W., de Bruin, A.: A New Paradigm for Minimax Search, Technical Report EUR-CS-95-03 (1994)
7. Plaat, A., Schaeffer, J., Pijls, W., de Bruin, A.: Best-First and Depth-First Minimax Search in Practice. In: Proceedings of Computing Science in The Netherlands (1995)
8. Plaat, A., Schaeffer, J., Pijls, W., de Bruin, A.: An Algorithm Faster than NegaScout and SSS* in Practice. In: Computer Strategy Game Programming Workshop at the World Computer Chess Championship (1995)
9. Plaat, A., Schaeffer, J., Pijls, W., de Bruin, A.: Best-First Fixed-Depth Minimax Algorithms. Artificial Intelligence 87 (1996)
10. Björnsson, Y.: Selective Depth-First Game-Tree Search. Ph.D. thesis, University of Alberta (2002)
11. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Pearson Education, Inc., Upper Saddle River (2010)
12. Rutko, D.: Fuzzified Algorithm for Game Tree Search. In: Second Brazilian Workshop of the Game Theory Society, BWGT 2010 (2010)
13. Rutko, D.: Fuzzified Algorithm for Game Tree Search with Statistical and Analytical Evaluation, Scientific Papers, vol. 770, pp. 90–111. University of Latvia (2011)
14. Hey That's My Fish - German Board Games,
<http://www.mrbass.org/boardgames/heythatsmyfish/>
15. Review of Hey! That's My Fish! – RPGnet,
<http://www.rpg.net/reviews/archive/11/11936.phtml>
16. Hey! That's My Fish! Remarkable Depth for Minimal Rules,
<http://playthisthing.com/hey-thats-my-fish>

On Generating Templates for Hypothesis in Inductive Logic Programming

Andrej Chovanec and Roman Barták

Charles University in Prague, Faculty of Mathematics and Physics

Malostranské nám. 25, 118 00 Praha 1, Czech Republic

chovanec.a@gmail.com, bartak@ktiml.mff.cuni.cz

Abstract. Inductive logic programming is a subfield of machine learning that uses first-order logic as a uniform representation for examples and hypothesis. In its core form, it deals with the problem of finding a hypothesis that covers all positive examples and excludes all negative examples. The coverage test and the method to obtain a hypothesis from a given template have been efficiently implemented using constraint satisfaction techniques. In this paper we suggest a method how to efficiently generate the template by remembering a history of generated templates and using this history when adding predicates to a new candidate template. This method significantly outperforms the existing method based on brute-force incremental extension of the template.

Keywords: inductive logic programming, template generation, constraint satisfaction.

1 Introduction

Inductive logic programming (ILP) is a discipline investigating invention of clausal theories from observed examples and additional background knowledge. Formally, for a given set of positive examples E^+ and a set of negative examples E^- we are finding a hypothesis H such that H entails all examples from E^+ and does not entail any example from E^- (a so-called *consistent hypothesis*). Background knowledge is a domain-specific knowledge which is available prior to learning process. Through this paper we suppose without loss of generality that background knowledge is empty. Since logical entailment is not complete in first-order logic θ -subsumption is used as a decidable restriction of logical entailment [8]. In our work, examples are represented as sets of fully instantiated atoms and hypothesis is a set of atoms with variables. In such setting, the process of inventing the consistent hypothesis H consists of (1) determining its structure, that is which atoms and how many of them are in the hypothesis, and (2) finding the unifications between variables in the atoms in such way that we obtain a consistent hypothesis. Hypothesis structure is called a *template* and hypothesis H is formed from template T by applying a substitution θ of variables such that $T\theta = H$. For instance, $T = \{\text{arc}(X_1, X_2), \text{arc}(X_3, X_4)\}$ is a template consisting of two atoms and four variables and $H = \{\text{arc}(X_1, X_2), \text{arc}(X_2, X_1)\}$ is one

particular hypothesis obtained from the template T by applying substitution $\theta = \{X_3/X_2, X_4/X_1\}$. Depending on the fact whether we do or do not know the template *a priori*, we speak about *template ILP consistency* in case we do know the template, and about *general ILP consistency* in case we do not know the template. Although both consistency problems belong to the same complexity class [5], the general task deals with an extra problem of determining the structure of the template, which has a radical impact on the overall performance of ILP system. Based on this fact, there are huge differences in running time between algorithms solving general and template ILP consistency problems. This situation motivated us to study the problem of general ILP consistency and we developed a new algorithm which should make solving general consistency more efficient.

In this paper we present a new approach to generate templates for the general ILP consistency problem. Our work focuses on a method how the initial template is generated. The method uses existing algorithms to transform the template into a consistent hypothesis by finding certain unifications between variables. We propose a generate-and-test algorithm which successively generates templates until it is possible to generate a consistent hypothesis from the template. The algorithm remembers the previously generated templates and their properties and takes them into account when generating a next template. In other words, the algorithm is continuously learning itself from the rejected templates in order to improve templates generated in later iterations.

The paper is organized as follows. In section 2 we will introduce existing algorithms used for the template ILP consistency problem. These algorithms are used as subroutines in our template generating algorithm. Simultaneously we will describe a basic version of the template generating algorithm. Section 3 will give the main result of the paper – we will propose how to extend the basic template generating algorithm with several improvements and extensions yielding a self-learning template generating algorithm which is significantly more efficient. In section 4 we will further enhance the algorithm and finally in section 5 we will present experimental results comparing the effectiveness of the proposed methods. In the whole paper we will work with examples of identifying common structure (i.e. hypothesis) in random graphs (i.e. positive and negative examples).

2 Background

As we have mentioned in the introduction, ILP deals with the problem of finding a hypothesis that covers a set of positive examples and excludes a set of negative examples. Hypothesis and examples are supposed to be clauses and the hypothesis is required to logically entail all *positive examples* and no *negative example*. Entailment is checked using θ -*subsumption* [8] which is a decidable restriction of logical entailment. We will assume the clauses to be expressed as sets of literals, and, without loss of generality, we will only work with positive literals, that is, non-negated atoms. All terms in learning examples (hypotheses, respectively) will be constants (variables) written in lower (upper) cases. For instance, $E = \{\text{arc}(a,b), \text{arc}(b,c), \text{arc}(c,a)\}$ is an

example and $H = \{\text{arc}(X,Y), \text{arc}(Y,Z)\}$ is a hypothesis. Hypothesis H *subsumes* example E , if there exists a substitution θ of variables such that $H\theta \subseteq E$. In the above example, substitution $\theta = \{X/a, Y/b, Z/c\}$ implies that H subsumes E .

In order to develop a general ILP system based on the generate-and-test approach, we must specify algorithms for three major components of the system. These components include generating the template, testing θ -subsumption and finally deciding template consistency. In this case, the ILP algorithm usually repeats the following three steps:

- generate a template,
- find unifications of variables in the template to obtain a hypothesis,
- test consistency of a hypothesis using θ -subsumption check.

If the hypothesis is not consistent, the algorithm tries other unifications and if all unifications are exhausted, it generates a new template.

The θ -subsumption check and deciding template consistency have already been addressed by the utilization of constraint satisfaction (CSP) techniques. θ -subsumption check is a crucial part of ILP systems and thus it naturally motivated new efficient approaches to be developed. Maloberti and Sebag proposed an efficient CSP algorithm Django [6] that significantly outperformed existing systems. Dramatic speed-up brought by Django encouraged exploitation of CSP techniques in other parts of ILP systems. In [2] Barták suggested a novel approach utilizing CSP towards deciding template consistency. His work was based on finding unifications between variables in the template to obtain a consistent hypothesis. The idea is as follows. We start with a hypothesis consisting of mutually different variables (the template) and then we systematically search the space of possible unifications until we obtain a consistent hypothesis or the search exhausts all possibilities. Barták proposed a sophisticated CSP model that extensively prunes the space of all unifications and thus keeps runtime in very sensible bounds.

For generating the template, Barták used a simple algorithm based on the generate-and-test method that works as follows: first we recall that each variable in the template appears exactly once (it is the goal of template consistency to decide which variables in atoms should be unified to obtain a consistent hypothesis). Hence template generation is about deciding how many atoms of each predicate symbol will appear in the template. In [2] this is done by exploring systematically all possible templates with the increasing length of template. Let k denote the number of different predicate symbols in all examples (in case the background knowledge is not empty, also predicate symbols from the background knowledge should be considered), for example $\{\text{arc}(a,b), \text{arc}(b,c), \text{arc}(c,a)\}$ contains one predicate symbol (arc) and three atoms for this predicate symbol. To simplify notation, we will write “predicate p ” instead of writing “the atom of predicate symbol p ”. For example, adding predicate p to template means adding a new atom of the predicate symbol p with fresh variables to the template. The algorithm starts with a template of length k containing one atom for each predicate symbol and as soon as the constraint model for template consistency rejects the template, a new template is generated. A template space is searched systematically by generating all possible templates of length $k, k+1, k+2, \dots$

The complete algorithm solving the general ILP consistency problem may look like in Figure 1. As the stop criterion we can use either reaching of the maximal k or reaching the maximal runtime.

```

1:  $k \leftarrow$  number of all different predicates in examples
2: repeat
3:    $S_T \leftarrow$  all templates of length  $k$ 
4:   for each  $T \in S_T$  do
5:      $H \leftarrow$  Decide-Consistency( $T$ )
6:     if  $H$  is consistent then return  $H$ 
7:   end for
8:    $k \leftarrow k + 1$ 
9: until stop criterion satisfied

```

Fig. 1. A basic generate-and-test template generating algorithm (IDS)

We can notice easily that algorithm in Figure 1 is performing iterative deepening search (IDS). An advantage of this approach is that it guarantees finding the shortest consistent hypothesis. However, we are not always interested in finding the shortest possible solution. In fact, the trade-off between finding the optimal hypothesis and the cost of this search is so cumbersome that this method becomes impractical, especially for longer templates (see the section with experimental results). Hence we suggest a different method for exploring the space of templates based on incremental extension of the template by adding new predicates.

3 Incremental Template Generation

When we analyzed reasons why the IDS algorithm was not capable to find more complicated (longer) hypotheses, we find out it is because of its “leaps” between two distinct parts of the template space. Think of the following example: we are solving a problem having the final template equal to $5 \times a, 5 \times b, 5 \times c$ (5 predicates a , 5 predicates b , and 5 predicates c) and we have examined all templates of length 14. Let us suppose without loss of generality, that the last examined template of the length 14 was $4 \times a, 5 \times b, 5 \times c$. We can see that it is sufficient to add predicate a to the template. Despite, the basic systematic algorithm starts blindly generating all templates of the length 15 from the beginning until it reaches the correct template.

Based on the above observation we suggest searching the space of templates only one way – new predicates can be added to the template and once they are added, they cannot be removed. Further, we need to determine which predicate is due to be added to the template in each step of the algorithm. We can use systematic search that adds different predicates one by one in cycles – add predicate a , followed by predicate b , followed by predicate c , and again predicate a etc. Or we can add each predicate with certain probability. If the probability of adding predicate to the template is distributed among all predicates uniformly, the earlier and the latter approaches are identical in

terms of developing the template. Both approaches are identical because the probability of the case when the probabilistic approach would significantly prefer adding one particular predicate (the number of its occurrences would be k-times of occurrences of other predicates for any constant $k > 0$) diminishes exponentially (a so-called Chernoff bound [7]). However, the probabilistic approach is still preferable to the systematic one as it allows us to guide the search algorithm according to custom conditions which may even evolve in time.

In Figure 2 we show the basic incremental probabilistic approach (IPS) to template generation. Procedure Decide-Consistency does the actual template validation. The stop criterion is met when either Decide-Consistency succeeded in validating the template or execution time expired. It may sometimes happen that the hypothesis created from the template contains some isolated atoms, i.e. atoms with variables that are not unified with other variables. These atoms are obviously redundant and hence they are removed from the final hypothesis (Remove-Isolated-Atoms procedure in the algorithm). A drawback of the whole algorithm is that it can negatively affect the performance of Decide-Consistency function as the function has to consider unifications between variables in atoms that are eventually identified as redundant.

-
- 1: $P \leftarrow$ all different predicate symbols in examples
 - 2: $T \leftarrow$ empty template
 - 3: **repeat**
 - 4: Generate predicate $p \in P$ with probability distributed uniformly
 - 5: $T \leftarrow T \cup \{p\}$
 - 6: $H \leftarrow$ Decide-Consistency(T)
 - 7: **until** stop criterion satisfied
 - 8: $H \leftarrow$ Remove-Isolated-Atoms(H)
-

Fig. 2. An incremental probabilistic template generation (IPS)

4 History-Driven Tabu Template Generation

The key drawback of the pure incremental approach as described in the previous section is that it generates new atoms without any respect to the previous work it did. In particular, we noticed that if the added atom contributes to “better” consistency of the hypothesis, it might be useful to add atom of the same predicate symbol again. We formalize this idea in another template generation algorithm based on two techniques:

- (1) For a given template, we check the maximal number B of broken negative examples achieved by some hypothesis generated from the template (hypothesis breaks the negative example if it does not subsume it and subsumes all positive examples at the same time). Then we check the value of B between two subsequent templates. Based on information whether B increases or decreases between the iterations we will decide about the next predicate to be added to the template.
- (2) We maintain a *tabu list* [4] of predicates such that these predicates cannot be added to the template in next steps of the algorithm.

Firstly, let us discuss in detail the point (1). After we launch the formerly described algorithms (either IDS or IPS), we analyze the maximal numbers of the negative examples which are broken by any hypothesis formed from the last two templates. In particular, if we added predicate p to the last template and the maximal number B of the broken negative examples increased in comparison to the number B of the last but one template, then we studied what happens with the maximal number of the broken negative examples for the next template if we add the same predicate p again. Empirical results showed that if a predicate added to the template increased the number of broken negative examples in one iteration then it is very likely to increase the number B again if we add the same predicate in the next iteration again. Particularly, in our experiments the average number of such iterations, where the number of broken negative examples was increased obeying the suggested rule, was 69.3% (the base was computed as the number of all subsequent pairs of templates, where the first template of the pair increased the maximal number of the broken negative examples). In contrast, we also averaged the number of iterations where the number of broken negative examples was increased by adding a different predicate than the one added last time. The number of such iterations was 26.19%, which is much less than the number of the improving iterations in the first case. Thus adding the same predicate seems to be beneficial. Finally we should note that the fact that in the remaining 30.7% of the first case when the rule did not increase the maximal number of the negative examples does not mean the heuristic went always wrong. It might also happen that there was no predicate at all that would increase the maximal number of the broken negative examples for a given template structure (no matter which one would be added).

Now let us continue with the point (2) that resembles the idea of Tabu Search [4] though there are some differences in handling the tabu list. If we add some predicate that did not cause the maximal number of broken negative examples to increase, then we forbid adding this predicate to the template in the following iterations of the algorithm until some condition (a so-called *aspiration criterion*) is met and the predicate is allowed to be used again. For this purpose we maintain a tabu list consisting of all predicates that cannot be added to the template in next iterations. As soon as the tabu list contains all predicates or some predicate increased the maximal number of broken negative examples, the tabu list is emptied and the process continues. In the second case, we empty the tabu list because of the fact that if the last predicate increased the maximal number of broken negative examples, then the template was changed relatively significantly and all predicates (also those in the tabu list) are now more likely to break some new negative examples than they were before.

Now if we put these two concepts together, we get a basic *history-driven tabu template generation algorithm* which is illustrated in Figure 3. In each iteration of the algorithm, we try to validate the current template (line 6) and if we succeed, we are done. Otherwise we store the maximal number of broken negative examples during the validation (line 9) and then we proceed with the next steps depending on the maximal number of broken negative examples in the previous iteration. If the number

increased, we add the same predicate to the template as we did the last time (line 11). In other case we generate a new predicate randomly such that it is not in the tabu list and we add it to the template (lines 16, 17).

```

1: Initialize template  $T$ 
2:  $B_{last} \leftarrow 0$  {max. number of broken neg. examples in the last iteration}
3:  $P_{last} \leftarrow$  any predicate {a predicate appended into template in the last iteration}
4:  $Tabu \leftarrow \emptyset$  {tabu list}
5: repeat
6:    $H_{consistent} \leftarrow$  Decide-Consistency( $T$ )
7:   if  $H_{consistent}$  exists then return  $H_{consistent}$ 
8:   else
9:      $B_{current} \leftarrow$  Max-Broken-Neg-Examples( $T$ )
10:    if  $B_{current} > B_{last}$  then
11:       $T \leftarrow T \cup \{P_{last}\}$ 
12:       $Tabu \leftarrow \emptyset$ 
13:    else
14:      if all predicates in  $Tabu$  then  $Tabu \leftarrow \emptyset$ 
15:       $P_{current} \leftarrow$  random predicate not in  $Tabu$ 
16:       $T \leftarrow T \cup \{P_{current}\}$ 
17:       $Tabu \leftarrow Tabu \cup \{P_{current}\}$ 
18:       $P_{last} = P_{current}$ 
19:    end if
20:     $B_{last} = B_{current}$ 
21:  end if
22: until stop criterion satisfied

```

Fig. 3. A history-driven template generation

The complexity of the above algorithm is strongly dependent on the complexity of Decide-Consistency function. It has been proved that the problem of deciding template consistency is Σ_2^P -complete [5], hence the whole algorithm belongs at least to this complexity class. If we want to determine the upper bound, it is necessary to identify the number of calls of Decide-Consistency function in the repeat loop. In general, the loop does not have to finish at all. However, we are usually not interested in finding arbitrary solution for the ILP problem and rather it makes sense to impose an upper bound on the size of the desired solution. Furthermore, if the bound is polynomially related to the size of the evidence, we get a so-called *bounded ILP problem* which is Σ_2^P -complete [5].

4.1 Stochastic Extension of History-Driven Tabu Template Generation

The algorithm in Figure 3 behaves like a modification of the well known hill-climbing algorithm and it is fully deterministic except for the step at line 15. We can further

improve its performance by modifying the probability to select a predicate at line 15. We developed a stochastic model as an extension of the basic history-driven algorithm yielding the *stochastic history-driven tabu algorithm*. The idea is that the predicate that “behaved well” in previous iterations is preferred to be added to the template.

First, besides the tabu list we introduce a *candidate list* which is the list of all predicates not in the tabu list. Furthermore, each predicate in the candidate list is assigned a probability such that all probabilities in the candidate list sum to one. In each iteration of the algorithm we pick a predicate from the candidate list according to its probability and add it to the template. The candidate list is a dynamic structure. In addition to adding and removing predicates into or from it, the probabilities may change as well. There are two situations when the probabilities in the candidate list are recomputed.

The first situation arises when the last predicate added to the template increased the maximal number of broken negative examples. In that case we (1) move all predicates from the tabu list back to the candidate list and set their probabilities to some low value p , (2) set the probability of the last added predicate to some high value p_{high} in the candidate list. After adding the predicates from the tabu list back to the candidate list, we should still prefer those predicates originally in the candidate list to those recently added from the tabu list when selecting the next predicate to be added to the template. The reason is that the predicates originally in the tabu list behaved “worse” than those in the candidate list (that is the reason why they were tabu). The intended effect can be achieved by modifying the probabilities of predicates in the candidate list in the following way. We find a predicate with the minimal probability p_{min} among all predicates in the candidate list and then we assign the probability $p = p_{min} \cdot p_{tabu}$ to all predicates moved from the tabu list. In our implementation we use $p_{tabu} = 0.1$ so the probability of selecting some predicate originally from the tabu list is at least ten times smaller than the probability of selecting any predicate originally in the candidate list. Setting the probability of the last successful predicate to p_{high} means that we prefer appending those literals that increased the maximal number of broken negative examples in the last iteration (as we have already proposed). In our algorithm we use $p_{high} = 0.95$. We should remark that after every change of the probabilities in the candidate list we normalize them to yield the sum equal to one.

The second situation when the probabilities of the predicates in the candidate list are adjusted is when the candidate list becomes empty (it means that all predicates are in the tabu list). In this case we put all predicates back to the candidate list and we distribute the probability among them uniformly. In Figure 4 we give a pseudo code for the stochastic history-driven tabu template generation. This algorithm is very similar to the algorithm in Figure 3 except it extends it with the stochastic steps. We use a few new procedures in the algorithm. Procedure Distribute-Probability-Uniformly returns the set of all predicates with identical probabilities. Procedure Predicate-With-Min-Prob returns the predicate with the minimal probability. Procedure Update-Probability takes the list provided in the first argument and updates the probability of the predicates in the second argument to the value in the third argument. Finally procedure Normalize-Probabilities adjusts the probabilities such that they sum to one. As the stop criterion we can use expiration timeout.

```

1: Initialize template  $T$ 
2:  $B_{last} \leftarrow 0$  {maximal number of broken negative examples in the last iteration}
3:  $P_{last} \leftarrow$  any predicate {a predicate appended to the template in the last iteration}
4:  $Tabu \leftarrow \emptyset$  {tabu list}
5:  $Cand \leftarrow$  Distribute-Probability-Uniformly {initialization of candidate list}
6: repeat
7:    $H_{consistent} \leftarrow$  Decide-Consistency( $T$ )
8:   if  $H_{consistent}$  exists then return  $H_{consistent}$ 
9:   else
10:     $B_{current} \leftarrow$  Max-Broken-Neg-Examples( $T$ )
11:    if  $B_{current} > B_{last}$  then
12:       $p_{min} =$  Predicate-With-Min-Prob( $Cand$ )
13:      Update-Probability( $Cand, \{P_{last}\}, p_{high}$ )
14:      Update-Probability( $Cand, Tabu, p_{min} \cdot p_{tabu}$ )
15:      Normalize-Probabilities( $Cand$ )
16:       $Tabu \leftarrow \emptyset$ 
17:    else
18:      if all predicates in  $Tabu$  then
19:         $Tabu \leftarrow \emptyset$ 
20:         $Cand \leftarrow$  Distribute-Probability-Uniformly
21:      end if
22:    end if
23:     $P_{current} \leftarrow$  predicate from  $Cand$  with corresponding probability
24:     $T \leftarrow T \cup \{P_{current}\}$ 
25:     $Tabu \leftarrow Tabu \cup \{P_{current}\}$ 
26:     $Cand \leftarrow Cand \setminus \{P_{current}\}$ 
27:    Normalize-Probabilities( $Cand$ )
28:     $B_{last} = B_{current}$ 
29:     $P_{last} = P_{current}$ 
30:  end if
31: until stop criterion satisfied

```

Fig. 4. A stochastic history-driven tabu template generation

5 Experimental Results

In order to compare effectiveness of the proposed methods, we implemented and benchmarked the algorithms in SICStus Prolog 4.1.2 on 2.0 GHz Intel Xeon processor with 12 GB RAM under Gentoo Linux. The first set of experiments was executed on ten instances of identifying common structures in random graphs generated according to Barabási-Réka model [1]. We used graphs consisting of 20 nodes that were constructed by incrementally adding new nodes and connecting them with three arcs

to existing nodes in the graph. The hidden structure that we were looking for (the consistent hypothesis) consisted of five nodes. Both positive and negative evidence contained ten instances of the graphs.

In Table 1 we show a comprehensive comparison of all methods described in the paper – the naïve iterative deepening search (IDS) from [2], the incremental probabilistic search (IPS) and finally the stochastic history-driven tabu search (SHDTS). For each method we show the overall running time and the length of the found template. The runtime limit was set to 600 seconds. Since IPS and SHDTS are randomized algorithms, the results were obtained by averaging values through their five runs. In case any of these runs exceeded the maximal time limit, the final values were averaged only from the successful runs and the number of these unfinished runs is showed in the corresponding column.

Table 1. A comparison of the naïve iterative deepening search (IDS), the incremental probabilistic search (IPS) and the stochastic history-driven tabu search (SHDTS)

IDS		IPS			SHDTS		
time[s]	length	time[s]	length	#unfinished	time[s]	length	#unfinished
1.75	6	0.33	6	-	0.37	6	-
13.97	7	2.34	7	1	1.85	7	-
13.92	7	1.31	7	1	0.62	7	-
0.27	5	0.11	5	1	1.17	7	-
455.43	8	334.51	8	-	273.75	8	-
10.93	7	1.16	7	1	1.02	7	-
>600	-	6.13	8	1	2.11	8	-
411.60	7	28.07	8	-	0.46	10	-
11.88	7	16.41	7	1	67.70	8	-
13.52	7	1.41	7	1	0.55	7	-

First let us compare the results between IDS and IPS. If we look at the hypotheses lengths¹, we can see that they are almost identical for both searches, thus IPS finds almost always the optimal solution (we know that the solution given by the IDS is optimal). Further let us compare the runtimes. Except for one dataset, IPS clearly outperforms IDS. On the other hand, we have to realize that these results are not guaranteed as IPS is a randomized algorithm. In fact, in seven of ten cases the algorithm did not finish in one of its five runs. The reason why the search algorithm ran so long is that the random generator generated repeatedly “wrong” predicates until the template was so long that its validation exceeded the time limit.

Now let us focus on the result of SHDTS. We can see that the length of the final hypothesis is almost identical to the optimal solution again and thus templates generated by SHDTS are appropriate. Actual significance of this method is evident when we analyze the runtimes. The method not only beats IPS for almost every case

¹ When comparing the lengths of the hypotheses, we only compare the number of all atoms in the hypothesis but we do not consider the discrepancies between actual predicate symbols of the atoms and their arities in the hypotheses.

but it also finished for all instances. The worse result in the ninth case is due to the randomized nature of the algorithms. On the other hand, IPS did not finish once on this instance. Hence the experiments proved that stochastic history-driven tabu search noticeably outperforms the former two methods.

To support further that the proposed algorithms contribute to better performance of whole ILP system, we evaluated them on another set of input instances generated according to different model than in the first case. In this case we were again identifying common structures in random graphs, however new nodes were connected to the existing ones obeying Erdös-Rényi [3] model. In particular, each input dataset consists of ten positive and ten negative examples (graphs), where the graphs contain 20 nodes with density of arcs 0.2. In Table 2 we present results for finding implanted hidden structure consisting of 5, 6 and 7 nodes with arc density of 0.4. In the first column there is a number of nodes of the hidden common structure and in other columns there are runtimes and hypothesis lengths of each algorithm. The overall runtime limit was set to 1200 seconds. From Table 2 it clearly follows that SHDTS is working best for most of the datasets.

Table 2. A comparison of the three methods for identifying common structures (subgraphs) of various number of nodes

nodes	IDS		IPS		SHDS	
	time[s]	length	time[s]	length	time[s]	length
5	1.41	6	0.74	6	0.74	6
5	>1200	-	204.11	9	59.65	9
5	>1200	-	5.23	9	17.7	9
5	>1200	-	518.47	10	448.72	10
6	>1200	-	533.25	9	241.57	9
6	>1200	-	313.25	9	211.78	9
6	>1200	-	426.70	10	366.41	10
6	>1200	-	181.28	9	215.01	9
7	>1200	-	716.15	10	950.04	10
7	27.71	7	>1200	-	3.98	7
7	>1200	-	>1200	-	>1200	-
7	>1200	-	201.28	10	164.61	10

6 Conclusions

In this paper we addressed the problem of generating templates for inductive logic programming. We presented a novel approach that uses existing CSP algorithms as its subroutines. Specifically we use subroutines for Θ -subsumption check [6] and for deciding template consistency [2]. We started with a simple algorithm from [2] performing iterative deepening search that is showed to be inefficient. The main result of the paper is a history-driven tabu template generating algorithm which is guided by evaluating how the negative evidence is covered by the generated template. Furthermore, we suggested a stochastic extension of this algorithm yielding a stochastic history-driven

tabu search. The efficiency of the last algorithm is demonstrated experimentally on the problem of identifying common structures in randomly generated graphs according to two different models. Stochastic history-driven search is doing better on almost all input instances and it significantly decreases the template generation time in comparison to former algorithms. An interesting feature of this algorithm is that it uses some form of learning so the algorithm learns itself during the learning process.

Future work should mainly deal with testing the performance of the algorithms on instances of domains other than those of the random graphs, for example those of the bioinformatics. Real-life problems have often a particular structure and thus it is challenging to examine our methods in these fields.

Acknowledgments. The authors would like to thank Filip Železný and Ondřej Kuželka for useful discussions on ILP techniques and for providing a generator of random problems for experiments.

References

1. Barabási, A.-L., Réka, A.: Emergence of scaling in random networks. *Science* 286, 509–512 (1999)
2. Barták, R.: Constraint Models for Reasoning on Unification in Inductive Logic Programming. In: Dicheva, D., Dochev, D. (eds.) *AIMSA 2010. LNCS (LNAI)*, vol. 6304, pp. 101–110. Springer, Heidelberg (2010)
3. Erdős, P., Rényi, A.: On Random Graphs I. *Publicationes Mathematicae Debrecen* 6, 290–297 (1959)
4. Glover, F., Laguna, M.: *Tabu Search*. Kluwer, Norwell (1997)
5. Gottlob, G., Leone, N., Scarcello, F.: On the complexity of some inductive logic programming problems. *New Generation Computing* 17, 53–75 (1999)
6. Maloberti, J., Sebag, M.: Fast Theta-Subsumption with Constraint Satisfaction Algorithms. *Machine Learning* 55, 137–174 (2004)
7. Mitzenmacher, M., Upfal, E.: *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press (2005)
8. Plotkin, G.: A note on inductive generalization. In: Meltzer, B., Michie, D. (eds.) *Machine Intelligence*, vol. 5, pp. 153–163. Edinburgh University Press (1970)

Towards Building a Masquerade Detection Method Based on User File System Navigation

Benito Camiña, Raúl Monroy, Luis A. Trejo, and Erika Sánchez

Computer Science Department

Tecnológico de Monterrey — Campus Estado de México

Carretera al Lago de Guadalupe Km. 3-5, Atizapán, Estado de México, 52926, México

{a00965049,raulm,ltrejo,snora}@itesm.mx

Abstract. Given that information is an extremely valuable asset, it is vital to timely detect whether one’s computer (session) is being illegally seized by a masquerader. Masquerade detection has been actively studied for more than a decade, especially after the seminal work of Schonlau’s group, who suggested that, to profile a user, one should model the history of the commands she would enter into a UNIX session. Schonlau’s group have yielded a masquerade dataset, which has been the standard for comparing masquerade detection methods. However, the performance of these methods is not conclusive, and, as a result, research on masquerade detection has resorted to other sources of information for profiling user behaviour. In this paper, we show how to build an accurate user profile by looking into how the user structures her own file system and how she navigates such structure. While preliminary, our results are encouraging and suggest a number of ways in which new methods can be constructed.

1 Introduction

For today’s world, where information is reckoned to be an extremely valuable asset, it is vital to timely detect whether one’s computer (session) is being illegally seized by some intruder, so-called a *masquerader*. A masquerader aims to put a computer holder into jeopardy, as quickly as possible, while impersonating the holder to avoid being caught.

Masquerade detection has been actively studied since the seminal work of Schonlau *et al.* [1], who suggested that, in order to profile a user, it should suffice to model the history of the (parameterless) commands she would type while logged in into a UNIX session. For that purpose, Schonlau *et al.* developed a masquerade dataset, commonly referred to as *SEA* [2], which has been a *de facto* standard for building, validating and comparing a number of masquerade detection methods (see, for example, [3,4,5,6]).

The performance of SEA-based methods, however, cannot be said to be overwhelming. Accordingly, in later experiments, SEA has been enriched to consider additional information; in particular, command arguments (see, for instance, [7]). Also as a result, research on masquerade detection has turned to using alternative sources of activity in an attempt to better characterise user behavior. Example

alternative information sources include device usage (*e.g.* the mouse [8], or the well-studied keyboard [9]), and application usage (*e.g.* the use of a document management system [10]). In a similar vein, user activity has been clustered into activity types, such as information gathering, browsing, communications, etc. [19]. These activity types, 22 in total, aim at capturing user intention, and they have been characterized in terms of executions of Windows applications, dynamic link libraries, and Microsoft Disk Operating System commands.

In this paper, we claim that to better characterise user behaviour, it is necessary to consider how and what the user browses while working on her own file system. After all, a user file system is an abstract representation of some user, at least as far as the operating system is concerned. Crucial to our approach is a structure, we call a user *navigation structure*, representing the navigation of a user through her file system. A navigation structure contains information of a user file system. In particular, it contains the most recently visited file system objects; it also contains information about both how a user structures her own file system, and how she uses and browses such directory.

We will show that the aim of using a user navigation structure is twofold. First, a navigation structure allows to better understand the behaviour of a given user. Second, and more importantly, it allows to build a user profile suitable for masquerade detection. While we reckon that a combination of various kinds of user activities will be necessary, at the end of the day, for practical masquerade detection, our results show that file system navigation is central to achieve an accurate user profile.

While preliminary, our results are very promising. We shall see that even Naïve Bayes, when applied as an intrusion detection mechanism on the file system objects visited by a user, surpasses the performance of similar methods, that have been designed to consider alternate sources of information. We shall argue that the information gathered in navigation structures can make this classifier take more informed intrusion detection decisions.

Overview of Paper. The remainder of this paper is organised as follows. First, §2, we shall show the limitations of UNIX-commands based masquerade detection methods. In particular, we shall argue that, on the one hand, they perform rather poorly, even when they are not subject to proper masquerader action; and that, on the other hand, they are difficult to implement, because the activity information chosen to profile a user is, *per se*, sparse. Second, §3, we will (operationally) introduce user navigation structures and the operations they take. Third, §4, we shall briefly outline the datasets we have collected throughout our experiments, both (honest) user and masquerader, and shall provide preliminary results on understanding and profiling user behaviour for masquerader detection, using the frequency of access to file system objects. Then, in §5, we shall discuss directions for further research, which aims to truly exploit the information contained in navigation structure. Finally, in §6, we will report on the conclusions we have drawn from our experiments.

2 Masquerade Detection

Masquerade detection is concerned with timely noticing that an intruder is impersonating a legitimate user in a computer session. [1]. It is usually undertaken as an anomaly detection task, where the masquerade detection mechanism aims at distinguishing any diversion in the current user activity from a given profile of ordinary user behaviour. Masquerade detection has been actively studied since the seminal work of Schonlau *et al.* [1], who have developed a database, called *SEA* [2], which is the *de facto* standard for building, validating and comparing masquerade detection mechanisms.

2.1 The SEA Dataset

SEA contains log information about the activity of 70 UNIX users. Each user log consists of 15,000 commands, that were gathered via the `aact` auditing mechanism, stripping off the arguments. 50 users were randomly selected to serve as honest holders, while the remaining ones were (artificially) set to act as masqueraders.

The *SEA* dataset is structured as follows. Every command sequence of a legitimate user is first divided into blocks of size 100. Each command block is called a *session*; thus, user activity is arbitrarily taken to be composed of, and represented by, 150 sessions. The first 50 sessions of every user are left untouched; they constitute ordinary user behavior, we call *the user history of commands*. The command history set is used as the *construction dataset* for any proposed masquerade detection system. The last 100 sessions, however, may or may not have been contaminated, and they are used as the detection mechanism *validation dataset*.

If contaminated, a validation session is taken to be a masquerade. A validation session is either totally contaminated or not. Session contamination in *SEA* amounts to replacing the selected, original validation session for one of a user who was arbitrarily set to be a masquerader. Contaminated sessions were inserted using this rule: if no masquerader is present, then a new masquerader appears in the next session with a probability of 1%. Otherwise, the same masquerader continues to be present in the next block with a probability of 80%. *SEA* comes with a matrix that, for each user, shows where masquerade sessions are located, if any.

As a test dataset, *SEA* is very restrictive, when more realistic conditions are considered. To begin with, a supposedly masquerader does not have any knowledge about the profile of an intended victim. A masquerade session is just a sequence of commands that a user, unwillingly marked as a ‘masquerader’, would type in a UNIX session. Moreover, in *SEA*, some of such masqueraders show very simple and repetitive behaviour, which is easily spotted as unusual, even by human inspection.

To overcome these limitations, one may embark oneself in gathering faithful UNIX sessions; however, such task has proven to be complex, given that real UNIX sessions are sparse and, hence, difficult to obtain. This has been shown

by Chinchani *et al.* [11], who have attempted to synthesise sequences of user commands, with the aim of enabling the construction of a model for ordinary behaviour. RACOON, Chinchani *et al.*'s tool, synthesises user sessions in order to get around of the inherently long time it takes the collection of real ones. Its ultimate aim is to speed up the process of development and evaluation of masquerade detection systems.

Recently, research on masquerade detection has been looking at alternative sources of user activity to better profile user behavior. Example alternative sources of information include user device usage (*e.g.* the mouse [8]), and user interaction with specific applications (*e.g.* the use of a document management system [10].) There is large archive of research on keystroke dynamics [9]; however, though exhaustive, it makes very unrealistic assumptions, for user activity is not recorded under normal, working conditions, but under very artificial ones.

In a complementary vein, research on masquerade detection has attempted to characterize user activity considering a combination of several observable actions. For example, [19] provides 22 types of user activities, including information gathering, browsing, communications, etc. These activity types are defined in terms of executions of Windows applications, dynamic link libraries, and Microsoft Disk Operating System commands. Thus, the masquerade detection mechanism is ad-hoc, in that it is Operating System dependent, and needs to be adapted whenever a change on platform is in order.

2.2 Masquerade Detection Mechanisms: An Overview

Research on masquerade detection has been very prolific. Space constraints preclude us from giving a reasonable overview, or a fair comparison of these mechanisms (even if restrained to the most significant ones). We shall hence confine ourselves to overview a few of them, referring the reader to [12,13] for a survey.

A masquerade detector is *global*, if, for determining masquerader presence, it uses both the profile of the user being protected and the profile of that user's colleagues. Otherwise, it is *local*. Similarly, a masquerade detector is *temporal-based*, if, for user profile formation, it considers both individual actions and the relations thereof; *e.g.* action sequences. Otherwise, it is *frequency-based*. Since global, temporal methods are more informed, they are usually more accurate than local, frequency-based ones; however, they demand more information and computing effort, and are inapplicable in some contexts.

Uniqueness [14] is a global, frequency-based detection method, which stems from the observation that the appearance of a command not seen in the construction of a user profile may indicate the presence of a masquerader. A command is said to be of *popularity i* if only *i* users use that command. It is said to be *unique* if it is of popularity one. The detection model consists of a score, computed on each test session, which increases (respectively decreases) upon the appearance of a unique (respectively unseen) command. Uniqueness has been an obliged reference for comparison purposes.

Customized grammars [5] is global, and temporal-based. It profiles ordinary user behavior using the unique and most repetitive sequences of user actions.

To identify such sequences of actions amongst a user history and across other users' audit logs, customized grammars applies *sequitur* [15], a method for inferring compositional hierarchies from strings. A test session is classified using an evaluation function, which yields a high score if the session includes very long action sequences with a high self, *i.e.* a high frequency in the user-history) and low unself (*i.e.* a low frequency in the all-users-history).

Naïve Bayes [16,7,17] assumes that the actions of a user are independent one another. Thus, the probability for an action c to have been originated from a user u , $\Pr_u(c)$, is given by $\frac{f_{u,c} + \alpha}{n_u + \alpha \times K}$, where $f_{u,c}$ is the number of times user u issued c in the user log of actions, n_u the length of u 's log, K is the total number of distinct actions, and where $0 < \alpha \ll 1$ prevents $\Pr_u(c)$ from becoming zero. Thus, naïve Bayes is frequency-based. It might be either local or global, with similar performance [17]. To evaluate a test session s , in which user u has allegedly participated, the cumulative probability of s is compared either against the probability that s has been output by somebody else [16] or against a user-defined threshold [17].

Sequitur-then-HMM [6] is a local, temporal masquerader detector, which characterises user behavior in terms of the temporal dependencies among action sequences of most frequent occurrence. This method first applies sequitur to a given user history of actions. Then, the main production rule of the grammar output by sequitur is used to train a Hidden Markov Model (HMM). Consequently, not only does this model consider temporal dependencies among actions, by virtue of sequitur, but it also considers temporal dependencies among sequences, by virtue of the HMM.

The performance of these (and other) methods, however, is far from conclusive. This can be seen in Figure 1, borrowed from [18] (there TRD stands for the method of [6]), which reports on that most of these methods, when operating at a 95% detection rate, produce over 40% of false alarms. Masquerader detectors have not been fully or faithfully tested. This is because they have not been put to work under stressing conditions, where it is required to distinguish an intruder who actually is aiming at effective user impersonation (see also [18]). From this, it follows that these methods have not been thoroughly compared either.

3 Navigation Structures

To capture how a user uses and browses her file system, we use a navigation structure. A navigation structure contains information about both how a user structures her own file system, and how she uses it and browses it. Objects in a navigation structure have been visited by the user, either directly, from a directory browser, or indirectly, from a user application. A navigation structure contains only user objects, ignoring system- or application-related files.

A navigation structure is composed of two graphs: an access graph and a directory graph. An *access graph* contains a record of the most recently visited file system objects, as well as the order (actually, a traversal path) which the user has followed to visit them. A *directory graph* is a proper subgraph of the

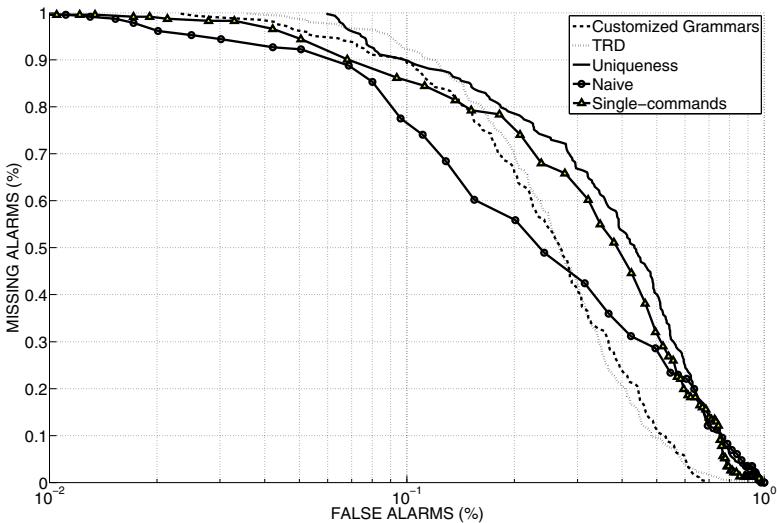


Fig. 1. Performance of detection methods with attack diffusion

user file system; it is indeed an arborescence, with a distinguished vertex, called the *root*, and denoted by “/”.

3.1 Access Graph

Let \mathcal{FS} denote the file system of a given user u .¹ The access graph for u is a directed graph, given by the pair $G = (V, E)$, where $V \subset \mathcal{FS}$ collects all the nodes (files or folders) that u has recently visited, and E records the node visiting history. Arcs in E are annotated: $n \xrightarrow{k} n' \in E$ if and only if, having visited n , u has visited n' next, k times.

Initially, $G = (V, E)$ is such that $E = \emptyset$, and V is a singleton, containing the node representing the root folder. Then, $G = (V, E)$ is updated, upon access to node n' , having visited node n , as follows:

$$(V, E) = \begin{cases} (V, (E - \{n \xrightarrow{k} n'\}) \cup \{n \xrightarrow{k+1} n'\}) & \text{if } n \xrightarrow{k} n' \in E \\ (V \cup \{n'\}, E \cup \{n \xrightarrow{1} n'\}) & \text{otherwise} \end{cases}$$

Every node $n \in V$ is a tuple with three elements: i) the path associated to the file system object identified with n , denoted $\text{path}(n)$, ii) the position of node n in graph D (see below, §3.2), and iii) the weight of the node. As expected, for every $n \in V$, $\text{path}(n)$ is a string of the form “ $(\alpha/)^*\alpha$ ”, where “ α ” stands for

¹ Given that u is understood from the context, we omit the subscript u in all these symbols, for the sake of clarity.

an alphanumeric, representing the name of the file system object, and \star for the transitive, reflexive closure of string concatenation, denoted by juxtaposition.

The weight of a node n , denoted $\text{weight}(n)$, represents the number of times n has been accessed and it is defined as follows:

$$\text{weight}(n) = \text{foldl}(\{k \mid \exists n'. n' \xrightarrow{k} n\}, +, 0)$$

where $\{\dots\}$ denotes a multiset, and where foldl is a higher-order function, given by $\text{foldl}(\{X_0, X_1, \dots, X_n\}, F, E) = F(\dots(F(F(E, X_0), X_1), \dots) X_n)$, and where E , and F , respectively, denote the base element, and the step function.

3.2 Directory Graph

A directory graph is a directed acyclic graph representing a subtree graph of the user file system. Let \mathcal{FS} denote the file system of a given user, u , with access graph $G = (V, E)$. Then, the directory graph for u is a pair $D = (V', E')$, where $V \subseteq V'$ and E' denotes the usual descendant relation of a tree. For every node $n \in V$, there are as many nodes in V' , one for each file system object along $\text{path}(n)$. Arcs in E are ordered pairs: $(n, n') \in E'$ if and only if n' is a descendant of n , in the user file system structure.

Like an access graph, a directory graph, $D = (V', E')$, is initially such that $E' = \emptyset$, and V' is a singleton containing the root node. Then, upon any access to an object o , we first use $\text{path}(o)$ to compute the pairs that potentially are to be inserted to update D as follows. Let t be a singleton string. Then,

$$\begin{aligned} \pi_o = \{(n, n') \mid \exists \alpha, \alpha', t. \text{path}(o) = \alpha t \alpha', \\ n, n' \text{ are created such that } \text{path}(n) = \alpha, \text{ path}(n') = \alpha t\} \end{aligned}$$

where we assume pairs can be freshly created as required. Next, for every pair $(n, n') \in \pi_o$, we update $D = (V', E')$ as follows:

$$(V', E') = \begin{cases} (V' \cup \{n'\}, E' \cup (n, n')) & \text{if } (n, n') \notin E' \\ (V', E') & \text{otherwise} \end{cases}$$

Every node $n \in V'$ is annotated with accounting information. Formally, it is a tuple of three elements: i) the path associated to the file system object identified with n , ii) the time the object was last accessed, and iii) the position of node n , with respect to D , denoted $\text{pos}(n, D, \text{path}(n))$. Given that D is of the form: $/(\alpha_1, \dots, \alpha_k)$; i.e. it has top node / and k object children, we define node position by:

$$\text{pos}(n, /(\alpha_1, \dots, \alpha_k), \text{path}(n)) = \begin{cases} [] & \text{if } \text{path}(n) = / \\ [i] & \text{if } \text{path}(n) = / \alpha_i \\ [i] \text{ pos}(n', \alpha_i, \text{path}(n')) & \text{if } \text{path}(n) = / \alpha_i \text{ path}(n') \end{cases}$$

Navigation Structure Maintenance To keep the navigation structure of a given user to a manageable size, we have readily applied Least Recently Used, a page replacement algorithm for virtual memory management. Other algorithms, such as FIFO or Most Frequently Used are both applicable and easy to implement, while others, *e.g.* Second Chance, or Not Used Recently require us to include more accounting information inside a navigation node. We update a navigation structure whenever required; for example, after the user has moved or removed a file system object. We shall have more to say, later on in text, see §5, about how to exploit a navigation structure in the study of the masquerader detection problem.

4 Some Preliminary Results

4.1 Construction- and Validation-Masquerade Datasets

Datasets. During the experimental phase, we invited six people, three to act as ordinary users (see Table 1), and the rest as masqueraders (see Table 2). On each user machine, we ran a process that, in the background, built the corresponding user navigation structure. Profile construction was, according to users, completely unnoticeable, and was carried out during normal working days. By way of comparison, the activities of masqueraders were tracked, while they were allowed a free five-minute navigation on the machine of each legitimate user; on each attempt, masqueraders aimed to compromise the victim user, as much as possible.

Table 1. User profiles

User	Profile	Logging		Log size	Directory
		Starting date	Ending date		
1	Assistant: invoice and purchase orders procedures	12/04/2010	01/24/2011	3697	Low
2	Employee: Logistics and process documentation	12/17/2010	01/24/2011	2338	Medium
3	Manager: Finance and accounting	01/03/2011	01/24/2011	11494	High

Users and intruders, as can be noticed from Tables 1 and 2, are of different nature. We selected them this way driven towards validating our working hypotheses, namely: that there are users who are easy to impersonate, and that, conversely, there are intruders who are readily detectable.

Naïve Bayes. As a classification model, we used Naïve Bayes, using access frequencies of objects (see Section 2.2). Hence, information about the behavior of the user under protection is needed. In our case, an action c , issued by user, u , is interpreted as user u has accessed file system object, c . Hence, the cumulative probability of a session s of length n is given by: $\text{Pr}_u(s) = \text{Pr}_u(c_1) \times \dots \times \text{Pr}_u(c_n)$. To evaluate a test session s , in which user u has allegedly participated, the cumulative probability of s has been compared against a user-defined threshold. If the probability is below this threshold, the session is considered normal, and, in that case, the user profile is updated. Otherwise, user u is considered a masquerader.

Table 2. Intruder profiles

Intruder	Profile	Intruder log size generated against		
		user 1	user 2	user 3
1	PhD student, highly skilled in information security	969	444	1076
2	Lawyer with basic knowledge of computer usage	239	224	230
3	MSc student using specialised tools to obtain sensitive data	107	366	146

4.2 Experimentation Results

Figure 2 summarizes, via a ROC curve, our results of experimentally testing Naïve Bayes against all three masqueraders. Our results are very promising, suggesting that file system navigation is a good means for masquerade detection. Two conclusions follow. First, it is more difficult to detect a masquerade attempt against a user who carries out a few, specific tasks, or who organizes badly his directory structure; this might be explained by that the access probability distribution associated to a user of this kind is near the uniform. For example, masquerade attempts against user 3 were readily detected (0% false negatives and 1.75% false positives); by contrast, attempts against user 1 were difficult to spot (10.96% false negatives and 20.16% false positives). Second, it is more difficult to detect an occasional or novice intruder; for example, our worst results were obtained on masquerade attempts of intruder 2 against user 1. This actually is good news, because, using our approach, masquerading becomes a challenge for a highly skilled intruder; this is because he must behave as the legitimate user, while compromising sensitive data (at first sight, a contradictory goal.)

5 Discussion

Although the above results are promising, we reckon that taking full advantage of a navigation structure will improve the performance of masquerade detection mechanisms. In this section, we lay out a few ideas which might be helpful for further research.

Task and Distribution Probability of Task Transition.

People tend to collect similar or related objects into one or more common folders. Using this observation, we conjecture that whenever a user is carrying out a specific task, she will visit objects of a few folders, and that these folders will all be closed one another too. Then, we may define a single, unique folder to be considered as an abstract representation of a task, and call that folder simply a *task*.

In what follows, we aim to formalise this notion of task. Let $G = (V, E)$ be the access graph of a given user u . Let $n, n' \in V$. Then, $n \ll n'$ if and only if there exists a nonempty substring α , such that $\text{pos}(n') = \text{pos}(n)\alpha$. Let $\aleph(S)$ stand for the cardinality of set S . Then, the load of n , denoted $\text{load}(n)$, is given by $\text{load}(n) = \aleph(\{t' | (t \xrightarrow{k} t') \in E \text{ and } n \ll t'\})$. Now, a node n is said to be a *task of u* if and only if $\text{load}(n) \neq 0$ and, except, possibly, from the root node, $/$, there does not exist a node t' , such that $t' \ll n$, and that $(t \xrightarrow{k} t') \in E$.

Using this notion of task, t , together with a relation of task activity, which consists of visiting nodes underneath t , we have, experimentally, proven that the principle of *spatial* locality of reference holds for the behaviour of a user u . Figure 3 depicts spatial locality for user 3; there, we have conducted a simple normalisation of the position of the accessed object for the coordinate; this has been done so that two objects that, relative to their positions, are close one another, they they will also get close coordinate values. Similar experiments need to be conducted towards establishing temporal locality of reference.

Now, we could apply the notion of task to define a Markov chain in terms of a *task transition probability*. This Markov model could be used to compute the probability of reaching a particular task, having departed from some other, not necessarily different task, after a few accesses. This probability could be applied in a model for masquerade detection. Of course, a penalisation would have to be considered, if there is an access to a node that does not belong to any fixed task. Other Bayesian models, such as Hidden Markov Models, could be applied to obtain a masquerade detection method.

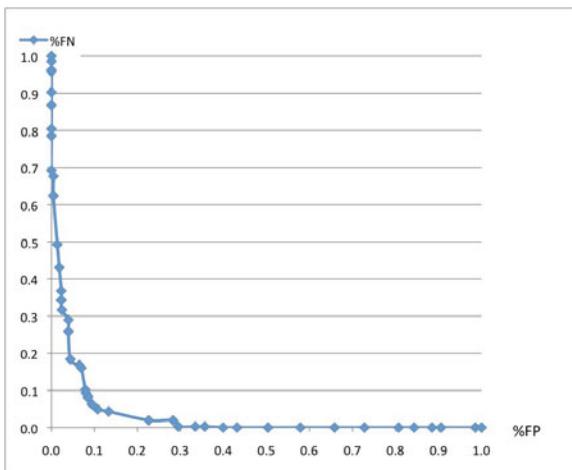


Fig. 2. ROC curve: masquerade detection results

Task Fault Rate. Actually, an access to a node that is not in the access graph could be regarded as a *task fault*. Then, the rate at which task faults occur could

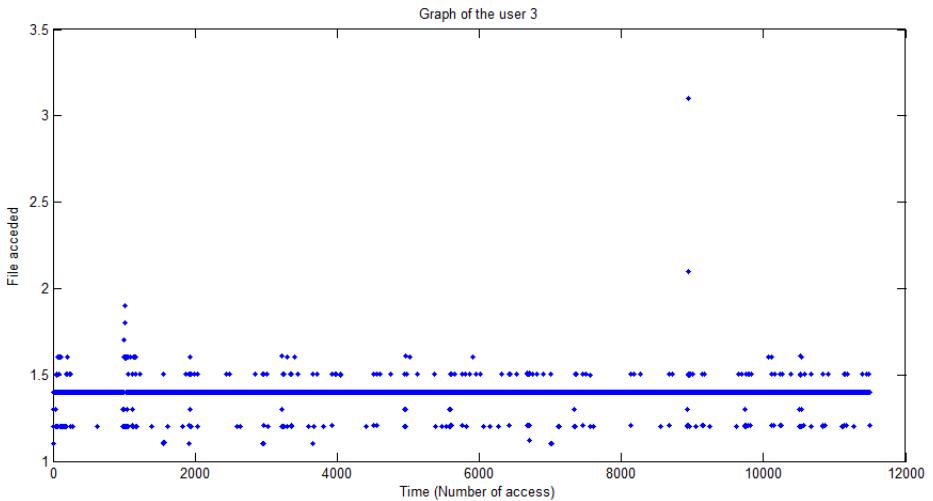


Fig. 3. Spatial Locality of reference for user 3

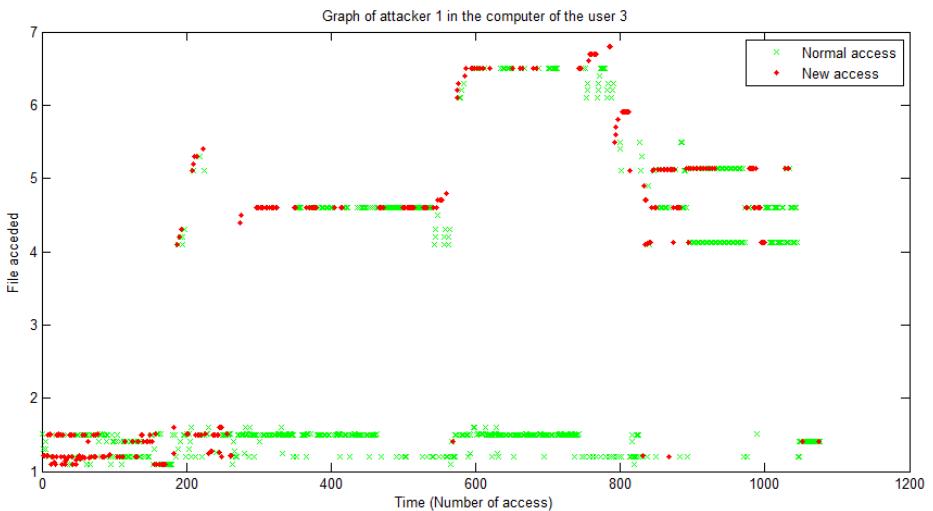


Fig. 4. Object creation rate when intruder 1 is masquerading user 3

also be applied in a model for masquerade detection. Figure 4 depicts how task faults occur when intruder 1 is trying to masquerade user 3. We have found that, under attack, task faults occur much more frequently than under normal conditions.

Depth in File System Navigation. Other measure that can be exploited to masquerade detection is the depth of node n , given by $\text{depth}(n) = \text{length}(\text{pos}(n)) + 1$.

Then, we may try to model the behaviour of a user in terms of the depth at which she usually navigates through her file system. The lower and the upper quantiles associated to a box plot could, for example, be directly used to masquerade detection.

6 Conclusions

Towards the characterisation of file-system user navigation, we have proposed here a navigation structure. In a preliminary experiment, we have built a masquerader detection mechanism, based on naïve Bayes, applying one output of file system navigation, namely: the frequency of visited objects. Our results are very promising, showing a 1.75% false positive ratio and 0% false negative ratio, for a particular user, for the best case, and 79.84% of detection rate, at a cost of a 10.96% of false positive detection rate, for the worst case.

Given people diversity, a user profile can be more suitable than others to detect some form of intrusion. On the one hand, our results show that a very tidy and highly structured user will be much easier to separate from an actual masquerader. By way of comparison, a user who carries out a very restricted collection of tasks may yield an object access probability distribution close to the uniform, and, hence, she might be very easy to masquerade. On the other hand, a highly skilled masquerader is more likely to obtain sensitive information in a short period of time. However, this behaviour might be much easier to spot, given the number of previously unseen visited objects he will raise to during an intrusion. Thus, a good profile, based on a navigation structure, makes a masquerader of this sort to become more easily detected. An occasional masquerader, however, will be harder to spot, due to his errant, and hesitant behaviour.

There exists a number of methods with which we can build a masquerade detection mechanism. In this paper, we have explored one method only. Further research is needed in order to explore the additional ideas laid out in this paper. Although a combination of various kinds of user activities will be necessary, at the end of the day, for practical masquerade detection, our results show that file system navigation is central to achieve an accurate user profile.

References

1. Schonlau, M., DuMouchel, W., Ju, W., Karr, A., Theus, M., Vardi, Y.: Computer intrusion: Detecting masquerades. *Statistical Science* 16, 58–74 (2001)
2. Schonlau, M.: Masquerading user data (2008), <http://www.schonlau.net>
3. Maxion, R.A., Townsend, T.N.: Masquerade detection augmented with error analysis. *IEEE Transactions on Reliability* 53, 124–147 (2004)
4. Oka, M., Oyama, Y., Abe, H., Kato, K.: Anomaly Detection using Layered Networks based on Eigen Co-occurrence Matrix. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 223–237. Springer, Heidelberg (2004)
5. Latendresse, M.: Masquerade Detection via Customized Grammars. In: Julisch, K., Kruegel, C. (eds.) DIMVA 2005. LNCS, vol. 3548, pp. 141–159. Springer, Heidelberg (2005)

6. Posadas, R., Mex-Perera, C., Monroy, R., Nolazco-Flores, J.: Hybrid Method for Detecting Masqueraders using Session Folding and Hidden Markov Models. In: Gelbukh, A., Reyes-Garcia, C.A. (eds.) MICAI 2006. LNCS (LNAI), vol. 4293, pp. 622–631. Springer, Heidelberg (2006)
7. Maxion, R.A.: Masquerade detection using enriched command lines. In: Proceedings of the International Conference on Dependable Systems and Networks, DSN 2003, pp. 5–14. IEEE Computer Society Press, San Francisco (2003)
8. Garg, A., Rahalkar, R., Upadhyaya, S., Kwiat, K.: Profiling users in GUI based systems masquerade detection. In: Proceedings of the 7th IEEE Information Assurance Workshop, pp. 48–54. IEEE Computer Society Press (2006)
9. Killourhy, K.S., Maxion, R.A.: Why did my detector do *that*?! - Predicting Keystroke-Dynamics Error Rates. In: Jha, S., Sommer, R., Kreibich, C. (eds.) RAID 2010. LNCS, vol. 6307, pp. 256–276. Springer, Heidelberg (2010)
10. Sankaranarayanan, V., Pramanik, S., Upadhyaya, S.: Detecting masquerading users in a document management system. In: Proceedings of the IEEE International Conference on Communications, ICC 2006, vol. 5, pp. 2296–2301. IEEE Computer Society Press (2006)
11. Chinchani, R., Muthukrishnan, A., Chandrasekaran, M., Upadhyaya, S.: RACOON: Rapidly generating user command data for anomaly detection from customizable templates. In: Proceedings of the 20th Annual Computer Security Applications Conference, ACSAC 2004, pp. 189–204. IEEE Computer Society Press (2004)
12. Salem, M.B., Hershkop, S., Stolfo, S.J.: A survey of insider attack detection research. In: Stolfo, S.J., Bellovin, S.M., Hershkop, S., Keromytis, A., Sinclair, S., Smith, S.W. (eds.) Insider Attack and Cyber Security: Beyond the Hacker. Advances in Information Security, pp. 69–90. Springer, Heidelberg (2008)
13. Bertacchini, M., Fierens, P.: A survey on masquerader detection approaches. In: Proceedings of V Congreso Iberoamericano de Seguridad Informática, Universidad de la República de Uruguay, pp. 46–60 (2008)
14. Schonlau, M., Theus, M.: Detecting masquerades in intrusion detection based on unpopular commands. Information Processing Letters 76, 33–38 (2000)
15. Nevill-Manning, C.G., Witten, I.H.: Identifying hierarchical structure in sequences: a linear-time algorithm. Journal of Artificial Intelligence Research, JAIR 7, 67–82 (1997)
16. Maxion, R.A., Townsend, T.N.: Masquerade detection using truncated command lines. In: Proceedings of the International Conference on Dependable Systems & Networks, pp. 219–228. IEEE Computer Society Press, Washington, DC (2002)
17. Wang, K., Stolfo, S.: One-class training for masquerade detection. In: Proceedings of the 3rd IEEE Conference Data Mining Workshop on Data Mining for Computer Security. IEEE (2003)
18. Razo-Zapata, I., Mex-Perera, C., Monroy, R.: Masquerade attacks based on user's profile. Journal of Systems and Software (2011) (submitted for evaluation)
19. Ben-Salem, M., Stolfo.: Modeling user search behavior for masquerade detection. Computer Science Technical Reports 033, Columbia University (2010)

A Fast SVM Training Algorithm Based on a Decision Tree Data Filter

Jair Cervantes¹, Asdrúbal López², Farid García³, and Adrián Trueba¹

¹ Autonomous University of Mexico State, UAEM-Texcoco, México

² Center of Research and Advanced Studies-IPN, Instituto Politécnico Nacional 2508, CP 07360, México DF

³ Autonomous University of Hidalgo State, CP 43800, Tizayuca-Hidalgo, México
`chazarra17@gmail.com, achau@computacion.cs.cinvestav.mx,`
`fglamont@yahoo.com.mx`

Abstract. In this paper we present a new algorithm to speed up the training time of Support Vector Machines (SVM). SVM has some important properties like solid mathematical background and a better generalization capability than other machines like for example neural networks. On the other hand, the major drawback of SVM occurs in its training phase, which is computationally expensive and highly dependent on the size of input data set. The proposed algorithm uses a data filter to reduce the input data set to train a SVM. The data filter is based on an induction tree which effectively reduces the training data set for SVM, producing a very fast and high accuracy algorithm. According to the results, the algorithm produces results in a faster way than existing SVM implementations (SMO, LIBSVM and Simple-SVM) with similar accurateness.

1 Introduction

Support Vector Machines were introduced by Vapnik [1] as a kernel based machine learning model for classification and also for regression task. Due to its extraordinary generalization performance on a wide variety of problems, optimal solution and discriminative power, SVM has received considerable attention and it has been used as a powerful tool for solving practical binary classification problems. On the other hand, the major drawback of SVM occurs in its training phase [2], i.e., the amount of memory and training time are considerably bigger than other algorithms [3].

Some SVM training algorithms have been proposed in order to tackle the problem of speed up the training of SVM, most of them use the techniques of preselecting support vector (SV) candidates [3] [4] or decomposition [2][5]. For example in [6] [7] [8] a Convex Hull approach is used to reduce the training data set, converting this reduction into a nearest point problem. Platt [2] proposed Sequential Minimal Optimization (SMO) in order to reduce the training time of SVM. SMO is a simple and efficient algorithm for solving the generally huge quadratic programming problem (QPP) associated with SVM. The idea behind

SMO is to iteratively subdivide the QPP into a series of smaller QP problems, until the number of them becomes a two variables QPP. Finally this problem is solved analytically without using a numeric library. Experimentally, the SMO performs well for SVMs with sparse inputs and outperforms methods that utilize “chunking”. Platt [2] asserts that SMO is a strong candidate for becoming the standard SVM training algorithm [9].

Even for SMO, training support vector machine is still a bottleneck for large training data sets. When data sets contain tens of thousands of elements, it is important to develop new fast training algorithms for SVM. According to the architecture of SVM, the training data near of the boundaries are the most important and the data far away from the separating hyperplane have no contribution on SVM training [3] [10]. Taking advantage of this fact, the training time of SVM can be reduced if the data far from the separating hyperplane is removed by a preprocessing step before training. Moreover, there exists evidence [11] that in some cases SVM tends to perform worse with training from the entire data than training from a fine quality of samples of the data set.

In this paper we present a novel SVM algorithm which is very efficient with large data sets, the proposed approach firstly uses a data filter based on decision trees, in order to eliminate non-essential data and then the algorithm recover the data points that are near of boundaries. SVM training is done on those remained “useful” data. The proposed method exploits the discriminative power of SVM algorithm and decision trees to speed up the training time. Computational comparison on benchmark data sets, shows that these modifications outperform significantly faster than the original SMO and other fast SVM implementations.

2 Preliminaries

This section presents some basic concepts of the SVM and decision trees. A more detailed explanation can be found in [12] [13] respectively.

2.1 Support Vector Machines

Considering binary classification case, we assume that a training set X is given as:

$$((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)). \quad (1)$$

In other words, $X = \{x_i, y_i\}_{i=1}^n$ where $x_i \in R^d$ and $y_i \in (+1, -1)$

The optimal separating hyperplane is given by:

$$y = sign[w^T \varphi(x_i) + b]. \quad (2)$$

Which is obtained by solving the following QPP:

$$\begin{aligned} \min_{w,b} J(w) &= w^T / 2 + c \sum_{i=1}^n \xi_i \\ \text{such that: } y_i [w^T \varphi(x_i) + b] &\geq 1 - \xi_i. \end{aligned} \quad (3)$$

Where ξ_i are slack variables used to tolerate misclassifications $\xi_i > 0, i = 1, 2, \dots, n, c > 0$. Equation (3) is equivalent to the QPP (4) which is a dual problem with the Lagrange Multipliers $\alpha_i > 0$,

$$\begin{aligned} \max_{\alpha_i} J(w) &= -\frac{1}{2} \sum_{i,j=1}^n \alpha_i y_i \alpha_j y_j K \langle x_i \cdot x_j \rangle + \sum_{i=1}^n \alpha_i \\ \text{such that: } &\sum_{i=1}^n \alpha_i y_i = 0, C \geq \alpha_i \geq 0, i = 1, 2, \dots, n. \end{aligned} \quad (4)$$

With $C > 0, \alpha_i = [\alpha_1, \alpha_2, \dots, \alpha_n]^T, \alpha_i \geq 0, i = 1, 2, \dots, n$, are the coefficients corresponding to x_i , all the x_i with nonzero α_i are called SV. The function K is the kernel which must satisfy the Mercer Condition [12]. The resulting optimal decision function is defined as:

$$y_i = \text{sign} \left(\sum_{i=1}^n \alpha_i y_i K \langle x_i \cdot x_j \rangle + b \right). \quad (5)$$

Where $x = [x_1, x_2, \dots, x_n]$ is the input data, α_i and y_i are Lagrange multipliers. A previously not seen sample x can be classified using (5). There is a Lagrangian multiplier α for each training point. When the maximum margin of the hyperplane is found, only the closed points to the hyperplane satisfy $\alpha > 0$. These points are called support vectors (SV), the other points satisfy $\alpha = 0$, so the solution vector is sparse. Where b is determined by Kuhn-Tucker conditions:

$$\begin{aligned} \frac{\partial L}{\partial w} &= 0, w = \sum_{(i=1)}^n \alpha_i y_i \varphi(x_i) \\ \frac{\partial L}{\partial b} &= 0, \sum_{(i=1)}^n \alpha_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} &= 0, \alpha_i - c \geq 0 \\ \alpha_i \{y_i[w^T \varphi(x_i) + b] - 1 + \xi_i\} &= 0. \end{aligned} \quad (6)$$

2.2 Decision Trees

Decision tree techniques have become one of the most popular tools for classification and regression. One of the attractiveness of decision trees is that they can extract knowledge by inferring human understandable rules from data. Typically, a decision tree algorithm begins with the entire set of data, splits the data into two or more subsets based on the values of attributes and then repeatedly splits each subset into finer subsets until the split size reaches an appropriate level. The entire modeling process can be represented in a tree structure and the model generated can be summarized as a set of “if-then” rules. Decision trees are easy to interpret, computationally inexpensive, and capable of work with noisy data.

We used the C4.5 algorithm [13] to construct the decision tree data filter. The selection of the best attribute node is based on the gain ratio (S,A) where S is a set of records and A is an attribute. This gain defines the expected reduction in entropy due to sorting on A. It is calculated as:

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{\text{Value}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v). \quad (7)$$

In general, if we are given a probability distribution $P = (p_1, p_2, \dots, p_n)$ then the information conveyed by this distribution, which is called Entropy of P is given by:

$$\text{Entropy}(P) = - \sum_{(i=1)}^n p_i \log_2 p_i. \quad (8)$$

3 Implementing a Data Filter Algorithm

The data filter that we present here, implements a heuristic searching method to obtain the relevant data points from the whole data set. The proposed algorithm introduces an optimization step to quickly remove most of the nonsupport vectors and to obtain the most important data points in the all data set (SV candidates). The data filter algorithm proposed has the following characteristics.

- It applies SVM on a subset of training data set in order to obtain a sketch of the SV, and then obtains a reduced data set filtering data points that are far from sketch of the SV.
- It uses a decision tree in order to classify data points that are near of SV and then filters less important data points from the original data set. Since the performance of data filter depends directly of the small data set selected, is very important to implement an algorithm to select a small data set that include representative data points from the entire data set. The data filter algorithm captures the best distribution patterns of the data and provides enough information for obtain a good accuracy classification.

The data filter of the algorithm proposed captures the statistical summaries of the entire data set. It provides enough statistical information in order to obtain an excellent generalization performance.

3.1 Algorithm Description

The data filter algorithm is constructed using SVM and decision trees. The sketch of the data filter algorithm is the following:

1. Selecting a small data set: At this stage, our algorithm extracts a small portion C of the entire data set (EDS), in order to construct a hyperplane in a later phase of the process. To choose the elements of C , the number of elements with positive (l^+) and negative labels (l^-) in EDS is computed first, and then the data points are easily selected in $O(n)$ considering the following three criteria.

Criterion I. If the amounts of both positive and negative labels are similar, i.e., the proportion between the number of positive and the number of negative labels is small ($\pm 5\%$), the algorithm uses a simple random sampling where each element of the data set has an equal probability of selection,

Criterion II. If the proportion between positive and negative labels is bigger than 5% and smaller than 99%, the algorithm uses a inverse probability proportional to size sampling, e.g., if there are 90% data points with negative labels and 10% data points with positive labels, the random selection algorithm selects 90% of the small data set with positive labels,

Criterion III. Finally, if the proportion between data points with positive and negative labels is bigger than 99%, the algorithm uses all data points that have label with fewer rates.

This random selection approach can improve the accuracy for a given sample size by concentrating sampling on few elements in this stage, but it has the greatest impact on determining a hyperplane. Formally, our training data set EDS consists of N pairs $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, with $x_i \in R^p$, and $y_i \in \{-1, 1\}$. After the applying the heuristic selection the obtained data set is $C = \{x_i, y_i\}_{i=1}^l$, with $l \ll n$.

2. Identifying Data Points: After previous step has finished, a small data set C has been selected from EDS in linear time. An sketch separating hyperplane is then computed using only this small set C . This hyperplane becomes important because it is used to obtain the features of SVs and non-SVs and it is used also to reduce the original input dataset.

After the SVM classification stage, the hyperplane obtained is given by:

$$\sum_{k \in V_{small}} y_k \alpha_{small}^* K(x_k, x) + b_{small}^* = 0 \quad (9)$$

Where V_{small} are the SV, i.e., the data points that contain the most important features in the small data set.

SVs are the most important data points of all input data set and generally constitute a small fraction of the training examples. If most non-SVs can be removed from the entire data set, then the SVM training can be accelerated drastically. However, we need identify the features of SVs in the small data set, in order to recover all SVs in the entire data set and remove data points that are non-SVs. The algorithm takes the SVs obtained and labels them with an extra class $E^{(+1)}$, and non-SVs as outliers and labels them with a extra class $E^{(-1)}$. The outliers are observations that do not follow the statistical distribution of the bulk of the data, in this case are data points far from hyperplane and are not important when obtaining a classification hyperplane.

Summarizing, the small data set is given by $C = \{x_i, y_i\}_{i=1}^l$ with $l \ll n$, the SVs in the small data set are given by $\{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\}$, where $k = 1, 2, \dots, p$, and $k \in V_{small}$ and non SVs are given by:

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}, 1, 2, \dots, m \notin V_{small}.$$

It is clear that $\{x_l, y_l\} = \{x_k, y_k\} \cup \{x_m, y_m\}$, consequently, data set $\{x_i, y_i\}_{i=1}^k \in V_{small}$ is labeled with a extra class $E^{(+1)}$ and $\{x_i, y_i\}_{i=1}^m \notin V_{small}$ with $E^{(-1)}$.

3. Recovering Best Data Points: From the SV identified (x_k, y_k) and outliers obtained (x_m, y_m) in the previous stage, we can construct and train a decision tree in order to get data points with the best properties of entire data set.

The decision tree constructed is a binary classifier which detects all data that are close to data points $(x_k, y_k) \in V_{small}$ and removes any data that is similar to outliers $\{x_i, y_i\} \notin V_{small}$ in the entire input data.

In order to obtain all data points close to the first decision hyperplane, at this stage we train a decision tree classifier with SVs and non-SVs, the new distribution class is expressed by:

$$(x_1, y_{E_1}), (x_2, y_{E_2}), \dots, (x_n, y_{E_n}). \quad (10)$$

In other words, training data set $X = \{x_i, y_{E_i}\}_{i=1}^q$ where $= k + m$, $x_i \in R^d$, and $y_{E_i} \in (+1)$ and the non SV are expressed as $y_{E_i} \in (-1)$. Where $y_{E_i} \in (+1)$ constitute data points that contain the most important features and $y_{ai} \in (-1)$ constitute the data points far from hyperplane (outliers).

To find a decision function that separate SV's from non SV's we train a decision tree classifier, from it we obtain a decision function which separates the two classes in the new distribution (data points close from hyperplane and data points far from hyperplane). Finally the decision function is obtained by the decision tree classifier. The reduced dataset RDS can be expressed as:

$$RDS = \{x_{pi}, y_{pi}\} \cup (\{x_i, y_i\}_{i=1}^k \in V_{small}). \quad (11)$$

Where x_{pi} are the data close from the decision hyperplane obtained in the first stage, which have a positive label, and V_{small} are the support vector from small data set.

The data filter proposed is a method for preprocessing large training data sets. It obtains small subsets very fast and with high quality. The obtained subsets carry the statistical summaries of the data and maximize the benefit of learning the SVM. The entire data set is reduced to only a small data set with the most important data points, the data filter proposed provides an efficient and effective data set for SVM to run.

4 SVM Training Based on Data Filter

In this Section, we describe the SVM-BDF algorithm for training SVM on large data sets. The algorithm uses the data filter algorithm presented in last section to considerably reduce the size of the entire data set, without degrading the accuracy of final classifying hyperplane.

Data filter algorithm obtains data points which maximize the benefit of learning using SVM and decision trees in order to obtain the SVs candidates and to eliminate outliers.

In Algorithm 1 SDS is sampling data set, SV support vectors, nSV non-support vectors, Change SDS is change of class for SDS and DT decision tree.

```

Input: Entire data set  $EDS = \{x_i, y_i\}_{i=1}^n$ 
Output: Reduced data set
 $RDS = \{x_i, y_i\}_{i=1}^l, l \ll n$ 
1: SDS=getDataSampling(EDS)
2: hyper=train.SVM(SDS)
3: SV=getSV(hyper,SDS)
4: nSV=getnSV(hyper,SDS)
5: changeSDS=getChangeClass(SV,nSV)
6: DT=train.DT(ChangeSDS,SV,nSV)
7: RDS=testing+.DT(EDS)
8: Outliers=testing-.DT(EDS)
9: Return RDS

```

Algorithm 1. Data Filter Algorithm for Reduce Input Data Set

By solving the problem stated in (4), the hyperplane is obtained, and this is given by

$$\sum_{k \in V_{reduced}} y_k \alpha_{reduced,k}^* K(x_k, x) + b_{reduced}^* = 0 \quad (12)$$

Where $V_{reduced}$ are data points with the most important features in the input data set. The hyperplane (12) es very similar to (9), however the accuracy obtained with final hyperplane is better than the accuracy with the first hyperplane.

The difference between the hyperplane obtained with a small data set and the reduced data set sometimes is very meaningful and depend directly of the data set. However, in both cases α_{small}^* and $\alpha_{reduced}^*$ converge to some suboptimal solution.

```

Input: Reduced data set RDS
Output: Sub optimal hyperplane Subhyper

1: Subhyper=train.SVM(RDS)
2: Return Subhyper

```

Algorithm 2. Training of SVM

The recovery data are the most important data from optimization point of view. Therefore, the hyperplane (12) is an approximation of hyperplane obtained with SVM using SMO with the original data set.

5 Experimental Results and Comparison

In order to show the performance of our algorithm, we tested it with six public datasets available on the web and two synthetic ones created by ourselves.

Organization of the experimental results is as follows. In the first part the characteristics of the eight data sets used are discussed briefly. The second part of this section shows the performance of the proposed algorithm in terms of generalization capability and training time is compared with two other state of the art support vector training algorithms.

5.1 Data Sets

The following datasets are used in the paper for doing several experiments: Checkerboard and UCI Adult data sets from UCI Machine learning repository¹; IJCNN1, Covtype Binary and Magic data sets from LibSVM²; ARN³ and finally Synthetic 1 and Synthetic 2 were designed by us.

The main characteristics of each data set (training size, number of features and class of each data set) are shown in the Table 1. The two synthetic data sets have two classes for purpose of testing binary classification. The Synthetic 1 data set is linearly separable whereas Synthetic 2 data set is linearly non-separable.

Table 1. Data Sets Used

Data set	Training size	Features	Class
Synthetic 1	109,000	2	2
Synthetic 2	500,000	2	2
UCI Adult	30956	123	2
IJCNN1	49990	22	2
Covtype Binary	581,012	54	2
ARN	23605	8	2
Magic	19020	10	2
Checkerboard	1,000,000	2	2

5.2 Clasification Accuracy and Training Time

The SVM-BDF algorithm was programmed in C++. Simple SVM (SSVM), SMO and LIBSVM also implemented in the same programming language were the algorithms used to compare the performance and accuracy of our algorithm. These algorithms were selected because are the most representatives in the area of SVM and have demonstrated good performance in practical applications.

For all data sets, except Synthetic 1, the RBF kernel function was used. Regularization constant C from {10, 100, 1000, 10000} was chosen by optimization of the test set for all experiments. The experiments were performed on a 2.1GHz Intel Core2 Duo CPU, 2 GB RAM machine running Windows XP. The Synthetic 1 is the only linearly separable data set used in the experiments. For this data set we use linear kernel.

¹ <http://archive.ics.uci.edu/ml/>

² <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

³ <http://www.ncbi.nlm.nih.gov>

Table 2 shows a comparative between algorithms, t is the training time in seconds, classification accuracy on test set is Acc. The test set has size 10 percent of that of the entire data set, and contains points that do not belong to the (90 percent) training set. In the case of UCI Adult, IJCNN1 and Magic data sets, there exist test data sets which contain data points that do not belong to the training data set. Means and standard deviations (SD) over 10 independent runs are reported for UCI adult and binary Covtype data sets.

Table 2. Comparative Between Algorithms

Data set	SMO SSVM		LIBSVM	SVM-BDF		
	t	Acc	t	Acc	t	Acc
Synth 1	12700	100	120	100	351	100
Synth 2	72700	99.9	223.5	99.9	861	100
IJCNN1	27032	98.5	20491	98.2	461	98.5
ARN	8309	93.5	3081	92.7	409	93.5
Magic	5314	88.1	7385	88.3	213	88.1
Checker	196000	99.8	22371	99.8	95400	99.85
					1315	99.85

It can be seen in Table 2 that SVM-BDF require several orders less training time as compared to others SVM Implementations, while providing comparable classification accuracies. Among them, the proposed approach achieves the highest or second highest classification score in the least time for all the data sets. The superiority of SVM-BDF becomes clearer in huge datasets as can be seen in Table 2. In the case of Chekerboard, Synthetic 1 and Synthetic 2 data sets, the SSVM algorithm requires substantially less time compared to Libsvm. However, SSVM is not suitable to data sets where dimension of input data is large. SSVM is very fast in data sets where the number of features is small due to nature of the algorithm.

It is observed from the results shown in Table 2 that Libsvm algorithm performs better than SSVM for IJCNN1, ARN and Magic data sets. However the proposed algorithm requires substantially less time compared to SMO, SSVM and Libsvm algorithms. In all data sets showed in the Table 3 the number of SV represents only a fraction of the entire training set. The number of SV obtained with the proposed technique is similar to that obtained with classical techniques of SVM, if the algorithm uses a smaller data set, however, the most important data recovered in the total data set, recovering many support vectors not considered in the initial stage. However, in UCI Adult and Covtype Binary many data points are support vectors. Preliminary studies show that over one hundred thousands of support vectors are required for a good approximation.

The SVM-BDF algorithm uses a data filter algorithm which uses the best data points from a small data set in order to select the best data points in the entire data set. This step may often provoke improvement of the generalization performance, maintaining a fast convergence.

Table 3. Comparative for two data sets

Data Set	Algorithm Acc		T	SV's
	Mean	SD		
UCI	SSVM 84.87	0.17	737	7310
Adult	LibSVM 84.85	0.1021	83	11077
	SVM-BDF 84.92	0.098	57	9735
	SSVM —	—	—	—
CovType	LibSVM 80.42	0.67	37419	317921
	SVM-BDF 80.35	0.47	10915	189038

6 Conclusions and Future Work

In this paper we presented SVM-BDF, a new algorithm for training SVM for classification. The proposed algorithm works very fast even with large data sets and outperforms the existing state of the art SVM implementations without substantial reduction of accuracy. SVM-BDF applies a data filter based on a decision tree that scans the entire data and obtains a small subset of data points. The proposed approach is conceptually simple, easy to implement, and for some experiments faster than other SVM training algorithms (SMO, LIBSVM and Simple-SVM) because it avoids calculating margins for nonsupport vector examples. The superiority of our algorithm is experimentally demonstrated for some real life data sets in terms training time. The results of experiments on synthetic and real data sets show that the proposed approach is scalable for very large data sets while generating high classification accuracy. We are now working in adapt SVM-BDF ideas to stream data mining SDM, a relatively new research area in data mining. The challenge in SDM for SVM rests in that the underlying model must consider concept drift phenomena and SV must be updated very quickly.

References

1. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
2. Platt, J.C.: Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In: Scholkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernel Methods-Support Vector Learning, pp. 185–208. MIT Press (1998)
3. Cervantes, J., Li, X., Yu, W., Li, K.: Support vector machine classification for large data sets via minimum enclosing ball clustering. Neurocomputing 71(4-6), 611–619 (2008)
4. Yu, H., Yang, J., Han, J.: Classifying Large Data Sets Using SVM with Hierarchical Clusters. In: Proc. of the ACM SIGKDD Intl. Conf. on Knowledge, pp. 306–315 (2003)
5. Dong, J., Krzyzak, A., Suen, C.Y.: Fast SVM Training Algorithm with Decomposition on Very Large Data Sets. IEEE Trans. Pattern Anal. Mach. Intell. 27(4), 603–618 (2005)

6. López, J., Barbero, Á., Dorronsoro, J.R.: Simple Clipping Algorithms for Reduced Convex Hull SVM Training. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS (LNAI), vol. 5271, pp. 369–377. Springer, Heidelberg (2008)
7. Wu, C., Wang, X., Bai, D., Zhang, H.: Fast SVM incremental learning based on the convex hulls algorithm. In: Proc. of the Intl. Conf. on Computational Intelligence and Security, pp. 249–252. IEEE Computer Society (2008)
8. Mavroforakis, M.E., Sdralis, M., Heodoridis, S.: A geometric nearest point algorithm for the efficient solution of the SVM classification task. *IEEE Trans. Neural Networks* 18, 1545–1549 (2007)
9. Bottou, L., Lin, C.J.: Support Vector Machine Solvers. In: Bottou, L., Chapelle, O., DeCoste, D., Weston, J. (eds.) Large Scale Kernel Machines, pp. 1–28. MIT Press (2007)
10. Chambers, R.L., Skinner, C.J. (eds.): Analysis of Survey Data. Wiley (2003)
11. Schohn, G., Cohn, D.: Less is more: Active learning with support vector machines. In: Proc. Intl. Conf. on Machine Learning, pp. 839–846 (2000)
12. Burges, C.J.: A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.* 2(2), 121–167 (1998)
13. Quinlan, J.R.: Improved use of continuous attributes in c4.5. *J. Artificial Intell. Res.* 4, 77–90 (1996)

Optimal Shortening of Covering Arrays

Oscar Carrizales-Turrubiates, Nelson Rangel-Valdez, and José Torres-Jiménez

CINVESTAV-Tamaulipas, Information Technology Laboratory,

Victoria Tamps., Mexico

{ocarrizales,nrangel}@tamps.cinvestav.mx, jtj@cinvestav.mx

Abstract. A Covering Array (CA), denoted by $CA(N; t, k, v)$, is a matrix of size $N \times k$ with entries from the set $\{0, 1, 2, \dots, v-1\}$, where in each submatrix of size $N \times t$ appears each combination of symbols derived from v^t , at least once. The Covering Arrays (CAs) are combinatorial structures that have applications in software testing. This paper defines the Problem of Optimal Shortening of Covering ARrays (OSCAR), gives its NP-Completeness proof and presents an exact and a greedy algorithms to solve it. The OSCAR problem is an optimization problem that for a given matrix M consists in finding a submatrix M' that is close to be a CA. An algorithm that solves the OSCAR problem has application as an initialization function of a metaheuristic algorithm that constructs CAs. Our algorithms were tested on a benchmark formed by 20 instances of the OSCAR problem, derived from 12 CAs taken from the scientific literature. From the solutions of the 20 instances of the OSCAR problem, 12 were transformed into CAs through a previously reported metaheuristic algorithm for the construction of CAs.

1 Introduction

According to Cohen et al. [5] the test phase represents between 20% and 50% of the total cost of the software development. In the process of software development the test phase is critical to guarantee the quality of the software. For a software component the set of all the possible configurations is usually too large to be tested exhaustively e.g. if we have a software component that has 10 parameters and every parameter takes 5 values, we need to make $5^{10} = 9765625$ tests; hence, the use of an exhaustive approach to accomplish the test phase requires many resources since it has to make a lot of tests, this makes the exhaustive approach impractical. An alternative approach to accomplish the test phase is called software interaction testing, this approach is based on the construction of test suites of economical size (number of test cases) that provide the coverage of all the combination of certain size of parameter values. This approach is based on the idea that a lot of errors are produced by the interaction of few parameter values.

Kuhn et al. [14] explain the advantage of interaction testing on the test phase by showing the efficient error detection in several software components.

This paper defines the Problem of Optimal Shortening of Covering ARrays (OSCAR). The solution of the OSCAR problem has application in the construction of CAs. We show that the solution of the OSCAR problem is an excellent initial solution for metaheuristic algorithms that construct CAs. In addition, we present a proof of the NP-completeness of the OSCAR problem, the design of exact and greedy algorithms that solve the OSCAR problem, and empirical results of the performance of these approaches when solving the problem.

The present document is organized as follows: Section 2 presents the notation followed in the document and related work about the construction of CAs and initial matrices for metaheuristic algorithms that construct CAs; this section also shows the proof that the OSCAR problem is NP-complete through a reduction of Maximum Coverage Problem (MAXCOVER) to it; Section 3 details the exact and greedy algorithms proposed in this paper; Section 4 presents the benchmark and the experimental results obtained from the comparison of our approaches and the construction of new CAs; finally, Section 5 shows the conclusions derived from the research presented in this paper.

2 Problem Definition

This section presents the notation followed in the document. Also, it formally defines the OSCAR problem and presents a proof of its NP-Completeness.

In order to define the OSCAR problem it is necessary to know the formal definition of Covering Array (CA). A CA, denoted by $CA(N; t, k, v)$ is a matrix M of size $N \times k$ which takes values from the set of symbols $\{0, 1, 2, \dots, v - 1\}$ (called the *alphabet*), and every submatrix of size $N \times t$ contains each tuple of symbols of size t (or t -tuple), at least once. The value N is the number of rows of M , k is the number of parameters, where each parameter can take v values and the interaction degree between parameters is described by the strength t . Each combination of t columns must cover all the v^t t -tuples. Given that there are $\binom{k}{t}$ sets of t columns in M , the total number of t -tuples in M must be $v^t \binom{k}{t}$. When a t -tuple is missing in a specific set of t columns we will refer to it as a missing t -wise combination.

Table 1 shows an example of a $CA(4; 2, 3, 2)$. The matrix has $N = 4$ rows and $k = 3$ columns. The alphabet $v = 2$ is formed by the symbols $\{0, 1\}$. The t -tuples are $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$, it is so because the interaction level is $t = 2$. The combination of t columns are $\{\{0, 1\}, \{0, 2\}, \{1, 2\}\}$. Given that the matrix is a CA, there is no missing t -wise combinations.

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Fig. 1. $CA(4; 2, 3, 2)$

When a matrix has the minimum possible value of N to be a $CA(N; t, k, v)$, the value N is known as the Covering Array Number. The Covering Array Number is formally defined as $CAN(t, k, v) = \min \{ N | \exists CA(N; t, k, v) \}$.

Given the values of t , k , and v , the optimal CA construction problem (CAC) consists in constructing a $CA(N; t, k, v)$ such that the value of N is minimized.

The OSCAR problem can be formally defined as follows. Let $\mathcal{A} = CA(N; t, k, v)$ be a CA or a quasi-CA¹, $\tau(\alpha)$ a function that indicates the number of missing t -wise combinations of a matrix α , δ , and Δ two integer values s. t. $(0 \leq \delta \leq N - v^t) \wedge (0 \leq \Delta \leq k - t)$, and at least one of them is greater than zero, the OSCAR problem consists in finding a submatrix \mathcal{B} of size $N - \delta \times k - \Delta$ of \mathcal{A} such that $\tau(\mathcal{B})$ is minimized.

Figure 2 shows an instance of the OSCAR problem and its solution. In this instance the inputs are a matrix \mathcal{A} that is a $CA(24; 2, 12, 4)$, $\delta = 4$ and $\Delta = 8$. The solution \mathcal{B} for this instance is shown in **Figure 2**, it was formed by the rows $\{0, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 19, 20, 21, 22, 23\}$ and the columns $\{3, 8, 10, 11\}$ taken from \mathcal{A} ; this matrix has zero missing t -wise combinations, therefore it is a CA.

2.1 Related Work

The CAC problem has an efficient solution only for some special cases, e.g. when the alphabet is binary and the strength is two [21] or the alphabet is a power of a prime number and $k \leq v + 1$ [2]. Another direct method to solve CAC when $k = t + 1$ is called zero-sum [8]. In [16] it is stated that the CAC problem when $t = 2$ is NP-Complete, however the proof of this is erroneous [15]. Though has not been proved that the CAC problem belongs to the class of NP-Complete problems, some related problems like the decision problem that consists in determining if it is possible to extend a matrix by one row and cover at least m missing t -wise combinations are NP-Complete [7,15].

Different approximated methods are used to solve the CAC problem. Among these methods are: a) Recursive methods [21,13,17]; b) Algebraic methods [3,9]; c) Greedy methods [5,23,1,12,6]; and d) Metaheuristic methods, such as tabu search [18], simulated annealing [22,4] and genetic algorithms [20]. The approximated algorithms construct non-optimal CAs but in a reasonable amount of time.

According to the related literature, there are two initialization functions that are commonly used by metaheuristic algorithms that construct $CA(N; t, k, v)$. The first one consists in generating randomly the initial matrix, i.e. each cell of a matrix $M_{N \times k}$ will take a random generated value from the set $\{0, 1, \dots, v - 1\}$. This initialization function has been used in different works such as [18,4,20]. The other initialization function creates a matrix M of size $N \times k$ where each

¹ This matrix has a small number of missing t -wise combinations compared to all the t -wise combinations to be covered.

1 2 2 2 3 1 0 0 3 3 2 1	1 2 3 2
2 1 1 2 3 3 3 3 0 0 1 2	2 1 0 2
3 2 2 1 2 0 1 3 0 1 0 0	0 2 1 3
0 2 3 3 1 3 1 0 1 3 2 0	3 3 3 1
2 2 0 1 3 0 1 1 3 0 3 3	0 1 3 0
2 2 1 0 2 2 3 0 3 3 1 0	2 0 3 3
0 1 1 2 1 0 2 0 3 2 3 3	3 2 0 3
2 0 1 3 1 1 3 2 3 1 0 2	1 1 1 3
1 1 2 3 2 2 0 3 0 0 2 3	2 2 2 0
2 3 3 3 2 0 0 2 1 0 1 1	2 3 1 0
1 0 0 0 2 0 3 1 2 3 2 2	3 1 2 0
0 0 3 0 3 2 1 2 1 1 3 2	1 3 0 0
3 1 0 0 1 1 2 3 2 0 1 3	3 0 1 2
1 3 2 0 1 3 0 1 0 2 3 1	0 0 0 0
3 0 0 2 2 3 2 2 1 1 0 3	1 1 1 1
0 0 2 1 1 2 3 3 1 2 1 1	3 3 3 3
0 0 0 0 0 0 0 0 0 0 0 0	2 2 2 1
1 1 1 1 1 0 1 1 1 1 1 1	0 0 0 1
2 2 2 2 0 2 2 2 2 2 2 2	1 0 2 3
3 3 3 3 0 3 3 3 3 3 3 3	0 3 2 2
3 3 1 1 0 3 0 0 2 1 2 2	
1 1 3 1 3 1 2 2 0 3 0 0	
3 3 0 3 3 2 2 1 2 2 0 1	
0 3 3 2 2 1 1 1 2 2 3 0	

Fig. 2. An instance of the OSCAR problem and its solution. a) The matrix $\mathcal{A} = CA(24; 2, 12, 4)$. b) The matrix \mathcal{B} that is the solution of the OSCAR problem for matrix \mathcal{A} , $\delta = 4$ and $\Delta = 8$. The matrix \mathcal{B} has value $\tau(\mathcal{B}) = 0$, which means that it is a $CA(20; 2, 4, 4)$.

column k_i , for $1 \leq i \leq k$ has a balanced number of symbols in it (i.e. if the alphabet is v there are approximately $\frac{N}{v}$ zeros, $\frac{N}{v}$ ones, etc.). Torres-Jimenez et al. [19] use matrices created this way to initialize the solution of a simulated annealing approach for the CAC problem.

2.2 The OSCAR Problem Is NP-Complete

This subsection has the purpose of showing that the OSCAR problem belongs to the class of NP-complete problems. In order to do that we will show the following about the OSCAR problem: 1) it is in NP; 2) there is an NP-complete problem \mathcal{R} that can be reduced to it in polynomial time; and 3) there is a solution for \mathcal{R} if and only if there is a solution for the equivalent instance of the OSCAR problem.

a) Step 1: The OSCAR problem belongs to the class of NP problems

First, let's consider an alternative representation of an instance of the OSCAR problem. In Section 2 an instance of the OSCAR problem is represented by a matrix $A = CA(N; t, k, v)$ and the values δ, Δ . The new representation will use a matrix \mathcal{A}' with N rows and $\binom{k}{t}$ columns. Each column j of \mathcal{A}' makes reference to a t -column of the CA that is constructed. A cell $a'_{i,j} \in \mathcal{A}'$ will contain the t -tuple that the row i cover in the t -column j .

Table 1 shows an example of the input matrix of an instance of the OSCAR problem and its new representation. **Table 1a** shows the matrix $\mathcal{A} = CA(4; 2, 3, 2)$, the values δ, Δ for this instance are 1, 0 respectively. **Table 1b** presents the different t -columns of \mathcal{A} and which t -tuples must be contained in each of them. **Table 1c** contains the instance of the OSCAR problem in the new representation, a matrix \mathcal{A}' of size 4×3 ; this matrix contains the t -tuple covered by each row in the different t -columns.

Table 1. An instance of the OSCAR problem and its new representation

	c_1	c_2	c_3	t-tuple	t-column		t_1	t_2	t_3
r_1	0	0	0	0)	(0, 0)	$t_1 = (c_1, c_2)$	r_1	0	0
r_2	0	1	1	1)	(0, 1)	$t_2 = (c_1, c_3)$	r_2	1	1
r_3	1	0	1	2)	(1, 0)	$t_3 = (c_2, c_3)$	r_3	2	3
r_4	1	1	0	3)	(1, 1)		r_4	3	2
r_5	1	1	1				r_5	3	3

(a) \mathcal{A}

(b)

(c) \mathcal{A}'

We proceed with the proof of the NP-completeness of the OSCAR problem by showing that the special case of the problem where $\Delta = 0$, and we call an instance of this kind as OSCAR¹. A solution for this special case of the OSCAR problem can be any subset of $N - \delta$ rows of the initial matrix \mathcal{M} of size $N \times k$. Hence, the solution space potentially has $\binom{N}{N-\delta}$ solutions. The problem is clearly in NP because the number of missing t -wise combinations can be counted directly from the matrix \mathcal{A}' in time $O(N \times \binom{k}{t})$, where $N \times \binom{k}{t}$ is the size of the matrix.

a) Step 2: An instance of the MAXCOVER problem can be reduced to an instance of the OSCAR problem in polynomial time.

Now, in order to prove that the special case of the OSCAR problem (when $\Delta = 0$) is NP-complete, we proceed to show the reduction of the NP-Complete problem of Maximum Cover [11] (MAXCOVER) to it. Particularly, we will show a reduction from MAXCOVER to the new representation of the OSCAR problem.

An instance of MAXCOVER is defined by an integer K , the set $\mathcal{U} = \{u_1, u_2, \dots, u_l\}$ and the set $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$, where $S_i \subseteq \mathcal{U}$ for all $1 \leq i \leq m$. The goal of the problem is to find a subset of \mathcal{S} of size K such that the union of its elements cover the maximum number of elements of \mathcal{U} .

Next we show the relation between the parameters of the MAXCOVER and an instance of the OSCAR problem when $\Delta = 0$. The set \mathcal{U} indicates the t -wise combinations of the matrix \mathcal{A} , the set \mathcal{S} has N elements where each S_i contains the t -wise combinations covered by each row i , and the value K is $N - \delta$.

An instance of MAXCOVER $\mathcal{U}, \mathcal{S}, K$ can be reduced to the OSCAR problem by forming the matrix \mathcal{A}' . The matrix \mathcal{A}' will have $m + 1$ rows; the first m rows

will be associated with the subsets of $S_i \subseteq S$. The number of columns will be $l + \max\{|S_i|\} + 1$. The first l columns will make reference to the elements $u_j \in \mathcal{U}$. The t -tuples to be covered in each column will be 0 and 1. A row i will cover the t -tuple 1 if any of the following cases happens: a) if its associated subset S_i covers the element $u_j \in \mathcal{U}$; or b) if $i > m$ and $j > l$; otherwise, the t -tuple that will be covered will be 0. Finally, the value δ will be $l - K + 1$. In general, the construction of \mathcal{A}' can be done in $O(lm)$ steps in the worst case.

Table 2 shows a MAXCOVER instance reduced to an instance of the OSCAR problem. The MAXCOVER instance is $\mathcal{U} = \{u_1, u_2, u_3, u_4\}$, $\mathcal{S} = \{S_1 = \{u_1, u_3\}, S_2 = \{u_1, u_2, u_4\}, S_3 = \{u_2, u_3\}, S_4 = \{u_3, u_4\}\}$, $K = 3$. The matrix \mathcal{A}' will have the columns u_1, u_2, \dots, u_8 , given that $\max\{|S_i|\} = 3$. The rows are the subsets S_1, S_2, S_3, S_4, S_5 . The value $\delta = 3 - 3 + 1 = 1$.

Table 2. The MAXCOVER instance $\mathcal{U} = \{u_1, u_2, u_3, u_4\}$, $\mathcal{S} = \{S_1 = \{u_1, u_3\}, S_2 = \{u_1, u_2, u_4\}, S_3 = \{u_2, u_3\}, S_4 = \{u_3, u_4\}\}$, $K = 3$ presented as an instance of the OSCAR problem

	\mathcal{A}'							
Row	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
S_1	1	0	1	0	0	0	0	0
S_2	1	1	0	1	0	0	0	0
S_3	0	1	1	0	0	0	0	0
S_4	0	0	1	1	0	0	0	0
S_5	0	0	0	0	1	1	1	1

c) Step 3: A solution to the MAXCOVER problem implies a solution of the associated instance of the OSCAR problem

The following step is to prove that a solution for MAXCOVER is maximum if and only if it is minimum for its associated instance of the OSCAR problem. First, note that the subset S' that is a solution in MAXCOVER is also a solution in the OSCAR problem domain, just choose the rows associated with the subsets in S' . Also, the minimization of missing t -wise combinations is equivalent to the maximization of cover t -wise combinations. With these facts in mind we continue with the formulation of **Theorem 1**.

Theorem 1. *Given an instance of MAXCOVER presented as an instance of the OSCAR problem, the optimal solution S^+ that maximizes the number of covered t -wise combinations must contain the row S_{m+1} .*

Proof. Suppose that there exists a solution S' that maximizes the number of covered symbols and it does not contain the row S_{m+1} . Due to any row s in S' is derived from the set \mathcal{S} of the original MAXCOVER instance, this row covers a maximum of $\max\{|S_i|\}$ ones. Now, the row S_{m+1} covers, by definition, $\max\{|S_i|\} + 1$ ones. Then, if we take out s from S' and add S_{m+1} the number of symbols covered will be increased by one; this is in fact a contradiction meaning that S' is not maximum. Hence, an optimal solution S^+ must contain S_{m+1} .

As a result of **Theorem 1**, an optimal solution S^+ with $K + 1$ rows contains K rows from the original set S . By definition, each row derived from S covers the t -tuple 0 for each column u_j , $j > l$. Also, the row S_{m+1} covers the t -tuple 0 in all the columns u_j , for $j \leq l$. Then, the only contribution of each row S_i , for $i \leq m$, is the number of elements that they cover from the set \mathcal{U} . Hence, in the OSCAR problem domain the number of covered t -wise combinations will be $c(S^+ - S_{m+1}) + (m + 1) \cdot \max\{|S_i|\} + l$, where $c(S^+ - S_{m+1})$ is a function that returns the number of elements covered in the original MAXCOVER instance using the subsets $S_i \subseteq S$ derived from $S^+ - S_{m+1}$. Having this into account, we proceed to end the demonstration that OSCAR problem when $\Delta = 0$ is NP-complete (see **Theorem 2**).

Theorem 2. *A MAXCOVER instance has a subset S^* of size K that maximizes the number of covered elements if and only if the subset of $K + 1$ rows formed by $S^* \cup S_{m+1}$ maximizes the number of covered t -wise combinations in its associated instance of the OSCAR problem.*

Proof. Suppose that there exists a MAXCOVER instance whose optimal solution S^* is different from the optimal solution S^+ of its associated instance of the OSCAR problem. According with **Theorem 1** and by definition of optimality, $c(S^*) + (m + 1) \cdot \max\{|S_i|\} + l < c(S^+ - S_{m+1}) + (m + 1) \cdot \max\{|S_i|\} + l$ which implies $c(S^*) < c(S^+ - S_{m+1})$, a contradiction. We can prove the converse in a similar fashion. Hence, a solution of MAXCOVER is optimal if and only if it is also an optimal solution of the OSCAR problem. We proceed to end the demonstration that OSCAR problem is NP-complete with the theorem **Theorem 3**.

Theorem 3. *An instance of the OSCAR problem when $\Delta > 0$ is NP-Complete.*

Proof. Given that an instance of the OSCAR problem when $\Delta > 0$ is equivalent to $\binom{k}{k-\Delta}$ instances of the OSCAR problem when $\Delta = 0$ (the input matrix \mathcal{A} has a size $N \times k - \Delta$), where each one of these instances is NP-Complete (by the previous proof), it is assumed that the OSCAR problem when $\Delta > 0$ is NP-Complete.

3 Algorithms to Solve the OSCAR Problem

One of the main contributions of this paper are an exact algorithm and a greedy algorithm to solve the OSCAR problem. Through this section we will describe the data structures and subroutines required to implement both strategies.

3.1 Greedy Algorithm to Solve the OSCAR Problem

A first algorithm to solve the OSCAR problem is through a non-optimal algorithm. This section presents a greedy algorithm to solve the OSCAR problem. This algorithm uses two functions to delete rows and columns such that a submatrix \mathcal{B} is obtained from a matrix \mathcal{A} with a few missing t -wise combinations. Our

greedy approach for an instance of the OSCAR problem $\mathcal{A}, t, \delta, \Delta$ is described in the remaining of this section.

The algorithm uses two greedy criteria \mathcal{F}_1 and \mathcal{F}_2 . The criteria \mathcal{F}_1 is a function that chooses a column to be deleted. It can be defined as follows. Let \mathcal{A}_C be the set of columns of \mathcal{A} ; K of size $k \times k$ be a matrix where each element $k_{i,j}$ stores the number of times that columns i and j participate in a missing t -wise combination; and $G = \{g_1, g_2, \dots, g_k\}$ be a vector where $g_i = \sum_{j=1}^k k_{i,j}$. The function \mathcal{F}_1 deletes the column $\{c_i | i = \max_j \{g_j\}\}$, ties are broken arbitrarily. Whenever a column i is deleted, the value $k_{i,j}$ is subtracted from g_j , for all $j \neq i$, to update the vector G . Each element in vector G associates to a column the number of times that it participates in a missing t -wise combination.

The criteria \mathcal{F}_2 is a function that chooses a row to be deleted. It can be defined as follows. Let \mathcal{A}_R be the set of rows of \mathcal{A} ; and $R = \{r_1, r_2, \dots, r_N\}$ a vector where each element associates to a row the number of times that it solely covers a t -wise combination. The function \mathcal{F}_2 deletes the row $\{i | i = \min_j \{r_j\}\}$, ties are broken arbitrarily. Now, everytime that a row i is deleted, the number of rows that cover the t -wise combinations associated with the deleted row i must be increased by one. Whenever a t -wise combination has a single row j that covers it, this row must increase the value of r_j by one. We update the vector R following this way.

Once that the greedy criteria $\mathcal{F}_1, \mathcal{F}_2$ have been defined, we proceed to describe our greedy approach. We present three variants for the algorithm $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ to generate a solution \mathcal{B} for an instance of the OSCAR problem. The greedy algorithm \mathcal{G}_1 first deletes Δ columns from \mathcal{A} to generate \mathcal{A}' . After that, it deletes δ rows from \mathcal{A}' to generate \mathcal{B} . The greedy algorithm \mathcal{G}_2 first deletes δ rows from \mathcal{A} to generate \mathcal{A}' . After that, it deletes Δ columns from \mathcal{A}' to generate \mathcal{B} . Finally, our third greedy approach \mathcal{G}_3 distributes the Δ columns into the rows in a round robin fashion, i.e. it assigns the first $\lceil \frac{\Delta}{\delta} \rceil$ columns to r_1 , the next $\lceil \frac{\Delta}{\delta} \rceil$ columns to r_2 , etc. Right after the greedy approach \mathcal{G}_3 deletes a row using \mathcal{F}_2 , it deletes $\lceil \frac{\Delta}{\delta} \rceil$ columns through \mathcal{F}_1 .

The next subsection presents the exact approach used to solve the OSCAR problem.

3.2 Exact Algorithm to Solve OSCAR Problem

This subsection shows two exact solutions for the OSCAR problem proposed in this paper. The solution obtained is the best one, i.e. from the $\binom{N}{N-\delta} \binom{k}{k-\Delta}$ possible solutions there will not be a solution with more covered t -wise combinations than the one reported by the exact approach. The exact algorithms are based on a Branch and Bound (B&B) approach. A B&B algorithm does not explore the whole search space of a problem, instead it uses an intelligent approach to avoid visiting solutions that do not improve the current solution [10]. In the remaining subsection we present the details about how the exact approaches were implemented.

An instance of the OSCAR problem is defined by a matrix \mathcal{A} of size $N \times k$ and the values δ and Δ . Let \mathcal{A}_R and \mathcal{A}_C be the sets of rows and columns

that forms the matrix \mathcal{A} . Then, the whole set of possible solutions is defined by $\{\mathcal{J}_R \cup \mathcal{J}_C | \mathcal{J}_R \subset \mathcal{A}_R, \mathcal{J}_C \subset \mathcal{A}_C, |\mathcal{J}_R| = \delta, |\mathcal{J}_C| = \Delta\}$, where \mathcal{J}_R and \mathcal{J}_C are the selected combinations of rows and columns that form a solution \mathcal{B} . We proposed two strategies to enumerate all the solutions for the OSCAR problem; the pseudocode of these approaches is presented in Figures 3 and 4.

Algorithm 3.1: CRBB($\mathcal{A}_{N \times k}, t, \delta, \Delta$)

```

 $\mathcal{B} \leftarrow \text{BESTGREEDY}(\mathcal{A}_{N \times k}, t, \delta, \Delta);$ 
 $\mathcal{U} \leftarrow \tau(\mathcal{B});$ 
for each  $\mathcal{J}_C \in \{\mathcal{J} | \mathcal{J} \subset \mathcal{A}_C, |\mathcal{J}| = \Delta\}$ 
   $\mathcal{B}' \leftarrow \mathcal{A}_R \cup \mathcal{A}_C - \mathcal{J}_C;$ 
  PUSH( $\mathcal{S}, 0, 0$ );
  while NOTEMPTY( $\mathcal{S}$ )
    POP( $\mathcal{S}, \alpha, i$ );
    if  $\alpha = \delta$  or  $i = N - \delta + \alpha$ 
      then  $\left\{ \begin{array}{l} \text{if } \alpha = \delta \text{ AND } \tau(\mathcal{B}') < \mathcal{U} \\ \text{then } \left\{ \begin{array}{l} \mathcal{B}' \leftarrow \mathcal{B}' \\ \mathcal{U} \leftarrow \tau(\mathcal{B}'); \end{array} \right. \end{array} \right.$ 
    do  $\left\{ \begin{array}{l} \text{PUSH}(\mathcal{S}, \alpha, i+1); \\ \mathcal{B}' \leftarrow \mathcal{B}' - r_\alpha; \\ \text{if } \tau(\mathcal{B}') < \mathcal{U} \\ \text{then } \left\{ \begin{array}{l} \text{PUSH}(\mathcal{S}, \alpha+1, i+1); \\ \text{else } \mathcal{B}' \leftarrow \mathcal{B}' \cup r_\alpha; \end{array} \right. \end{array} \right.$ 
  return ( $\mathcal{B}$ )

```

Fig. 3. B&B with row branching

Algorithm 3.2: RCBB($\mathcal{A}_{N \times k}, t, \delta, \Delta$)

```

 $\mathcal{B} \leftarrow \text{BESTGREEDY}(\mathcal{A}_{N \times k}, t, \delta, \Delta);$ 
 $\mathcal{U} \leftarrow \tau(\mathcal{B});$ 
for each  $\mathcal{J}_R \in \{\mathcal{J} | \mathcal{J} \subset \mathcal{A}_R, |\mathcal{J}| = \delta\}$ 
   $\mathcal{B}' \leftarrow \mathcal{A}_R - \mathcal{J}_R;$ 
  PUSH( $\mathcal{S}, 0, 0$ );
  while NOTEMPTY( $\mathcal{S}$ )
    POP( $\mathcal{S}, \alpha, j$ );
    if  $\alpha = \Delta$  or  $j = k - \Delta + \alpha$ 
      then  $\left\{ \begin{array}{l} \text{if } \alpha = \Delta \text{ AND } \tau(\mathcal{B}') < \mathcal{U} \\ \text{then } \left\{ \begin{array}{l} \mathcal{B}' \leftarrow \mathcal{B}' \\ \mathcal{U} \leftarrow \tau(\mathcal{B}'); \end{array} \right. \end{array} \right.$ 
    do  $\left\{ \begin{array}{l} \text{PUSH}(\mathcal{S}, \alpha, j+1); \\ \mathcal{B}' \leftarrow \mathcal{B}' \cup c_\alpha; \\ \text{if } \tau(\mathcal{B}') < \mathcal{U} \\ \text{then } \text{PUSH}(\mathcal{S}, \alpha+1, j+1); \\ \text{else } \mathcal{B}' \leftarrow \mathcal{B}' - c_\alpha; \end{array} \right.$ 
  return ( $\mathcal{B}$ )

```

Fig. 4. B&B with column branching

The B&B algorithm shown in **Figure 3** can be described in three simple steps. First, it selects a combination of columns $\mathcal{J}_C = \{J_{C,0}, J_{C,1}, \dots, J_{C,\Delta-1}\}$, where $J_{C,j} \subset \mathcal{A}_C$. After that it deletes \mathcal{J}_C from \mathcal{A} to form the submatrix \mathcal{B}' . Finally, the algorithm recursively constructs each possible set $\mathcal{J}_R = \{J_{R,0}, J_{R,1}, \dots, J_{R,\delta-1}\}$, where $J_{R,i} \subset \mathcal{A}_R$, that can be deleted from \mathcal{B}' so that \mathcal{B} of size $N - \delta \times k - \Delta$ is build. To construct \mathcal{J}_R , the algorithm chooses a value for $J_{R,0}$ and proceeds in lexicographical order. Each time that an element $J_{R,i}$ is deleted from \mathcal{B}' the bounding function is called to prune the search space as much as possible. This algorithm is called the Column-Row B&B (CRBB).

The second B&B design is similar to the CRBB algorithm (see **Figure 4**). This algorithm chooses the set \mathcal{J}_R and deletes it from \mathcal{A}_R to form \mathcal{B}' . Then, the columns are added to \mathcal{B}' one at a time until \mathcal{B} of size $N - \delta \times k - \Delta$ is constructed. Again, in this B&B approach a bound is applied each time that a column is added to \mathcal{B}' . We choose adding columns in this approach instead of deleting them because it allows the same bounding function used in CRBB. This algorithm is called the Row-Column B&B (RCBB).

The bound of the algorithms is performed by the value \mathcal{U} and the value $\tau(\mathcal{B}')$. The initial \mathcal{U} is provided by the best obtained solution from the greedy algorithms.

4 Experimentation and Results

This subsection shows a set of experiments designed to evaluate the performance of the exact and greedy algorithms. The algorithms were implemented in C language and compiled with gcc using the optimization option -O3. The algorithms were run in a cluster with eight nodes, each one with two dual-core Opteron Processors. Each node has 4GB RAM memory. The operating system is Red Hat Enterprise Linux 4.

The first experiment makes the comparison of the greedy and exact approaches. A benchmark formed by 20 instances of the OSCAR problem, derived from 12 CAs taken from the literature related to CAs was used. These CAs have strength $t = 2$ and alphabet values $v = \{3, 5, 6, 7, 8, 9\}$.

Table 3. Results from the experiments and new upper bounds

\mathcal{A}	δ	Δ	$\tau(\mathcal{B})$	Time [secs]	New bound	Old bound	SA(\mathcal{B})	Time [secs]
CA(53;2,52,5)	4	9	272 _{\mathcal{G}_3}	0.15	CAN(2,43,5) \leq 49	50	0	2
CA(53;2,52,5)	3	7	229 _{\mathcal{G}_3}	0.15	CAN(2,45,5) \leq 50	52	0	1
CA(53;2,52,5)	2	5	173 _{\mathcal{G}_3}	0.10	CAN(2,47,5) \leq 51	53	0	27.43
CA(93;2,113,6)	1	6	154 _{\mathcal{G}_3}	.2	CAN(2,107,6) \leq 92	93	0	1
CA(120;2,80,8)	1	51	33 _{\mathcal{G}_2}	0.15	CAN(2,29,8) \leq 119	120	0	1
CA(120;2,80,8)	2	52	60 _{\mathcal{G}_3}	0.15	CAN(2,28,8) \leq 118	120	0	186.4
CA(120;2,80,8)	6	55	138 _{\mathcal{G}_3}	0.30	CAN(2,25,8) \leq 114	120	0	1822.4
CA(120;2,80,8)	7	56	147 _{\mathcal{G}_3}	0.30	CAN(2,24,8) \leq 113	120	0	1
CA(153;2,99,9)	9	74	180 _{\mathcal{G}_3}	0.70	CAN(2,25,9) \leq 144	153	0	1177.1
CA(153;2,99,9)	8	73	176 _{\mathcal{G}_3}	0.70	CAN(2,26,9) \leq 145	153	0	4709
CA(153;2,99,9)	2	69	60 _{\mathcal{G}_3}	0.30	CAN(2,30,9) \leq 151	153	0	812
CA(188;2,140,9)	3	0	103 _{\mathcal{G}_3}	1.80	CAN(2,140,9) \leq 185	188	0	1616.51
CA(82;2,17,7)	1	1	19 _{CRBB}	0.10	-	82	4	-
CA(11;2,5,3)	1	0	6 _{\mathcal{G}_2}	0.10	-	11	4	-
CA(15;2,20,3)	1	9	9 _{\mathcal{G}_2}	0.10	-	15	3	-
CA(33;2,8,5)	1	0	10 _{\mathcal{G}_2}	0.10	-	33	4	-
CA(62;2,17,6)	1	0	1 _{\mathcal{G}_2}	0.10	-	62	1	-
CA(102;2,11,9)	1	0	12 _{\mathcal{G}_2}	0.10	-	102	2	-
CA(36;2,10,5)	1	0	1 _{\mathcal{G}_2}	0.10	-	36	1	-

In order to compare the approaches $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, CRBB, RCBB$, we measure the number of missing t -wise combinations and the time (in seconds) spent. The results from this experiment are shown in **Table 3**. The CAs (or input matrices for the OSCAR problem) are described in column 1. The values of δ and Δ considered for each instance are shown in columns 2 and 3. The column 4 presents the best value $\tau(\mathcal{B})$ obtained for each instance, i.e. the minimum number of missing t -wise combinations. The subindex indicates the algorithm that obtained the value $\tau(\mathcal{B})$ in less time.

The second experiment was performed with the purpose of evaluating how well a metaheuristic behave when constructing CAs if it takes as initial matrix the solution given by the solution of the OSCAR problem. In this case we tested the

simulated annealing algorithm that constructs CAs (we refer to SACAC to this algorithm) reported in [22]. **Table 3** shows in column 8 the number of missing t -wise combinations produced by SACAC. Note that in 12 of the 20 instances the t -wise combinations were reduced to zero. This enables the generation of new CAs, which are reported in column 6. The CAs constructed through SACAC, using the submatrices derived from the solution of instances of the OSCAR problem, reduced the upper bound for the CAs (shown in column 7).

5 Conclusions

This paper has defined the problem of Optimal Shortening of Covering ARrays (OSCAR). The OSCAR problem is an optimization problem that for a given matrix \mathcal{M} consists in finding a submatrix \mathcal{M}' that is near to be a CA. Also, we presented the proof that the OSCAR problem is an NP-complete problem, from the reduction of MAXCOVER to it. A greedy and an exact algorithm to solve the OSCAR problem were presented. These algorithms were tested using a benchmark formed by 20 instances of the OSCAR problem, derived from 12 CAs taken from the related literature. The best submatrices obtained through these approaches were used as initial matrices of a simulated annealing algorithm for constructing CAs (SACAC). The SACAC algorithm constructed CAs for 12 of the 20 cases, and produced new upper bounds.

Acknowledgements. This research was partially funded by the following projects: CONACyT 58554-Cálculo de Covering Arrays, 51623-Fondo Mixto CONACyT y Gobierno del Estado de Tamaulipas

References

1. Bryce, R.C., Colbourn, C.J., Cohen, M.B.: A framework of greedy methods for constructing interaction test suites. In: International Conference on Software Engineering, pp. 146–155 (2005)
2. Bush, K.A.: Orthogonal arrays of index unity. Annals of Mathematical Statistics 23(3), 426–434 (1952)
3. Chateauneuf, M., Kreher, D.L.: On the state of strength-three covering arrays. Journal of Combinatorial Design 10(4), 217–238 (2002)
4. Cohen, D.M., Colbourn, C.J., Ling, A.C.H.: Constructing strength three covering arrays with augmented annealing. Discrete Mathematics 308(13), 2709–2722 (2008)
5. Cohen, D.M., Dalal, S.R., Fredman, M.L., Patton, G.C.: The aetg system: An approach to testing based on combinatorial design. IEEE Transactions on Software Engineering 23(7), 437–444 (1997)
6. Colbourn, C.J., Cohen, M.B.: A deterministic density algorithm for pairwise interaction coverage. In: Proceedings of the IASTED International Conference on Software Engineering, pp. 242–252 (2004)
7. Colbourn, C.J.: Combinatorial aspects of covering arrays. Le Matematiche 58, 121–167 (2004)
8. Colbourn, C.J., Dinitz, J.H.: The CRC handbook of combinatorial designs. CRC Press (1999)

9. Colbourn, C.J., Martirosyan, S.S., Mullen, G.L., Shasha, D., Sherwood, G.B., Lucas, J.L.: Products of mixed covering arrays of strength two. *Journal of Combinatorial Design* 14(2), 124–138 (2006)
10. Donald, L.K., Stinson, D.R.: Combinatorial algorithms: generation, enumeration, and search. CRC Press (1999)
11. Feige, U.: A threshold of $\ln n$ for approximating set cover. *J. ACM* 45(4), 634–652 (1998)
12. Forbes, M., Lawrence, J., Lei, Y., Kacker, R.N., Kuhn, D.R.: Refining the in parameter order strategy for constructing covering arrays. *Journal of Research of the National Institutes of Standards and Technology* 113(5), 217–238 (2008)
13. Hartman, A., Raskin, L.: Problems and algorithms for covering arrays. *Discrete Mathematics* 284(1-3), 149–156 (2004)
14. Kuhn, D.R., Wallace, D.L., Gallo, A.M.: Software fault interaction and implications for software testing. *IEEE Transactions on Software Engineering* 30(6), 418–421 (2004)
15. Lawrence, J.F., Kacker, R.N., Lei, Y., Kuhn, D.R., Forbes, M.: A survey of binary covering arrays. *Journal of Combinatorial Designs* 18(84), 1741–1758 (2011)
16. Lei, Y., Tai, K.C.: In-parameter-order: A test generation strategy for pairwise testing. In: The 3rd High-Assurance Systems Engineering, IEEE International Symposium on High-Assurance Systems Engineering, pp. 254–261. IEEE Computer Society, Los Alamitos (1998)
17. Martirosyan, S., Trung, T.V.: On t-covering arrays. *Des. Codes Cryptography* 32(1-3), 323–339 (2004)
18. Nurmela, K.J.: Upper bounds for covering arrays by tabu search. *Discrete Applied Mathematics* 138(1-2), 143–152 (2004)
19. Rodriguez-Tello, E., Torres-Jimenez, J.: Memetic Algorithms for Constructing Binary Covering Arrays of Strength Three. In: Collet, P., Monmarché, N., Legrand, P., Schoenauer, M., Lutton, E. (eds.) EA 2009. LNCS, vol. 5975, pp. 86–97. Springer, Heidelberg (2010)
20. Shiba, T., Tsuchiya, T., Kikuno, T.: Using artificial life techniques to generate test cases for combinatorial testing. In: Proceedings of the 28th Annual International Computer Software and Applications Conference, COMPSAC 2004, vol. 1, pp. 72–77 (2004)
21. Sloane, N.J.A.: Covering arrays and intersecting codes. *Journal of Combinatorial Designs* 1(1), 51–63 (1993)
22. Torres-Jimenez, J., Rodriguez-Tello, E.: Simulated annealing for constructing binary covering arrays of variable strength. In: Proceedings of the CEC 2010, pp. 4102–4109 (2010)
23. Tung, Y.W., Aldiwan, W.S.: Automating test case generation for the new generation mission software system. In: 2000 IEEE Aerospace Conference Proceedings, vol. 1, pp. 431–437 (2000)

An Exact Approach to Maximize the Number of Wild Cards in a Covering Array

Loreto Gonzalez-Hernandez, José Torres-Jiménez, and Nelson Rangel-Valdez

Information Technology Laboratory, CINVESTAV-Tamaulipas

Km. 5.5 Carretera Cd. Victoria-Soto la Marina, 87130, Cd. Victoria Tamps., Mexico
agonzalez@tamps.cinvestav.mx, jtj@cinvestav.mx, nrangel@tamps.cinvestav.mx
<http://www.tamps.cinvestav.mx/~jtj/>

Abstract. Covering Arrays CA($N; t, k, v$) are combinatorial structures that can be used to define adequate test suites for software testing. The smaller a CA is, the smaller the number of test cases that will be given to test the functionality of a software component in order to identify possible failures. Due to the fact that the construction of CAs of optimal size is a highly combinatorial problem, several approximated strategies have been developed. Some constructions of these strategies can be further improved through a post optimization process. For example, the wild card profile of a CA is the set of symbols that can be modified without changing the properties that define a CA. It has been shown that some CAs can be reduced by merging rows that contain wild cards. This paper presents a Branch and Bound (B&B) strategy that maximizes the number of wild cards in the profile of an already constructed CA. We identify such profiles in 106 CAs of strength $t = 2$ and alphabets v from 6 to 25. Also, it is shown that for an specific CA(42; 2, 8, 6) different profiles can be obtained; such profiles vary in the number of wild cards and their distribution in the CA.

Keywords: Software Testing, Covering Arrays, Wild Card Profile, Exact Approach.

1 Introduction

Most of the enterprises nowadays depend on computer software [13], thus a failure on it can lead disastrous consequences. The National Institute of Standards and Technology (NIST) reported that the USA economy has been affected because software defects (representing a cost of 60 billion), therefore is essential to carry out the testing stage appropriately.

It will be ideal to test all the possible inputs cases but it usually demands too large cost and time, for example if we need to test a system with 12 parameters and 4 possible settings each, it would required $4^{12} = 16,777,216$ test configurations to carry out testing in exhaustive way. Given that the number of necessary tests grows exponentially when the number of components increases, another alternative to create a test suite has to be used. An approach called *interaction*

testing is an alternative to carry out the verification of the quality of software, which is based on the fact that many software systems of different kinds are built using parameters and their faults are caused by an unexpected interaction among subsets of the parameters. While not conclusive, the results reported in [8,9] suggest that combinatorial testing which exercises strength interaction of size 2 to 6 can be an effective method to software assurance.

Interaction testing uses combinatorial objects to indicate the test cases, a Covering Array ($CA(N; t, k, v)$) is one of the mostly used; this is a matrix where every row indicates the test case and the columns represent the k parameters of the software. When a CA contains the minimum number of rows is optimal and it is represented by $CAN(t, k, v)$. The problem to obtain optimal CAs is highly combinatorial [11], so many techniques have been implemented in order to construct CAs [6,10], however, some of the fastest approaches do not always guarantee to generate the optimal, hence, some CAs are susceptible of being improved by decreasing their number of rows through post-optimization process.

The work presented in this paper is related with maximizing the number of wild cards (or symbols that can be changed arbitrarily in the CA without losing its degree of coverage). Wild cards have been also used in some recursive constructions [4,5,3] and recently in post-optimization processes [12]. This paper presents an exact algorithm based on the Branch and Bound (B&B) approach; it maximizes the wild cards of a profile of a CA (i.e. the number of symbols that are wild cards). The Branch and Bound algorithm to maximize the number of wildcards will be called **wcBBCA**. The **wcBBCA** was used to construct the wild card profile of 100 CAs reported in the literature. Also, the **wcBBCA** was used to identify 36 different wild card profiles for $CA(48; 2, 8, 6)$.

The remaining of the paper is organized in the following sections. Section 2 presents the basic background about CAs and notation used in the paper; it also shows some related work about post-optimization processes of CAs. Section 3 gives an overview of the proposed exact algorithm **wcBBCA**. After that, the Section 4 presents the benchmarks used to test our exact approach. It includes the results from the experiments performed to evaluate **wcBBCA**. Finally, Section 5 summarizes the main contributions of our work.

2 Background

One of the combinatorial objects that can be constructed in order to perform an interaction test is a *Covering Array* (or CA). A $CA(N; t, k, v)$ is an $N \times k$ array consisting of N vectors of length k with entries from $\{0, 1, \dots, v - 1\}$ (v is the cardinality of the alphabet) such that every one of the v^t possible vectors of size t occurs at least once in every possible selection of t elements from the vectors [7]. The parameter t is referred as the *strength*. The minimum number of rows required to construct a CA for an specific value of v , k and t is called covering array number which is denoted by $CAN(t, k, v)$ [2].

When a combination in a t -tuple is covered more than once, there is the possibility of changing some symbols arbitrarily such that the CA does not lose its level of coverage, these symbols are referred as wild cards. **Table 1** shows the use of them using asterisks *.

Table 1. Identification of wild cards in the CA(6; 2, 4, 2)

(a) CA(6; 2, 4, 2)				(b) CA(5; 2, 4, 2)			
C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
1	1	1	0	1	1	1	0
1	1	0	1	1	1	0	1
1	0	1	1	1	0	1	1
0	1	1	1	0	1	1	1
0	0	0	0	0	0	0	0
0	1	0	0	*	*	*	*

In **Table 1b** can be observed that the matrix is still a CA, because the combinations (0,0), (0,1) (1,0) (1,1) appear at least once in every pair of columns (C_1, C_2) (C_1, C_3) ... (C_3, C_4). Due to all symbols in the last row are wild cards, it can be deleted to create the CA(5; 2, 4, 2) which is better than the original CA(6; 2, 4, 2).

Wild cards have different uses, recently they have been used by Nayery et al. [12] in a post-optimization process for CAs. They presented an algorithm to find wild cards and use them to reduce, if it is possible, the number of rows of a CA.

3 Exact Approach for Wild Card Detection

The methodology that we proposed for identifying the maximum number of wild cards consists in three steps: a) the determination of the combinations of symbols that are covered only once; b) the determination of unfixed symbols; and c) the enumeration of all the possible wild card configuration using a B&B approach. The exact approach uses as initial lower bound the solution obtained from a simple greedy approach.

The next subsections describe in detail each step of the general methodology. For this purpose we define $\mathcal{M}_{N \times k}$ as a Covering Array CA($N; t, k, v$), and $m_{i,j} \in \mathcal{M}$ a cell in the matrix that contains a symbol from $\{0, 1, \dots, v - 1\}$.

3.1 Symbol Fixing Process

The first step in the methodology is the fixing process. Here, we look for those combination of symbols that are covered only once. For example, lets consider the matrix $\mathcal{M} = \text{CA}(6; 2, 4, 2)$ shown in **Table 2a** and the combination of symbols (0, 0), the columns (C_1, C_2) and row R_1 . The combination (1, 1) appears only once in (C_1, C_2), then it can't never be a wild card * because if we change one of its values, then the matrix \mathcal{M} will not be a CA any longer. The symbols

that belong to this kind of combinations are referred as **fixed symbols** or just **fixed**.

Table 2b shows the matrix $\mathcal{M} = \text{CA}(6; 2, 4, 2)$ after the symbol fixing process. The symbols marked with U are considered **UNFIXED**. These symbols can eventually become a wild card. The remaining cells in the matrix are considered **fixed**; a symbol change in any of them will convert \mathcal{M} in a non CA.

Table 2. CA(5; 2, 3, 2) and its fixed symbols

	(a) \mathcal{M}					(b) \mathcal{M}'			
	C_1	C_2	C_3	C_4		C_1	C_2	C_3	C_4
R_1	0	0	0	0		R_1	0	0	0
R_2	0	1	1	1		R_2	U	1	U
R_3	1	0	1	0		R_3	1	0	1
R_4	1	1	0	1		R_4	1	1	0
R_5	0	1	1	0		R_5	U	1	U
R_6	0	0	1	1		R_6	U	0	U

3.2 Exact Algorithm for Wild Card Identification: wcBBCA

In order to identify the maximum number of possible wild cards, we need to define a scheme to enumerate all the possible solutions. The fixing process leads to the definition of such scheme. The scheme is described in the following paragraphs.

A wild card is a symbol that can be modified without affecting the property that a matrix \mathcal{M} be a CA. Based on this definition, only the symbols marked as **UNFIXED** in the fixing process can be wild cards. These facts lead to the following simple method to enumerate all the possible configurations of wild cards: for each possible subset \mathcal{W} of **UNFIXED** symbols, mark in \mathcal{M} the elements $w_i \in \mathcal{W}$ as wild cards (with the symbol *), and let the remaining symbols in the matrix as fixed symbols, then test if the matrix is still a CA. The solution for the problem of maximizing the number of wild cards will be the subset \mathcal{W}^* of the maximum size.

Table 3 illustrates the simple process of maximizing wild cards, described in the previous paragraph. This table shows three possible wild cards distributions (denoted by \mathcal{A}_i) derived from the CA shown in **Table 2**. **Tables 3a** and **3b** show a matrix with two wild cards each; the matrices are CAs. **Table 3c** shows a matrix with four wild cards; this matrix is not a valid distribution of wild cards due to it is not a CA (the combination of symbols (0, 1) of columns (C_1, C_4) is missing in the matrix).

The process exemplified in **Table 3** suggests an enumeration scheme that requires to test every possible subset of **UNFIXED** symbols, i.e. around 2^m solutions, where m is the number of **UNFIXED**. An alternative enumeration scheme can be obtained through a different representation of the solution. The rest of this subsection explains this new representation and presents the algorithm for the B&B algorithm **wcBBCA**.

Table 3. Three different distributions of wild cards for the CA(5; 2, 3, 2) shown in **Table 2**

	(a) \mathcal{A}_1				(b) \mathcal{A}_2				(c) \mathcal{A}_3			
	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
R_1	0	0	0	0	0	0	0	0	0	0	0	0
R_2	*	1	*	1	0	1	1	1	*	1	*	1
R_3	1	0	1	0	1	0	1	0	1	0	1	0
R_4	1	1	0	1	1	1	0	1	1	1	0	1
R_5	0	1	1	0	*	1	*	0	0	1	1	0
R_6	0	0	1	1	0	0	1	1	*	0	*	1

Let \mathcal{M} be a CA($N; t, k, v$); we can represent \mathcal{M} through the sets $R, C, \tau, \mathcal{V}, \mathcal{R}_{C_i, V_j}$. The set $R = \{R_1, R_2, \dots, R_N\}$ is the set of rows of \mathcal{M} . The set $C = \{C_1, C_2, \dots, C_k\}$ is the set of columns. The set $\tau = \{\tau_1, \tau_2, \dots, \tau_{\binom{k}{t}}\}$ is the set of all the possible combination of t columns (e.g. $\tau_1 = (C_1, C_2, C_3)$ for $t = 3$). The set $\mathcal{V} = \{V_1, V_2, \dots, V_{v^t}\}$, where each V_i is a possible combination of v^t . The set $\mathcal{R}_{\tau_i, V_j}$ is the subset of rows $R' \subset R$ that can cover combination V_j in the columns τ_i of \mathcal{M} .

Using the sets $R, C, \tau, \mathcal{V}, \mathcal{R}_{C_i, V_j}$ we can define the solution of the problem of maximizing wild cards using the matrix $\mathcal{B}_{\binom{k}{t} \times v^t}$. Each row of this new matrix makes reference to a particular combination of t columns τ_i . The columns are related with the combination of symbols V_j . The value of the cell $\mathcal{B}_{\tau_i, V_j} \in \mathcal{B}$ is a row $R_l \in R$; this means that the row R_l will be used to cover the combination of symbols V_j in the columns described by τ_i .

Based on the previous described representation, Algorithm 1 enumerates all the possible wild card distributions and finds the one that maximizes the number of wild cards. The inputs of the algorithm are: a) the solution that is being constructed \mathcal{B} , which initially is \emptyset (i.e. it has no value assigned); b) an initial solution \mathcal{B}^* obtained through a greedy approach; c) the set \mathcal{R} ; and d) the value i which indicates the pair (τ_k, V_l) that most be analyzed in each of its iterations, initially $i = 1$. For each pair (τ_k, V_l) , the algorithm tries every different row $r \in \mathcal{R}_{\tau_k, V_l}$ as a possible cover of the specific combination of symbols V_l in the subset of t columns τ_k . The solution that is being constructed is stored in \mathcal{B} . Whenever the solution \mathcal{B} becomes smaller than \mathcal{B}^* , a new branching is tried. Every time that the algorithm reaches the last recursion level (when $i = v^t \cdot \binom{k}{t}$), an improvement in the best solution is obtained and \mathcal{B}^* is changed by \mathcal{B} . The function $\text{Decode}(i, \tau_k, V_l)$ is a map between the set of natural numbers N and the vectors defined by (τ_k, V_l) ; this function makes it possible to test every possible assignation of rows to combination of t columns to fix symbols.

The algorithm for the construction of the initial solution \mathcal{B}^* follows a simple greedy approach. For each combination of columns τ_i , starting from $i = 1$, first of all it verifies if V_i was covered for a combination of **fixed** symbols after the first stage, otherwise, it chooses the row R_j with the minimum number of **UNFIXED**

Algorithm 1. B&B algorithm wcBBCA for the identification of wild cards in a CA

```

wcBBCA( $\beta, \beta^*, \mathcal{R}, i$ )
if  $i \geq v^t \cdot \binom{k}{t}$  then
     $\beta^* \leftarrow \beta$ ;
    return 0;
end
Decode( $i, \tau_k, V_l$ );
if  $|\mathcal{R}_{\tau_k, V_l}| = 1$  then
    return 0;
end
foreach  $r \in \mathcal{R}_{\tau_k, V_l}$  do
     $\beta_{\tau_k, V_l} \leftarrow r$ ;
    if  $f(\beta) > f(\beta^*)$  then
        wcBBCA( $\beta, \mathcal{R}, i + 1$ );
    end
     $\beta_{\tau_k, V_l} \leftarrow \emptyset$ ;
end
1 return 0;

```

symbols found in the columns defined by τ_i . If more than one row have the same number of UNFIXED symbols, then the first row with the minimum number of UNFIXED symbols in all the columns is selected. The process is repeated in increasing order of the number of combination of columns $i = \{1, 2, \dots, \binom{k}{t}\}$.

4 Experimental Results

This section presents an experimental design to test the performance of our B&B approach. Three experiments were developed with this purpose. The first experiment evaluates the approach in terms of the number of iterations and the time spent by it to identify the wild cards, to obtain the goal of this experiment, a benchmark was proposed, its main characteristic is that the CAs were constructed using the method reported by Bush [1]. The second experiment involves the identification of wild cards in CAs constructed using the procedure reported in [4]. The third experiment shows the importance of the input matrix for wcBBCA in the identification of wild card profiles. The experiments are detailed in the following subsections.

The wcBBCA was implemented in C language and compiled with gcc. The experiments were carried out in a CPU Intel Core 2 Duo at 1.5 GHz, 2 GB of RAM and Ubuntu 8.10 Intrepid Ibex Operating System.

4.1 Experiment 1: Identification of Wild Cards in a CA Benchmark

The first experiment had as goal to test the algorithm in a set of CAs proposed in this paper, such CAs were obtained through the Bush construction [1]. For this purpose, we firstly define a target benchmark composed by 10 CAs of strength $t = 2$ and alphabets v from 6 to 25. For each CA in this benchmark, we identify three different wild card profiles using wcBBCA.

Let t and v be the strength and alphabet of a CA in the benchmark, we get the profiles $\text{CA}(N; t, w_1 + 1, v)$, $\text{CA}(N; t, w_2 + 1, v)$, $\text{CA}(N; t, w_3 + 1, v)$ from $\text{CA}(M_1; t, w_1 + 1, w_1)$, $\text{CA}(M_2; t, w_2 + 1, w_2)$, $\text{CA}(M_3; t, w_3 + 1, w_3)$, where w_1, w_2, w_3 are the following three prime powers greater than v . For example, given the $\text{CA}(N; 2, k, 6)$ we obtain its profiles from $\text{CA}(M_1; 2, 8, 7)$, $\text{CA}(M_1; 2, 9, 8)$, $\text{CA}(M_1; 2, 10, 9)$.

The methodology to construct a $\text{CA}(N; t, k, v)$ from a $\text{CA}(M; t, k, w)$ (here $w = v + \delta$ for $\delta \geq 1$) is simple. Firstly we place, in a list L , all symbols in increasing order of frequently, i.e. at the beginning of the list is placed the symbol that appears the most times in the CA and the least frequent symbol is place at the end. Secondly, we change to 0 all cells that contain the symbol indicated in the position l_0 from the list L , after that, we change to 1 all the cells that contain the symbol indicated in l_1 and so on until l_{w-1} . After that, the reorganized $\text{CA}(M; t, k, w)$ is transformed into a $\text{CA}(M; t, k, v)$ by changing all the symbols in the range $\{(w - v - 1), \dots, (w - 1)\}$ in $\text{CA}(M; t, k, w)$ to UNFIXED. Finally, our exact approach (see Section 3) is used to identify a wild card profile in the transformed matrix $\text{CA}(M; t, k, v)$.

Table 4 shows the benchmark of CAs and their profiles. Each row reports a different profile. The column 1 shows the initial matrices. The column 2 shows the CAs that were constructed. The column 3 presents the number of wild cards identified by wcBBCA. The column 4 presents the wild card profile; the format $w_1^{g_1} w_2^{g_2} \dots$ shows the number of wild cards per column, i.e. $w_i^{g_i}$ means that g_i columns have w_i wild cards. The columns 5 and 6 show the number of iterations and time (in seconds) spent by wcBBCA to find the profile with the maximum number of wild cards. The total number of iterations includes the times that the evaluation function $f(\mathcal{B})$ is called by wcBBCA, and the iterations produced by the greedy approach (i.e. $2 \cdot \binom{k}{t}$ iterations per alphabet).

The results shown in **Table 4** lead us to point out two main observations. The first one is that the wcBBCA algorithm spent less than a second to identify the profiles for each of the 60 CAs that form the benchmark; this also includes the cases where the profile for a $\text{CA}(N; t, k, v)$ was identified from a CA of greater alphabet.

The second observation is that after the identification of the profiles, all the CAs reduced their size; it was possible because rows fulled with wild cards were produced during the process, and then some rows could be eliminated.

In conclusion, this experiment showed that wcBBCA had a good performance in the identification of wild card profiles for CAs of different sizes. The following section extends the analysis of the performance of wcBBCA, it analyses the wild card profile of CAs constructed from an algebraic method.

4.2 Experiment 2: Wild Card Profile Detection for Larger CAs

The purpose of this experiment is to analyze the wild card profile of CAs constructed through algebraic methods. The methodology is simple, we use the CAs constructed in the previous experiment as input of the method reported by Colbourn in [4]; then, we determine the wild card profile for the new CAs using wcBBCA. An analysis of the results is presented at the end of the subsection.

Table 4. Benchmark of CAs where the wild card profile for each instance is identified using wcBBCA

Initial Matrix	Final CA	Profile		Iter.	wcBBCA Time(secs.)
		Total WC	WC Distribution		
$CA(49; 2, 8, 7)$	$\rightarrow CA(48; 2, 8, 6)$	49	$6^7 7^1$	56	0.000
$CA(64; 2, 9, 8)$	$\rightarrow CA(62; 2, 9, 6)$	128	$14^8 16^1$	72	0.000
$CA(81; 2, 10, 9)$	$\rightarrow CA(78; 2, 10, 6)$	243	$24^9 27^1$	90	0.000
$CA(64; 2, 9, 8)$	$\rightarrow CA(63; 2, 9, 7)$	64	$7^8 8^1$	72	0.000
$CA(81; 2, 10, 9)$	$\rightarrow CA(79; 2, 10, 7)$	162	$16^9 18^1$	90	0.000
$CA(121; 2, 12, 11)$	$\rightarrow CA(117; 2, 12, 7)$	484	$40^{11} 44^1$	132	0.000
$CA(81; 2, 10, 9)$	$\rightarrow CA(80; 2, 10, 8)$	81	$8^9 9^1$	90	0.000
$CA(121; 2, 12, 11)$	$\rightarrow CA(118; 2, 12, 8)$	363	$30^{11} 33^1$	132	0.000
$CA(169; 2, 14, 13)$	$\rightarrow CA(164; 2, 14, 8)$	845	$60^{13} 65^1$	182	0.000
$CA(121; 2, 12, 11)$	$\rightarrow CA(119; 2, 12, 9)$	242	$20^{11} 22^1$	132	0.000
$CA(169; 2, 14, 13)$	$\rightarrow CA(165; 2, 14, 9)$	676	$48^{13} 52^1$	182	0.000
$CA(256; 2, 17, 16)$	$\rightarrow CA(249; 2, 17, 9)$	1792	$105^{16} 112^1$	272	0.000
$CA(121; 2, 12, 11)$	$\rightarrow CA(120; 2, 12, 10)$	121	$10^{11} 11^1$	132	0.000
$CA(169; 2, 14, 13)$	$\rightarrow CA(166; 2, 14, 10)$	507	$36^{13} 39^1$	182	0.000
$CA(256; 2, 17, 16)$	$\rightarrow CA(250; 2, 17, 10)$	1536	$90^{16} 96^1$	272	0.000
$CA(169; 2, 14, 13)$	$\rightarrow CA(167; 2, 14, 11)$	338	$24^{13} 26^1$	182	0.000
$CA(256; 2, 17, 16)$	$\rightarrow CA(251; 2, 17, 11)$	1280	$75^{16} 80^1$	272	0.000
$CA(289; 2, 18, 17)$	$\rightarrow CA(283; 2, 18, 11)$	1734	$96^{17} 102^1$	306	0.010
$CA(169; 2, 14, 13)$	$\rightarrow CA(168; 2, 14, 12)$	169	$12^{13} 13^1$	182	0.000
$CA(256; 2, 17, 16)$	$\rightarrow CA(252; 2, 17, 12)$	1024	$60^{16} 64^1$	272	0.000
$CA(289; 2, 18, 17)$	$\rightarrow CA(284; 2, 18, 12)$	1445	$80^{17} 85^1$	306	0.000
$CA(256; 2, 17, 16)$	$\rightarrow CA(253; 2, 17, 13)$	768	$45^{16} 48^1$	272	0.000
$CA(289; 2, 18, 17)$	$\rightarrow CA(285; 2, 18, 13)$	1156	$64^{17} 68^1$	306	0.000
$CA(361; 2, 20, 19)$	$\rightarrow CA(355; 2, 19, 13)$	2166	$108^{19} 114^1$	380	0.000
$CA(256; 2, 17, 16)$	$\rightarrow CA(254; 2, 17, 14)$	512	$30^{16} 32^1$	272	0.010
$CA(289; 2, 18, 17)$	$\rightarrow CA(286; 2, 18, 14)$	867	$48^{17} 51^1$	306	0.000
$CA(361; 2, 20, 19)$	$\rightarrow CA(356; 2, 20, 14)$	1805	$90^{19} 95^1$	380	0.000
$CA(256; 2, 17, 16)$	$\rightarrow CA(255; 2, 17, 15)$	256	$15^{16} 16^1$	272	0.000
$CA(289; 2, 18, 17)$	$\rightarrow CA(287; 2, 18, 15)$	578	$32^{17} 34^1$	306	0.010
$CA(361; 2, 20, 19)$	$\rightarrow CA(357; 2, 20, 15)$	1444	$72^{19} 76^1$	380	0.010
$CA(289; 2, 18, 17)$	$\rightarrow CA(288; 2, 18, 16)$	289	$16^{17} 17^1$	306	0.000
$CA(361; 2, 20, 19)$	$\rightarrow CA(358; 2, 20, 16)$	1083	$54^{19} 57^1$	380	0.010
$CA(529; 2, 24, 23)$	$\rightarrow CA(522; 2, 24, 16)$	3703	$154^{23} 161^1$	552	0.020
$CA(361; 2, 20, 19)$	$\rightarrow CA(359; 2, 20, 17)$	722	$36^{19} 38^1$	380	0.000
$CA(529; 2, 24, 23)$	$\rightarrow CA(523; 2, 24, 17)$	3174	$132^{23} 138^1$	552	0.010
$CA(625; 2, 26, 25)$	$\rightarrow CA(617; 2, 26, 17)$	5000	$192^{25} 200^1$	650	0.020
$CA(361; 2, 20, 19)$	$\rightarrow CA(360; 2, 20, 18)$	361	$18^{19} 19^1$	380	0.000
$CA(529; 2, 24, 23)$	$\rightarrow CA(524; 2, 24, 18)$	2645	$110^{23} 115^1$	552	0.010
$CA(625; 2, 26, 25)$	$\rightarrow CA(618; 2, 26, 18)$	4375	$168^{25} 175^1$	650	0.020
$CA(529; 2, 24, 23)$	$\rightarrow CA(525; 2, 24, 19)$	2116	$88^{23} 92^1$	552	0.010
$CA(625; 2, 26, 25)$	$\rightarrow CA(619; 2, 26, 19)$	3750	$144^{25} 150^1$	650	0.020
$CA(729; 2, 28, 27)$	$\rightarrow CA(721; 2, 28, 19)$	5832	$208^{27} 216^1$	756	0.020
$CA(529; 2, 24, 23)$	$\rightarrow CA(526; 2, 24, 20)$	1587	$66^{23} 69^1$	552	0.010
$CA(625; 2, 26, 25)$	$\rightarrow CA(620; 2, 26, 20)$	3125	$120^{25} 125^1$	650	0.010
$CA(729; 2, 28, 27)$	$\rightarrow CA(722; 2, 28, 20)$	5103	$182^{27} 189^1$	756	0.020
$CA(529; 2, 24, 23)$	$\rightarrow CA(527; 2, 24, 21)$	1058	$44^{23} 46^1$	552	0.010
$CA(625; 2, 26, 25)$	$\rightarrow CA(621; 2, 26, 21)$	2500	$96^{25} 100^1$	650	0.020
$CA(729; 2, 28, 27)$	$\rightarrow CA(723; 2, 28, 21)$	4374	$156^{27} 162^1$	756	0.020
$CA(529; 2, 24, 23)$	$\rightarrow CA(528; 2, 24, 22)$	529	$22^{23} 23^1$	552	0.010
$CA(625; 2, 26, 25)$	$\rightarrow CA(622; 2, 26, 22)$	1875	$72^{25} 75^1$	650	0.020
$CA(729; 2, 28, 27)$	$\rightarrow CA(724; 2, 28, 22)$	3645	$130^{27} 135^1$	756	0.020
$CA(625; 2, 26, 25)$	$\rightarrow CA(623; 2, 26, 23)$	1250	$48^{25} 50^1$	650	0.020
$CA(729; 2, 28, 27)$	$\rightarrow CA(725; 2, 28, 23)$	2916	$104^{27} 108^1$	756	0.020
$CA(841; 2, 30, 29)$	$\rightarrow CA(835; 2, 30, 23)$	5046	$168^{29} 174^1$	870	0.030
$CA(625; 2, 26, 25)$	$\rightarrow CA(624; 2, 26, 24)$	625	$24^{25} 25^1$	650	0.020
$CA(729; 2, 28, 27)$	$\rightarrow CA(726; 2, 28, 24)$	2187	$78^{27} 81^1$	756	0.020
$CA(841; 2, 30, 29)$	$\rightarrow CA(836; 2, 30, 24)$	4205	$140^{29} 145^1$	870	0.030
$CA(729; 2, 28, 27)$	$\rightarrow CA(727; 2, 26, 25)$	1458	$52^{27} 54^1$	756	0.020
$CA(841; 2, 30, 29)$	$\rightarrow CA(837; 2, 28, 25)$	3364	$112^{29} 116^1$	870	0.030
$CA(961; 2, 32, 31)$	$\rightarrow CA(955; 2, 30, 25)$	5766	$180^{31} 186^1$	992	0.040

Table 5 summarizes the results from this experiment. The column 1 shows the CAs taken from experiment 1 as inputs to create the inputs for this new experiment. The column 2 shows the new CAs. The maximum number of wild cards identified in each new CA, using **wcBBCA**, is shown in column 3. The distribution of the wild cards is presented in column 4. Finally, the number of iterations (as defined in the experiment 1) and the time (in seconds), are shown in the last two columns. In this experiment, there was no reduction in the size of the CAs that were constructed by the algebraic method *TimesCA*.

Again, the performance of the algorithm **wcBBCA** is good. We can see that in the results shown in **Table 5**, where the time reported for **wcBBCA** to find the wild card profile was only of a few seconds, even though the matrices have k^2 columns and almost $2 \cdot N$ rows, in comparison with the CAs of the previous experiment. Moreover, the number of iterations performed by the exact approach is small, in comparison with the search space defined for the problem.

4.3 Experiment 3: Diversity in the Wild Card Profiles

We end the analysis of the algorithm **wcBBCA** in this section. Here, we present how different wild card profiles can be generated according to the input matrix given to **wcBBCA**.

The methodology followed in this last experiment was based in four steps. The first step consists in reorganizing the initial matrix of the CA(42; 2, 8, 6) such that $\mathcal{C}(0) < \mathcal{C}(1) < \dots < \mathcal{C}(5)$, where $\mathcal{C}(\mathbf{s})$ is a function that counts the number of times that the symbol \mathbf{s} appears in the matrix of the CA. In the second step, the CA(42; 2, 8, 6) was subject of a set of random changes in the symbols of its matrix. After that, the new matrix was transformed into a CA using the Simulated Annealing approach reported in [14]. Finally, the wild card profile of the new CA(42; 2, 8, 6) was determined using **wcBBCA**. The methodology was repeated a total of 36 times, each of which a different profile for CA(42; 2, 8, 6) was produced. The different wild card profiles of CA(42; 2, 8, 6) are shown in **Table 6a-e**. There, the column 1 shows the total number of wild cards and column 2 shows the distribution. The wild card profiles are ordered according to the total number of wild cards in the matrix.

We compared the wild card profiles presented in **Table 6**, with the wild card profiles of the two matrices provided by PhD Charles Colbourn¹ for the CA(42; 2, 8, 6). The two original matrices have the profile distributions $0^5 2^3$ and $0^2 1^6$, respectively. These profiles only contain 6 wild cards, our results report that some matrices that are CA(42; 2, 8, 6) have up to 12 wild cards. This last result combined with algebraic methods such as the one reported by Colbourn in [4], show the necessity of an approach to identify the maximum number of wilds cards in a CA.

¹ Private communication.

Table 5. Wild card profiles of the new CAs that were created by the algebraic method reported by Colbourn, et al. [4] using as input the CAs shown in column 1 of **Table 4**

Initial Matrix	CA with TimesCA	Profile		wCBBCA	Time (secs.)
		Total WC	WC Distribution	Iter.	
$CA(48; 2, 8, 6)$	$CA(90; 2, 63, 6)$	770	$12^{49}13^{14}$	3906	0.010
	$CA(94; 2, 64, 6)$	784	$12^{49}13^{14}14^1$	4032	0.010
$CA(62; 2, 9, 6)$	$CA(118; 2, 80, 6)$	2272	$28^{64}30^{16}$	6320	0.010
	$CA(122; 2, 81, 6)$	2304	$28^{64}30^{16}32^1$	6480	0.020
$CA(78; 2, 10, 6)$	$CA(150; 2, 99, 6)$	4806	$48^{81}51^{18}$	9702	0.030
	$CA(154; 2, 100, 6)$	4860	$48^{81}51^{18}54^1$	9900	0.040
$CA(63; 2, 9, 7)$	$CA(119; 2, 80, 7)$	1136	$14^{64}15^{16}$	6320	0.020
	$CA(124; 2, 81, 7)$	1152	$14^{64}15^{16}16^1$	6480	0.020
$CA(79; 2, 10, 7)$	$CA(151; 2, 99, 7)$	3204	$32^{81}34^{18}$	9702	0.040
	$CA(156; 2, 100, 7)$	3240	$32^{81}34^{18}36^1$	9900	0.040
$CA(117; 2, 12, 7)$	$CA(227; 2, 143, 7)$	11528	$80^{121}84^{22}$	20306	0.140
	$CA(232; 2, 144, 7)$	11616	$80^{121}84^{22}88^1$	20592	0.140
$CA(80; 2, 10, 8)$	$CA(152; 2, 99, 8)$	1602	$16^{81}17^{18}$	9702	0.040
	$CA(158; 2, 100, 8)$	1620	$16^{81}17^{18}18^1$	9900	0.040
$CA(118; 2, 12, 8)$	$CA(228; 2, 143, 8)$	8646	$60^{121}63^{22}$	20306	0.120
	$CA(234; 2, 144, 8)$	8712	$60^{121}63^{22}66^1$	20592	0.140
$CA(164; 2, 14, 8)$	$CA(320; 2, 195, 8)$	23530	$120^{169}125^{26}$	37830	0.360
	$CA(326; 2, 196, 8)$	23660	$120^{169}125^{26}130^1$	38220	0.360
$CA(119; 2, 12, 9)$	$CA(229; 2, 143, 9)$	5764	$40^{121}42^{22}$	20306	0.130
	$CA(236; 2, 144, 9)$	5808	$40^{121}42^{22}44^1$	20592	0.130
$CA(165; 2, 14, 9)$	$CA(321; 2, 195, 9)$	18824	$96^{169}100^{26}$	37830	0.370
	$CA(328; 2, 196, 9)$	18928	$96^{169}100^{26}104^1$	38220	0.360
$CA(249; 2, 17, 9)$	$CA(489; 2, 288, 9)$	60704	$210^{250}217^{32}$	82656	1.090
	$CA(496; 2, 289, 9)$	60928	$210^{256}217^{32}224^1$	83232	1.140
$CA(120; 2, 12, 10)$	$CA(230; 2, 143, 10)$	2882	$20^{121}21^{22}$	20306	0.130
	$CA(238; 2, 144, 10)$	2904	$20^{121}21^{22}22^1$	20592	0.140
$CA(166; 2, 14, 10)$	$CA(322; 2, 195, 10)$	14118	$72^{169}75^{26}$	37830	0.370
	$CA(330; 2, 196, 10)$	14196	$72^{169}75^{26}78^1$	38220	0.360
$CA(250; 2, 17, 10)$	$CA(490; 2, 288, 10)$	52032	$180^{250}186^{32}$	82656	1.190
	$CA(498; 2, 289, 10)$	52224	$180^{256}186^{32}192^1$	83232	1.130
$CA(167; 2, 14, 11)$	$CA(323; 2, 195, 11)$	9412	$48^{169}50^{26}$	37830	0.370
	$CA(332; 2, 196, 11)$	9464	$48^{169}50^{26}52^1$	38220	0.040
$CA(251; 2, 17, 11)$	$CA(491; 2, 288, 11)$	43360	$150^{256}155^{32}$	82656	1.260
	$CA(500; 2, 289, 11)$	43520	$150^{256}155^{32}160^1$	83232	1.240
$CA(283; 2, 18, 11)$	$CA(555; 2, 323, 11)$	62220	$192^{289}198^{34}$	104006	1.730
	$CA(564; 2, 324, 11)$	62424	$192^{289}198^{34}204^1$	104652	1.730
$CA(168; 2, 14, 12)$	$CA(324; 2, 195, 12)$	4706	$24^{169}25^{26}$	37830	0.380
	$CA(334; 2, 196, 12)$	4732	$24^{169}25^{26}26^1$	38220	0.410
$CA(252; 2, 17, 12)$	$CA(492; 2, 288, 12)$	34688	$120^{250}124^{32}$	82656	1.150
	$CA(502; 2, 289, 12)$	34816	$120^{256}124^{32}128^1$	83232	1.260
$CA(284; 2, 18, 12)$	$CA(556; 2, 323, 12)$	51850	$160^{289}165^{34}$	104006	2.000
	$CA(566; 2, 324, 12)$	52020	$160^{289}165^{34}170^1$	104652	1.860
$CA(253; 2, 17, 13)$	$CA(493; 2, 288, 13)$	26016	$90^{256}93^{32}$	82656	1.300
	$CA(504; 2, 289, 13)$	26112	$90^{256}93^{32}96^1$	83232	1.350
$CA(285; 2, 18, 13)$	$CA(557; 2, 323, 13)$	41480	$128^{289}132^{34}$	104006	1.960
	$CA(568; 2, 324, 13)$	41616	$128^{289}132^{34}136^1$	104652	1.970

Table 6. Wild card profiles produced for the CA(42; 2, 8, 6)

(a)		(b)	
WC	Distribution	WC	Distribution
12	$0^6 6^2$	9	$0^3 1^3 3^2$
11	$0^5 1^1 5^2$	9	$0^4 1^1 2^1 3^2$
10	$0^4 1^2 4^2$	9	$0^4 1^2 3^1 4^1$
10	$0^5 1^1 4^1 5^1$	9	$0^5 1^1 3^1 5^1$
10	$0^5 2^1 4^2$	9	$0^5 2^1 3^1 4^1$
10	$0^6 4^1 6^1$	9	$0^5 3^3$

(c)		(d)		(e)	
WC	Distribution	WC	Distribution	WC	Distribution
8	$0^2 1^4 2^2$	7	$0^1 1^7$	6	$0^3 1^4 2^1$
8	$0^3 1^2 2^3$	7	$0^2 1^5 2^1$	6	$0^4 1^2 2^2$
8	$0^3 1^3 2^1 3^1$	7	$0^3 1^3 2^2$	6	$0^4 1^3 3^1$
8	$0^4 1^1 2^2 3^1$	7	$0^3 1^4 3^1$	6	$0^5 1^2 4^1$
8	$0^4 1^2 2^1 4^1$	7	$0^4 1^3 4^1$	6	$0^6 3^2$
8	$0^5 1^1 2^1 5^1$	7	$0^5 1^1 2^1 4^1$	5	$0^3 1^5$
8	$0^5 2^1 3^2$	7	$0^5 1^1 3^2$	5	$0^7 5^1$
8	$0^5 2^2 4^1$	7	$0^5 1^2 5^1$		
8	$0^6 2^1 6^1$				

5 Conclusions

A wild card is a symbol in a matrix that is a Covering Array (or CA($N; t, k, v$)) which can be substituted for another symbol such that the CA does not lose its level of coverage. We presented an algorithm to find the maximum number of wild cards in a CA. The algorithm is based on a branch and bound approach. It may be used to generate rows which can be merged to reduce the size of CAs.

The performance of the exact approach (called **wcBBCA**) was initially tested in 60 CAs that were constructed using the method reported by Bush [1]. The strength of the CAs was $t = 2$ and their alphabets v varied from 6 to 25. Taking into account that a matrix $\mathcal{M} = \text{CA}(N; t, k, v)$ has $N \cdot k$ symbols, our results report that **wcBBCA** identified at least 10% of the symbols as wild cards, in most of the instances analyzed. The time (in seconds) spent by **wcBBCA** in the identification of the profiles was smaller than one second in these cases.

In a second experiment, matrices that were CAs with a larger number of columns and rows, were analyzed using **wcBBCA**. These matrices were produced using an algebraic method; their wild card profile also contained at least 10% of the symbols of the matrices as wild cards. The time spent by **wcBBCA** was of up to 26 seconds. The number of iterations required by **wcBBCA** is considerably smaller than the search space defined by the problem of maximizing the number of wild cards. These results show that **wcBBCA** had a good performance in the identification of wild card profiles for CAs of different sizes.

Finally, we present a simple methodology for generating different wild card profiles for a given $\text{CA}(N; t, k, v)$. The methodology involved the construction of new CAs from a given CA using random changes over it and a local search procedure. If the resulting matrix is not a CA, it is used as the input matrix of an approximated algorithm (which in turn it converted into a CA). Once that the new CA is obtained, its wild card profile is determined using wcBBCA. We tested this methodology in the $\text{CA}(42; 2, 8, 6)$ and constructed 36 different profiles of it. The number of wild cards of this case varies from 6 to 12 wild cards.

References

1. Bush, K.A.: Orthogonal arrays of index unity. *Annals of Mathematical Statistics* 13, 426–434 (1952)
2. Cohen, M.B., Colbourn, C.J., Ling, A.C.H.: Constructing strength three covering arrays with augmented annealing. *Discrete Mathematics* 308, 2709–2722 (2008)
3. Colbourn, C.J.: Strength two covering arrays: Existence tables and projection. *Discrete Mathematics* 308(5-6), 772–786 (2008); Selected Papers from 20th British Combinatorial Conference
4. Colbourn, C.J., Martirosyan, S.S., Mullen, G.L., Shasha, D., Sherwood, G.B., Yucas, J.L.: Products of mixed covering arrays of strength two. *Journal of Combinatorial Designs* 14(2), 124–138 (2006)
5. Forbes, M., Lawrence, J., Lei, Y., Kacker, R.N., Kuhn, D.R.: Refining the in-parameter-order strategy for constructing covering arrays. *Journal of Research of the National Institute of Standards and Technology* 113(5), 287–297 (2008)
6. Grindal, M., Offutt, J., Andler, S.F.: Combination testing strategies: a survey. *Software Testing Verification & Reliability* 15, 167–199 (2005)
7. Hnich, B., Prestwich, S.D., Selensky, E., Smith, B.M.: Constraint models for the covering test problem. *Constraints* 11(2-3), 199–219 (2006)
8. Kuhn, D.R., Reilly, M.J.: An investigation of the applicability of design of experiments to software testing. In: Proceedings 27th Annual NASA Goddard/IEEE Software Engineering Workshop, pp. 91–95 (2002)
9. Kuhn, D.R., Wallace, D.R., Gallo Jr., A.M.: Software fault interactions and implications for software testing. *IEEE Transactions on Software Engineering* 30, 418–421 (2004)
10. Kuliamin, V.V., Petukhov, A.A.: A survey of methods for constructing covering arrays. *Program. Comput. Softw.* 37, 121–146 (2011)
11. Lawrence, J., Kacker, R.N., Lei, Y., Kuhn, D.R., Forbes, M.: A survey of binary covering arrays. *Electronic Journals of Combinatorics* 84, P84 (2011)
12. Nayeri, P., Colbourn, C.J., Konjevod, G.: Randomized Postoptimization of Covering Arrays. In: Fiala, J., Kratochvíl, J., Miller, M. (eds.) IWOCA 2009. LNCS, vol. 5874, pp. 408–419. Springer, Heidelberg (2009)
13. Tassey, G.: RTI, The economic impacts of inadequate infrastructure for software testing. Technical Report 02-3, National Institute of Standards and Technology, Gaithersburg, MD, USA (May 2002)
14. Torres-Jimenez, J., Rodriguez-Tello, E.: Simulated annealing for constructing binary covering arrays of variable strength. In: Proceedings of the CEC 2010, pp. 4102–4109 (2010)

Intelligent Learning System Based on SCORM Learning Objects

Liliana Argotte¹, Julieta Noguez², and Gustavo Arroyo¹

¹ Instituto de Investigaciones Eléctricas, Reforma 113, Col. Palmira,
62490, Cuernavaca, Morelos, México
{lagotte,garroyo}@ie.org.mx

² Tecnológico de Monterrey, Campus Ciudad de México,
14380, México, D.F., México
jnoguez@itesm.mx

Abstract. This paper shows the creation of the adaptive SCORM sequencing models, taking advantage of the latest developments offered by the artificial intelligence field, to provide the best choice to the student, based in learning objects, using a tutor model in self learning. The Tutor uses decision networks also called influence diagrams, to improve the development of resources and learning materials in a learning content management system, to offer students the best pedagogical decision according to their performance. The intelligent learning system is validated in an online environment. The results of the evaluation process in undergraduate engineering courses are encouraging because they show improvements in student's learning who used this approach, compared to those who did not use it. The paper also shows the potential application of this learning approach for power system's operators.

Keywords: adaptive learning, learning objects, tutor model, tutorial systems.

1 Introduction

The use of technology in learning helps to create a situation learner-centered education that promotes self learning and development of their critical and creative thinking. E-Learning environment is the use of Internet technologies that provide a wide range of solutions for knowledge [1]. E-Learning systems facilitate learning without constraints of time and space, characteristics that affect any organization to extend the option of learning to a greater number of workers and enter new training models. Learning objects (LOs) are emerging, as an effort to create educational components or modules that can be reused in different learning contexts. In education and business training, the use of e-Learning platforms and the design of online courses to their own educational technologies have increased significantly. The use of LOs in e-Learning platform has allowed us to give knowledge, provide information and guide learning activities in the materials.

The Sharable Content Object Reference Model (SCORM) is the standard for e-Learning more used in the world for the development of courses made up of

independent units of information residing in a repository and can be seen on any platform that is also compliant with this standard [2]. The usage and effectiveness of learning using SCORM is the purpose of this work, because you can define different paths for a student to learn also called sequencing models) and we solve as well the space and time situation in a safe training environment based on a reliable tool. The modeling adaptive sequence SCORM proposed in this work), enables the tutor or instructor to add complexity to the models of sequence or different paths of learning resources in less time and promotes improvements in the student's learning with the use of these models.

2 SCORM Sequencing

SCORM sequencing provides the necessary tools to developers of E-learning Courses so they can be able to create complex designs that can adapt it selves to student's individual learning needs, consistently applied in sequencing capabilities that offer the following models [3]:

1. Tracking or monitoring model: the values of this model are used to control the behavior sequence. For every learning attempt made for a student in a certain activity, it will have data on the status of the activity associated with it.
2. Sequencing definition model: sequence uses a model based on rules. One activity can apply a set of zero or more sequencing rules and these are evaluated at specific times during the various sequencing behaviors.
3. Execution model: it is used by LOs to communicate information about a student's interaction) with that content object (for example state or grades).

SCORM neither considers nor excludes the artificial intelligence-based sequencing, its use in contexts involving other intelligent agents is not prohibited either. However, it does not define the roles of these other agents or sequencing behaviors that arise as a result of the involvement of such agents. The navigation model provides flexibility for SCORM content developers to provide navigation controls to determine whether the student will have the option or not to navigate freely in the course content. However, this can also be a disadvantage by not being clear about the rules of navigation for SCORM content developers.

3 Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITS) are interactive learning environments that have the ability to adapt them selves to a specific student during the teaching process [4]. In general, the adaptation process can be described by three phases: (i) getting the information about the student, (ii) processing the information to initialize and update a student's model), and (iii) using the student's model to provide the adaptation.

An ITS is a system that has as main objective to reproduce the behavior of a human tutor that can adapt it self to the rhythm of student's learning [5]. According to

[6] the four main components of an ITS are the student's model, pedagogical model, the knowledge domain model and communication model. Unlike Woolf, other authors such as [7] proposed as the fifth component the expert model, which was previously thought to be within the knowledge domain model.

4 Sequencing and Navigation in ITS

These ITSs did not emerge to be used in the Web; however the potential of the Web in distance learning has led to the implementation of projects and studies to develop ITSs based on that platform. According to [8], one of the techniques more relevant to achieve this, is the use of intelligent sequence of knowledge, involving the adaptive selection of the following item to learn, using the student's model and knowledge about the learning material. The sequence of the courses is a well established technology in the field of ITSs that creates a course individualized for each student, selecting dynamically the optimal operation of instruction (presentation, example, question or problem) at any time.

For their simplicity and effectiveness the techniques of Artificial Intelligence used predominantly in ITS to create the student's model are the Bayesian network and the decision networks or influence diagrams, both classified in Probabilistic Graphical Models (PGMs). According to Koller, the PGMs modelate the uncertainty with probabilistic variables related by dependencies that are expressed in the form of acyclic graphical elements [9].

4.1 Using Probabilistic Graphical Models for Adaptive Sequence

As mentioned above, the sequencing of LOs in SCORM is a rule's-based model, where the instructor plays a key role in establishing previously the different ways of learning that a student can take. This does not guarantee the effectiveness of learning, by not considering all the possible paths of learning and adaptation for specific learning needs of each student, as a solution for this problem, is suggested in this work the use of the MGP tools to add intelligence and adaptation to existing SCORM sequencing model. The MGP (Koller, Friedman, Getoor, and Taskar 2007) in Artificial Intelligence are a powerful mechanism for adaptation to the needs of a student's learning. From the set of techniques that provide this type of models the ones proposed in this work to be used are the decision networks (also called influence diagrams). In its most general form a decision network represents information about the current state of the tutor, possible actions, the state that will result from the action of the tutor and the usefulness of the resulting state (Russell and Norvig 2008). Combining utility theory with probability theory allows a tutor to select actions that maximize their expected performance.

4.2 Using Decision Networks in the Sequencing and Navigation of Los

Decision networks are a simple formalism to express and solve decision problems and therefore there is a direct application to the problem of sequence of LOs, where the

result of the propagation of the evidences in this network would provide the best pedagogical action followed by the student, in other words the sequence of LOs suitable for each student. Being also the decision networks an extension of Bayesian networks is still inferring and diagnosing the current cognitive state of the student, making possible the sequencing and adaptive navigation. However, to actually achieve an adaptive learning situation, must be taken into consideration students' prior knowledge in LO and propagate that evidence with the current cognitive state. This can be achieved considering the "time factor" to create a dynamic decision network.

5 Intelligent Learning System Based on Learning Objects

The model proposed in this work has been called Intelligent Learning System or SI-APRENDE. This system provides the Tutor Model of an ITS to establish an adaptive sequence and navigation of LOs. They were used dynamic decision networks to select the pedagogical action that best suits the situation of each student's learning; one of the main advantages of decision networks is precisely the combination of the theories of probability, utility and decision.

The decision network used in SI-APRENDE manages the uncertainty associated with the status of a student's knowledge of LO by the conditional probabilities that the expert should initially provide. The ability of inference which has a decision network is another of its advantages. It has observable variables resulting from the student's interaction with the system, such as the progress of the activity or the satisfaction of the objective and can also makes predictions based on evidence of knowledge of LO. The utility function provided by the application domain expert ensures that the Tutor Model makes the best decision.

5.1 General Architecture of the System

The architecture of the intelligent learning system SI-APRENDE is designed in three layers: presentation, logic and data. Each one is described below and is represented in (Fig. 1):

- a) Presentation Layer: contains the graphical interface that connects the system with the user in a Web environment. This user interface uses AJAX components that facilitate its development in .NET platform, as well as a simple and friendly interaction with the user.
- b) Business Logic Layer: referring to the business rules and validations required in the application, such as educational activities of the tutor model and monitoring of learning activities related to knowledge, skills, goal achievement, activity's satisfaction and progress. In this layer other components are associated with other components such as: user management, monitoring of learning activities, sequencing and navigation, which are inherent components of the same application SI-APRENDE; PGMs such as Elvira, Carmen or any other application that can create nodes and relationships and make inferences and propagation with evidence; tagger Software of LOs, i.e. any environment or application that allows creating labels and SCORM packages.

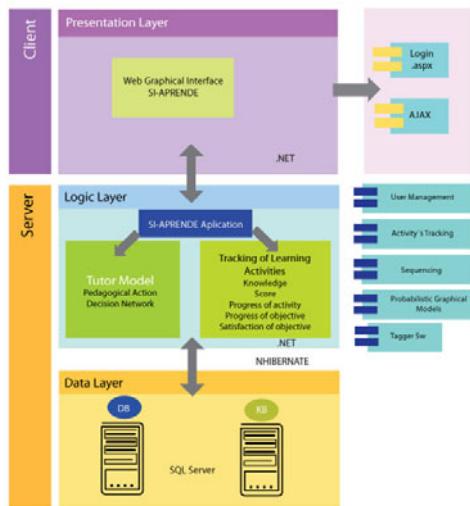


Fig. 1. Architecture of Adaptive Intelligent Environment SI-APRENDE

c) Data Layer, easy access and manipulation of information stored in databases. The technology to use is: Hibernate .NET (NHibernate). Specifically, it contains access to the repository of SCORM learning objects database and SI-APRENDE system. Manager is proposed as a relational database to Microsoft SQL Server.

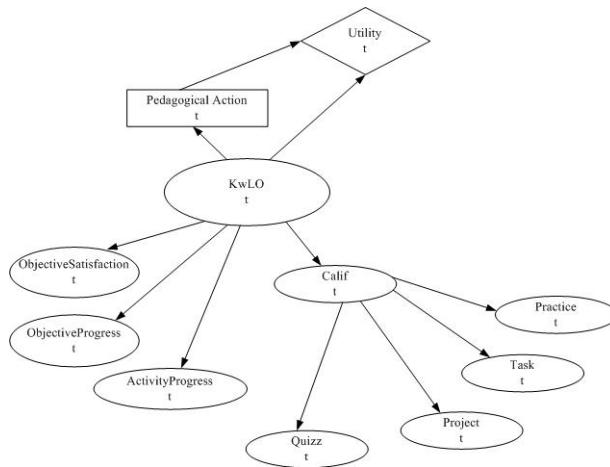
5.2 Domain Knowledge Model

The domain knowledge model has a structure where a course has different topics, each topic in turn has various subtopics and each subtopic is made up of one or more LOs containing the concept or knowledge units required for this item. In this way the generic base can be applied in any domain of course's application.

5.3 Tutor Module

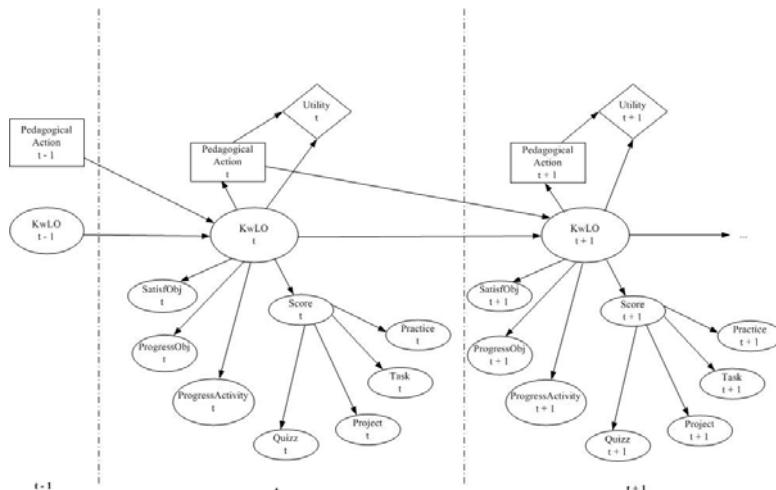
An important aspect of the tutor model, also known as instructional or educational model, is the sequence of the learning material presented to each student. We present in this paper a tutor model with a decision network that selects the best pedagogical action for each student.

This decision network consisted of the following random variables (see Fig. 2): knowledge of LO, satisfaction of the objective, progress of the objective, progress of the activity, score of LO, quiz, project, task and practice. KwLO random node has four values: Very Good, Good, Enough and Not Enough. The decision node considers the pedagogical actions that will be evaluated according to the utility function to select the best one, but have identified four possible pedagogical actions have been identified: LO challenge, Next LO, Repeat refined and Repetition of LO.

**Fig. 2.** Decision Network

The calibration of the decision network is given by experts in the domain. The utility's table entries are based on the teacher's experience according to the best over all possible pedagogical actions according to the given knowledge state and student's interaction.

The Fig. 3 shows the same decision network described in the previous figure now considering the evidence of LO's Knowledge random node in the previous time as well as pedagogical action selected from the previous time, making it a dynamic decision network.

**Fig. 3.** Dynamic Decision Network

This dynamic model allows the user to take the accumulated evidence from previous experiments with the results of the current knowledge object. The aim is to update the knowledge objects and take the best pedagogic action.

6 Case Study

It was chosen as case study the model's application to undergraduate courses such as Mathematics II, Electricity and Magnetism, and Introduction to Physics. The instructors designed four LOs of a specific topic for each course, which were integrated with the standard SCORM technique to leave them available in the SI-APRENDE. It was planned that all students started at the same LO of their respective courses. The following actions were designed to achieve pedagogical adaptation to the learning needs of each student:

- LO's challenge or higher difficulty level: when the student gets a good grade (or its numeric equivalent $9 \leq 10$ on a scale of 1 to 10).
- Next LO with the same difficulty level as the previous one: when the student gets a grade in the following range $7 \leq 9$ on a scale of 1 to 10.
- Refined repetition or lower difficulty level, when the student scores sufficiently (or its numeric equivalent $6 \leq 7$ on a scale of 1 to 10).
- LO's repetition, when the student scores noncredit able (or its numeric equivalent $1 \leq 6$ on a scale of 1 to 10).

This set of pedagogical actions that were based only in the score, was initially defined by the instructors. However, adaptive navigation in a SCORM environment is more complex than that, the pedagogical action taken by the SI-APRENDE results of the propagation of a dynamic decision network, than besides the LO's score, gets other variables like learning activities and objective's progress as well as learning objective's satisfaction for each student.

7 Description of the Evaluation Process

The evaluation process was conducted in three groups of undergraduate courses: Engineering Mathematics II, Electricity and Magnetism and, Introductory Physics of Tecnológico de Monterrey, Campus Ciudad de Mexico, taught by three instructors recognized for their extensive experience in teaching and in the study area. It had a total population of 58 students ($N = 58$) distributed as shown in Table 1.

Table 1. Total population of Case Study

	Focus Group	Control Group
Math II	15	15
Electricity and Magnetism	9	9
Introduction to Physics	5	5
	NFocus= 29	NControl= 29

The total population's division by the instructors was random, to form two groups: Focus and Control. The focus group consisted of students who used the SI-APRENDE during a specific period of time, while the control group had no access to the system and used the resources provided by the instructor from the Blackboard platform (with the same amount of learning resources in the SI-APRENDE system). The three focus groups of each course were formed by students of various disciplines or careers. They were also heterogeneous in the level of subject's knowledge, skills or even interests.

A pre-test was applied to all students in the class, grading on a scale from 0 to 100. The learning objective was the same for both focus and control group, yet the way to reach that goal was different, in the case of focus group were used the LOs contained in the SI-APRENDE system, meanwhile in the control group the activities were on paper, Blackboard or taught one on one according to course's traditional planning. After two weeks using the SI-APRENDE system, instructors applied a final test (post-test) to both focus group and control group on the same subject that the pre-Test.

The Figures 4, 5 and 6 show learning gains for each group, obtained from the average of the group in the Pre-Test, focus groups being those with higher learning gain.

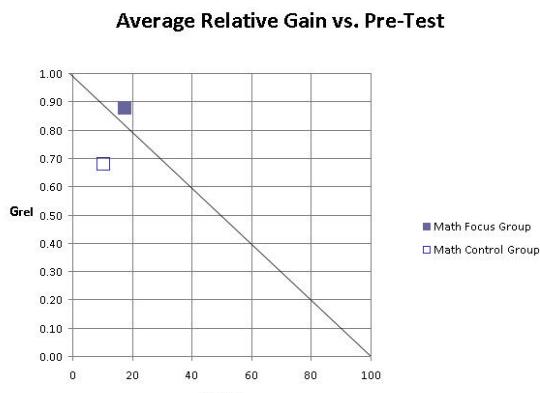


Fig. 4. Math Group

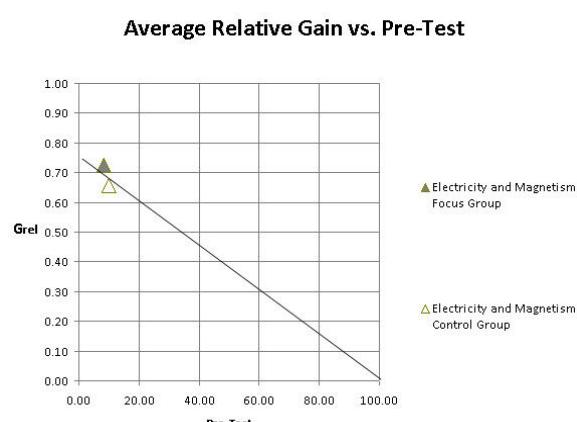


Fig. 5. Electricity and Magnetism Group

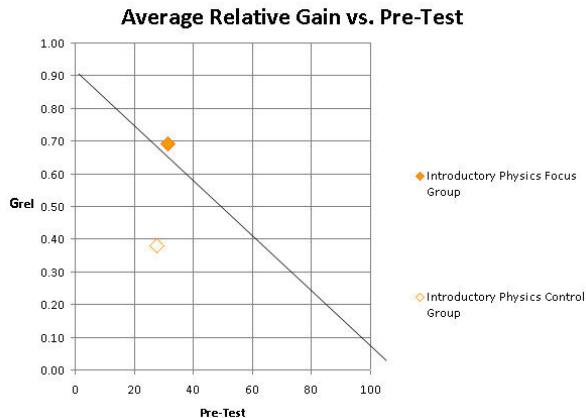


Fig. 6. Introductory Physics

8 Results and Discussion

All focus groups had higher learning gains compared to control groups. Higher learning gain is observed in the Introductory Physics group, then the Math group and finally in the Electricity and Magnetism group. Students in the course of Electricity and Magnetism, in both focus and control groups, received an instructor's personalized assistance, so that the differential in the learning relative gain was considerably reduced compared to the Physics and the Math's groups. However in this subject, the focus group also had a higher learning relative gain compared to the control group.

SI-APRENDE system allowed each student to have their own learning process, in different time and LO's sequence, depending on prior knowledge and progress achieved, both evaluated by the Tutor of this system.

9 Model Validation in Electrical Sector

SI-APRENDE is being validated in CFE (Comisión Federal de Electricidad – the National Electric Utility in Mexico). The main goal of the intelligent learning system is to certify operators in knowledge, skills, expertise, abilities and attitudes for power system's operation. Because of the fact that the operators can acquire knowledge in different ways or with different paths of learning, the tutor model of the SI-APRENDE system selects the sequence of the learning material presented to each operator. The adaptive sequence is represented as a decision network that selects the best pedagogical action for each specific operator. The decision network represents information about the tutor's current state, their possible actions, the state resulting from the action of the tutor and the usefulness of the resulting state.

10 Conclusions

The contribution of this work was to achieve the convergence between two existing tools for sequence and navigation of learning materials, such as SCORM and ITS, adding intelligence with the use of PGMs tools such as decision networks and dynamic decision networks, achieving an intelligent learning system adaptive to the situation of each student's learning.

The implementation of the intelligent learning system in the National Electrical Sector provides a more dynamic and interactive training to the Operators of Power Plants, where the employees really experience the acquisition and transfer of skills and knowledge.

References

1. Rosenberg, M.J.: E-Learning. In: Strategies for Delivering Knowledge in the Digital Age, vol. 28. Mc Graw-Hill (2001)
2. Advanced Distributed Learning: SCORM 2004 4th Edition Overview Version 1.0. ADL-Overview, Washington (2009)
3. Advanced Distributed Learning: SCORM 2004 4th Edition Sequencing and Navigation (SN) Version 1.0. ADL-SN, Washington (2009)
4. Peachey, D.R., McCalla, G.I.: Using planning techniques in intelligent tutoring systems. International Journal Man-Machines Studies, 77–98 (1986)
5. Frasson, C., Mengelle, T., Aimeur, E., Gouardéres, G.: An Actor-Based Architecture for Intelligent Tutoring. In: Lesgold, A.M., Frasson, C., Gauthier, G. (eds.) ITS 1996. LNCS, vol. 1086, pp. 57–65. Springer, Heidelberg (1996)
6. Woolf, B.P.: Building Intelligent Interactive Tutors: Student-centered Strategies for Revolutionizing E-learning. Morgan Kaufmann (2008)
7. Beck, J., Stern, M., Haugsjaa, E.: Applications of AI in Education. Crossroads, 11–15 (1996)
8. Brusilovsky, P.L.: A framework for intelligent Knowledge Sequencing and Task Sequencing. In: Frasson, C., McCalla, G.I., Gauthier, G. (eds.) ITS 1992. LNCS, vol. 608, pp. 499–506. Springer, Heidelberg (1992)
9. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
10. Koller, D., Friedman, N., Getoor, L., Taskar, B.: Graphical Models in a Nutshell. In: Getoor, L., Taskar, B. (eds.) Introduction to Statistical Relational Learning (2007)
11. Russell, S., Norvig, P.: Inteligencia Artificial. Un enfoque moderno, 2nd edn. Pearson Prentice-Hall (2008)

A Weighted Profile Intersection Measure for Profile-Based Authorship Attribution

Hugo Jair Escalante¹, Manuel Montes-y-Gómez^{2,3}, and Thamar Solorio³

¹ Universidad Autónoma de Nuevo León,
San Nicolas de los Garza, 66450, N. L., Mexico

² Instituto Nacional de Astrofísica, Óptica y Electrónica,
Tonantzintla, 72840, Puebla, Mexico

³ University of Alabama at Birmingham,
Birmingham, AL, 35294, USA

hugojair@gmail.com, mmontesg@inaoep.mx, solorio@cis.uab.edu

Abstract. This paper introduces a new similarity measure called weighted profile intersection (WPI) for profile-based authorship attribution (PBAA). Authorship attribution (AA) is the task of determining which, from a set of candidate authors, wrote a given document. Under PBAA an author's profile is created by combining information extracted from sample documents written by the author of interest. An unseen document is associated with the author whose profile is most similar to the document. Although competitive performance has been obtained with PBAA, the method is limited in that the most used similarity measure only accounts for the number of overlapping terms among test documents and authors' profiles. We propose a new measure for PBAA, WPI, which takes into account an inter-author term penalization factor, besides the number of overlapping terms. Intuitively, in WPI we rely more on those terms that are (frequently) used by the author of interest and not (frequently) used by other authors when computing the similarity of the author's profile and a test document. We evaluate the proposed method in several AA data sets, including many data subsets from Twitter. Experimental results show that the proposed technique outperforms the standard PBAA method in all of the considered data sets; although the baseline method resulted very effective. Further, the proposed method achieves performance comparable to classifier-based AA methods (e.g., methods based on SVMs), which often obtain better classification results at the expense of limited interpretability and a higher computational cost.

1 Introduction

Recent advances on information technology have motivated the generation of huge amounts of data. For example, users of social networks (e.g., Facebook¹) and microblogging websites (e.g., Twitter²) generate millions of texts every day.

¹ <http://www.facebook.com/>

² <http://www.twitter.com/>

Analyzing such information has important benefits (e.g., anticipation of terrorist attacks, cyber-crime detection, tracking of marketing trends, and opinion mining), thus posing a major challenge to the Natural Language Processing and Artificial Intelligence communities, in terms of both efficiency and performance.

Authorship attribution (AA) is the task of identifying whom, from a set of candidates, is the author of a given document [19]. AA applications include spam filtering [2], fraud detection, computer forensics [10], cyber bullying [14] and plagiarism detection [16]. Because of its wide applicability, mainly in security aspects, the development of automated AA techniques has received much attention recently [19]. Many AA methods have been proposed so far, some more complex than others. One of the most used approaches nowadays is that based on author profiles [5,11].

In profile-based authorship attribution (PBAA) an author's profile is created by combining information extracted from sample documents written by the author of interest [5,19]. An unseen document is associated with the author whose profile is most similar to the document. Figure 1 depicts the PBAA formulation. The PBAA approach is highly efficient, and it has the additional benefit that the generated profiles can reveal helpful and interpretable information about the writing style of authors. Competitive performance has been obtained with PBAA [5,7,8,9,11,19], however, the method is limited in that the most used similarity measure only accounts for the number overlapping terms among test documents and authors' profiles.

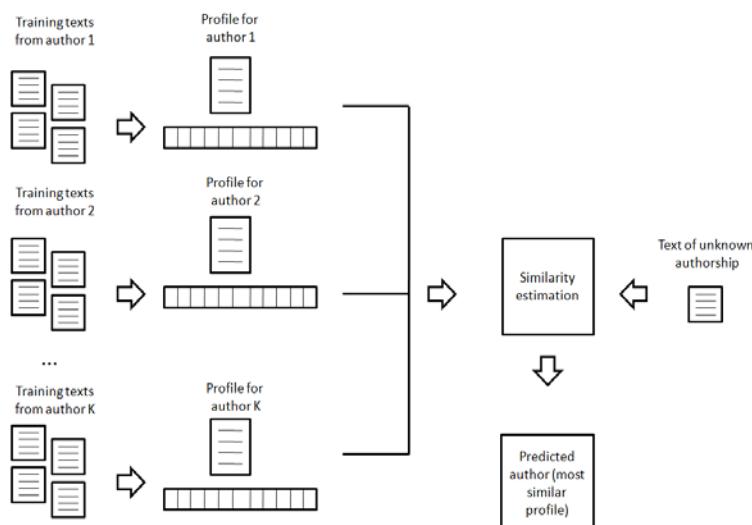


Fig. 1. Diagram of the profile-based authorship attribution approach. Documents from each author are combined to obtain a profile (e.g., a prototypical vector or document). Unseen documents are compared to every profile. The document is associated with the author of the most similar profile. Figure inspired from [19].

In this paper we propose a new similarity measure for PBAA, the weighted profile intersection (WPI) measure, which takes into account an inter-author term penalization factor, besides the number of overlapping terms. Intuitively, the WPI measure gives a high weight to terms that are used by less authors and it assigns a low weight to terms that are shared across profiles for different authors. The term weights are considered for weighting the number of overlapping terms among profiles and documents.

We evaluate the proposed method in several AA data sets, including many subsets from Twitter (a challenging corpus because of the short length of its texts). Experimental results confirm that the baseline PBAA method resulted very effective for AA. However, the proposed technique outperforms the standard PBAA method in all of the considered data sets. Also, we compare the performance of the proposed method to that obtained by several classifiers (under classifier-based AA). We found that the proposed method achieves comparable performance to that of classifier-based AA methods. However, one should note that PBAA methods are advantageous over classifier-based methods because they are based on more interpretable representations and are more efficient.

The rest of the paper is organized as follows. The next section reviews related work on AA. Section 3 presents the proposed PBAA method. Section 4 describes the experimental evaluation of the proposed method. Section 5 presents conclusions and outlines future work.

2 Related Work

The AA task can be faced as one of single-label multiclass classification, with as many classes as candidate authors. However, unlike usual text categorization tasks, where the core problem is modeling the thematic content of documents [18], the goal in AA is modeling authors writing style [19]. Hence, most of the work in AA proposes the use of document representations that are believed to be topic-free and geared towards revealing the writeprint of authors [7,9,12,19]. These features have been loosely called stylometric features, due to the early work on stylometry. Stylometric features typically include character, lexical, syntactical, grammatical and semantic features [7,9,19]. Nevertheless, despite the fact that elaborated stylometric features have been used for AA, representations based on character n-grams or words are predominant [1,4,6,7,8,9,12,19]. In the classifier-based approach several standard learning algorithms have been evaluated, including the popular support vector machine (SVM) classifiers [4,6] neural networks [20], Bayesian classifiers [1] and decision trees [9].

Another popular formulation for AA is the so called profile-based AA (PBAA) approach [7,8,9,11,19]. Under this approach the information of sample documents from each author is combined for building author profiles (i.e., a prototypical document or vector). An unseen document is associated with the author whose profile is most similar to the document. PBAA methods are advantageous over classifier-based approaches in that they are easy to implement, efficient to apply, scale well to large numbers of documents and authors, and they have attractive

interpretability properties (e.g., we can know what terms are more important for each author, and we can also compare profiles of different authors).

In the most used PBAA method called *common n-grams* (CNG) a profile is the set of the top- L more frequent words used by the author in their sample documents [8,19]. A test document and a profile are compared by a normalized frequency of overlapping terms. CNG, introduced by Keselj et al., is perhaps the most used PBAA method [7,8,9,11,19]. Similar profiles are adopted by Frantzeskou et al. although they propose a simplified similarity measure called: simplified profile intersection (SPI) [5]. SPI is simply the un-normalized number of overlapping terms among profiles and test documents. The SPI similarity measure outperformed the similarity measure adopted by Keselj et al. [8], which is slightly more complex. Therefore, in recent AA studies the SPI similarity measure is preferred [11]. This paper proposes an extension to the SPI measure that accounts for an inter-author term-weighting factor besides considering the number of overlapping terms among profiles and test documents. In Section 4 we show that our method outperforms the standard PBAA that uses the SPI measure in a suite of AA data sets, and that it compares favorably with classification-based methods.

3 Proposed Method

The PBAA approach to AA is depicted in Figure 1. In agreement with related work we adopt the CNG approach as base model for developing our PBAA method. Under the CNG method with SPI similarity measure (hereafter CNG-SPI), a profile for an author is the set of the L -most frequent words in the sample documents from that author. Consider a scenario where we have K -candidate authors, let P_1, \dots, P_K denote the profiles for each of the K -authors. When a test document needs to be classified, a profile is obtained for the test document as well. We consider a profile for a test document to be the set of all of the terms appearing in that document. Let T_j denote the j^{th} test document and let $I_j^i = P_i \cap L_j$ be the set of terms that overlap between the profile of the i^{th} author, P_i , and the j^{th} test document, T_j , then the SPI similarity measure is defined as [5]:

$$S_{spi}(P_i, T_j) = |I_j^i| \quad (1)$$

The test document will be assigned to the author's profile with the largest SPI measure.

Formula (1) above only takes into account the raw number of terms in the intersection between profiles. Despite its simplicity, very good results have been reported with the CNG-SPI technique [5,11,19]. However, we think that the CNG method can be improved by adopting a weighted similarity measure.

We propose the weighted profile intersection (WPI) measure, which incorporates a term-weighting factor proportional to the usage of the term across the profiles of the candidate authors. The intuition behind CNG-WPI, is that terms in the intersection I_j^i that are used by a single author (or by a small number of authors) must receive a higher weight, as it is more likely that the intersection

of these terms is indicative of the agreement between profile and test document. On the other hand, terms in I_j^i that appear in the profiles of most of the candidate authors should receive a lower weight, as these terms are likely to appear in many profiles. In agreement with the above arguments we consider the following weighting factor for each term that appears in at least one author profile:

$$w_l = \frac{1}{\sum_{k=1}^K \mathbf{1}_{t_l \in P_k}} \quad (2)$$

where w_l is the weight associated with term t_l and $\mathbf{1}_{t_l \in P_k}$ is an indicator function taking the value 1 when $t_l \in P_k$ is true and 0 otherwise. Therefore, the weight associated to term t_l is inversely proportional to the number of profiles in which the term occurs. This weighting scheme was partially inspired by the well known tf-idf weighting scheme in information retrieval, where the idea is to sink the relevance of terms that occur frequently in most of the documents in the collection while boosting the weight of terms that are rare in the document collection. The motivation behind both weighting schemes is very similar in spirit with the exception that in our approach we do not account for the frequency of the terms explicitly and we penalize the terms according to their presence in the profiles of the authors, not the entire collection of sample documents. One should note that term frequency is considered implicitly in the proposed approach, as we use it as the only criterion to select terms for building the profile for an author.

Besides including a term weighting factor we wanted to reduce the impact of test documents with a large number of terms, which are more likely to have a larger number of overlapping terms in the intersection with the authors' profiles just by chance. Therefore, we define the WPI similarity measure between an author profile P_i and the test document T_j as follows:

$$S_{wspi}(P_i, T_j) = \frac{1}{|I_j^i|} \times \sum_{k=1}^{|I_j^i|} w_{I_j^i} \quad (3)$$

Formula (3) is just the average weight of overlapping terms between the test document and the author profile. By using the average instead of the sum we reduce the influence of the number of terms in the test documents.

The next section reports experimental results in several AA data sets that have been used in previous AA studies. From Formulas (1) and (3) we can see that the only parameter of CNG-SPI and CNG-WPI is L , the number of more frequent terms to consider for building the authors' profiles. We evaluate the performance of both methods with respect to this parameter in the next section as well.

4 Experimental Evaluation

This section reports an experimental evaluation of the proposed CNG-WPI method. We first describe the considered data sets and then we present the experimental results.

4.1 Authorship Attribution Data Sets

We consider several AA data sets that have been used in previous studies [11,15,17]. Table 1 shows some statistics about these data sets³. Our inclusion criteria was to provide a good variety of genre and domains, as well as corpora sizes and number of candidate authors. Five data sets are due to Raghavan et al. (rows 2-6), these documents were collected from the web [17]. The CCAT data set was first used by Stamatatos et al. [6] and then by Plakias et al. [15]. This data set contains documents from news about the same topic written by different authors. Finally, the Twitter data set is a subset of the data set collected by Layton et al. [11]. It contains around 5,000 documents written by 50 authors. Note that this data set is particularly challenging as each document is a tweet of 140 characters or less. With the exception of Twitter, in each of the considered data sets the authors wrote documents in the same topic. Hence, it is expected that the theme of documents does not have an impact in the performance of the considered methods.

Table 1. Data sets considered for experimentation. We show the number of authors, terms, training and test documents for each data set.

Data set	Authors	Terms	Train	Test	Reference
Football	3	8620	52	45	[17]
Business	6	10550	85	90	[17]
Travel	4	11581	112	60	[17]
Cricket	4	10044	98	60	[17]
Poetry	6	8016	145	55	[17]
CCAT	10	15587	500	500	[6]
Twitter	50	26156	4500	500	[11]

Besides the Twitter subset described in Table 1, we performed experiments by generating subsets of different sizes taken from the original corpus of Layton et al. [11]. Specifically, we considered documents in the folder *raw-depth_sample_II*, where there are documents written by 100 different authors with an average of 177 documents per author. Different evaluation subsets were randomly generated with replacement. In each subset 70% of the documents are used for training and the rest for testing.

Based on previous studies we used as terms character n-grams with $n = 3$, where spaces and punctuation marks are considered terms. Character 3-grams have proved to be very effective, this is the most used representation for the AA task [4,6,8,13,15,19]. For the evaluation of the different methods we consider accuracy (percentage of documents assigned to the correct author) as leading evaluation measure.

³ Each data set can be obtained by contacting the authors of the respective references.

4.2 CNG-SPI vs CNG-WPI

In this section we compare the performance of the CNG-SPI and CNG-WPI methods. For these experiments we consider subsets of the Twitter collection, our choice is justified by the fact that this is the more challenging collection in terms of sparsity (each document is a tweet of 140 characters or less [11]) and number of candidate authors (50). Also, this collection contains many documents per author, therefore we can control the number of documents per author and generate multiple subsets for the evaluation of different aspects of CNG-WPI. Furthermore, the generation of multiple subsets allows us to determine whether differences in performance are statistically significant. For this experiment we randomly generated subsets of documents from the Twitter data set. For each subset we randomly chose 50 authors and a different number of documents per author: 20, 50, 100, 150 and 200.

Figure 2 shows the performance obtained with the CNG-WPI method for different values of L (the number of most frequent terms to consider for the profiles). Each result is the average over five runs (using a different subset randomly generated for each run) of the application of CNG-WPI. Rather than using specific values for L we defined it in terms of the size of the vocabulary for each collection. We tried the values of L that represent 10%, 20%, 30%, 40% and 50% of the number of terms in each data set. Then a total of $5 \times 5 \times 5 = 125$ runs were performed, (5-runs, 5-author sizes and 5-values of L).

From Figure 2 we can see that the accuracy is low for all of the settings, although it is much higher than the random-guessing baseline (expected performance of 2%). As expected, the performance of the CNG-WPI method improves as the number of documents per author increases. Regarding the value of L , using the 10% of the vocabulary size seems to be the best option. This can be due to the fact that as more terms are considered, the number of terms shared by different profiles increases, thus reducing the impact of the proposed term-weighting.

Table 2. Improvement of CNG-WPI over CNG-SPI for Twitter subsets. Each result is the average of five runs of the methods in different data subsets. Columns show the percentage of terms from the vocabulary that was used for building profiles (L) and rows show the number of documents per author considered. Light grayed cells show differences that were statistically significant at the 95% level.

Terms	Accuracy Improvement over CNG-SPI				
	10%	20%	30%	40%	50%
20 Docs.	11.25%	7.50%	9.60%	3.80%	5.00%
50 Docs.	5.68%	7.36%	6.56%	4.16%	5.60%
100 Docs.	2.56%	3.35%	2.53%	0.47%	1.90%
150 Docs.	2.47%	1.93%	2.20%	1.53%	1.87%
200 Docs.	-0.20%	1.10%	-0.20%	-1.10%	-0.05%

Table 2 shows the average (over five runs in different data sets) improvement offered by CNG-WPI over CNG-SPI, for different values of L and different

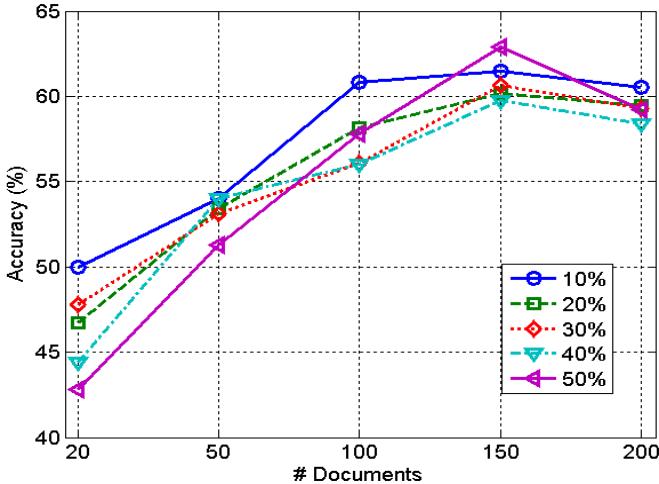


Fig. 2. Performance of CNG-WPI method in the Twitter subsets with 50 authors. Each result is the average of five runs of the method in different data subsets. We show accuracy curves for different values of L , the percentage of most frequent terms (3-grams) that was used for building profiles.

numbers of training documents per author. The differences that were statistically significant at the 95% level are colored in light gray. We used a Wilcoxon signed-rank test as it is the recommended test when comparing classification methods over different data sets [3]. We can see that CNG-WPI outperforms CNG-SPI for most of the considered settings. The improvements are more important for Twitter subsets with a small number of documents per author (e.g., see columns 10 Docs. and 20 Docs. columns). This result is interesting since in real scenarios one usually deals with data sets with a limited number of documents per author. The largest improvement is obtained when L is 10% of the vocabulary. This result is somewhat expected as when less terms are considered, the weighting factor is more important. When 200 documents per author were used, the CNG-SPI method outperformed CNG-WPI for 4 out of the 5 values of L tested. This can be due to the fact that for this number of documents more words tend to co-occur in profiles of many authors, which causes some words to be underestimated by the weight factor in Formula (2). However, the differences in performance in Table 2, row 7 (200 Docs.) are only statistically significant for the columns 20% and 40%.

Summarizing the results from this section, we showed that the proposed method outperforms the CNG-SPI technique. Recall that the experimental results reported in Table 2 comprise a total of 125 runs using different subsets for CNG-SPI and CNG-WPI. While the improvement is not dramatic it is important and evidences the benefits of our approach.

4.3 CNG-SPI vs Classification-based methods

This section reports experimental results of the comparison of CNG-WPI and classification-based approaches. Different classifiers were evaluated using the bag-of-words⁴ representation for documents (using character 3-grams as terms). The considered classifiers are naive Bayes, support vector machine (SVM), neural network (neural), random forest (RF) and 1-nearest neighbor (KNN). Table 3 shows the results of the comparison for the data sets described in Section 4.1.

Table 3. Experimental results of different AA methods for the data sets described in Table 1

	CNG-SPI	CNG-WPI	Naive	SVM	Neural	RF	KNN
Football	91.11%	93.34%	88.89%	91.11%	91.11%	84.44%	77.78%
Business	77.78%	80.00%	82.22%	81.11%	77.78%	71.11%	50.00%
Travel	71.67%	73.33%	70.00%	75.00%	74.00%	75.00%	55.00%
Cricket	88.33%	90.00%	93.33%	98.33%	90.00%	83.33%	70.00%
Poetry	78.18%	85.45%	74.55%	60.00%	65.45%	40.00%	27.27%
CCAT	64.00%	73.60%	73.60%	79.00%	76.80%	73.00%	63.60%
Twitter	53.20%	58.20%	60.80%	52.60%	58.20%	N/A	26.40%
Avg.	74.90%	79.13%	77.63%	76.74%	76.19%	71.15%	52.86%

We can see from this table that the CNG-SPI method is a very effective AA method when compared with classification-based approaches, however, CNG-WPI is more effective. It is worth mentioning that the performances achieved by the different classifiers are comparable to that reported in related works that have used the same data sets [6,11,15,17]. The proposed method outperforms the different classification based approaches in different data sets. CNG-WPI outperforms the naive Bayes classifier in 3 out of 7 data sets and these methods tie in one data set. The proposed method outperforms the neural net classifier in a similar way. CNG-WPI outperforms KNN in all of the data sets, and it outperforms random forest in 6 out of the 7 data sets. It is interesting that CNG-WPI even outperforms an SVM (the most used classifier in classification-based AA [4,15,6]) in 3 out of the 7 data sets. On average (last row in Table 3) the CNG-WPI method obtained the best performance among the considered methods. Giving evidence of its suitability for the AA task.

One should note that though CNG-WPI did not outperform the other methods in all of the considered data sets, PBAA methods are advantageous over classification-based approaches as they are more informative in terms of interpretability. More important, PBAA methods are much more efficient. Also, one should note that we have used only lexical features for representing documents with CNG-WPI. We would like to evaluate the performance of the proposed formulation using other types of features.

⁴ We also performed preliminary experiments with the tf-idf representation for documents, although we found that the performance of most of the considered classifiers was worse than that obtained when the boolean bag-of-words was used.

5 Conclusions

We have described CNG-WPI⁵, a prototype-based authorship attribution method based on a new similarity measure, the weighted profile intersection (WPI). Under PBAA, an author's profile is created by combining information extracted from sample documents written by the author of interest. An unseen document is associated with the author whose profile is most similar to the document. Traditional PBAA methods consider the number of overlapping terms as similarity measure. The proposed similarity measure incorporates a term-weighting factor that accounts for the usage of terms across profiles for different authors. Terms shared by several authors receive a lower weight than those terms that are used by one (or a few authors).

We performed experiments with several data sets previously used for the evaluation of AA methods, including multiple subsets of a Twitter corpus. Experimental results show that the proposed approach outperforms the standard PBAA method for most of the considered settings. The improvement was consistent for several values of L and different data set sizes, evidencing the effectiveness of the proposed method. Furthermore, we compared the performance of the proposed method to that obtained by classification-based AA methods. We found that the proposed method outperforms classifier-based techniques for different data sets. Besides achieving comparable performance, PBAA methods are advantageous over classification-based methods in terms of interpretability and efficiency. Thus showing the suitability of the proposed approach to AA.

Future work directions include the use of the proposed method with other type of features (e.g., syntactical, grammatical or semantic), as well as studying the relationship between L and the number of training documents and between L and the length of documents. Also, we would like to evaluate other term-weighting factors (e.g., inverse-document frequency and related weighting factors) for weighting the intersection of terms in PBAA. Additionally, it would be interesting to evaluate the impact of feature (term) selection in the construction of author profiles for AA.

Acknowledgements. The authors are grateful with the reviewers for providing helpful suggestions to improve the paper as well as interesting ideas that we will pursue as future work. The first author acknowledges the support of PROMEP under grant 103.5/11/4330. This work was supported in part by the CONACYT-Mexico (project no. 134186) and by the European Commission as part of the WIQ-EI project (project no. 269180) within the FP7 People Programme.

References

1. Coyotl-Morales, R.M., Villaseñor-Pineda, L., Montes-y-Gómez, M., Rosso, P.: Authorship Attribution using Word Sequences. In: Martínez-Trinidad, J.F., Carrasco

⁵ The code with the implementation of the proposed method, as well as the preprocessed data used for experimentation are available on demand by contacting the authors.

- Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 844–852. Springer, Heidelberg (2006)
- 2. de Vel, O., Anderson, A., Corney, M., Mohay, G.: Multitopic email authorship attribution forensics. In: Proceedings of the ACM Conference on Computer Security - Workshop on Data Mining for Security Applications, Philadelphia, PA, USA (2001)
 - 3. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
 - 4. Escalante, H.J., Solorio, T., Montes, M.: Local histograms of character n-grams for authorship attribution. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pp. 288–298. Association for Computational Linguistics, ACL (2011)
 - 5. Frantzeskou, G., Stamatatos, E., Gritzalis, S., Chaski, C.E.: Identifying authorship by byte-level n-grams: The source code author profile (SCAP) method. *International Journal of Digital Evidence* 6(1), 1–18 (2007)
 - 6. Houvardas, J., Stamatatos, E.: N-gram Feature Selection for Authorship Identification. In: Euzenat, J., Domingue, J. (eds.) AIMSA 2006. LNCS (LNAI), vol. 4183, pp. 77–86. Springer, Heidelberg (2006)
 - 7. Joula, P.: Authorship attribution. *Foundations and Trends in Information Retrieval* 1(3), 233–334 (2006)
 - 8. Keselj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: Proceedings of the Pacific Association for Computational Linguistics, Halifax, Canada, pp. 255–264 (2003)
 - 9. Koppel, M., Schler, J., Argamon, S.: Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology* 60, 9–26 (2009)
 - 10. Lambers, M., Veenman, C.J.: Forensic Authorship Attribution using Compression Distances to Prototypes. In: Geraarts, Z.J.M.H., Franke, K.Y., Veenman, C.J. (eds.) IWCF 2009. LNCS, vol. 5718, pp. 13–24. Springer, Heidelberg (2009) ISBN 978-3-642-03520-3
 - 11. Layton, R., Watters, P., Dazeley, R.: Authorship attribution for twitter in 140 characters or less. In: Second Cybercrime and Trustworthy Computing Workshop (CTC), pp. 1–8. IEEE, Ballarat (2010)
 - 12. Luyckx, K., Daelemans, W.: Authorship attribution and verification with many authors and limited data. In: Proceedings of the 22nd International Conference on Computational Linguistics, vol. 1, pp. 513–520. ACM Press, Manchester (2008)
 - 13. Luyckx, K., Daelemans, W.: The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing*, pp. 1–21 (August 2010)
 - 14. Pillay, S.R., Solorio, T.: Authorship attribution of web forum posts. In: Proceedings of the eCrime Researchers Summit (eCrime), pp. 1–7. IEEE, Dallas (2010)
 - 15. Plakias, S., Stamatatos, E.: Author identification using a tensor space representation. In: Proceedings of the 18th European Conference on Artificial Intelligence, vol. 178, pp. 833–834. IOS Press, Patras (2008)
 - 16. Potthast, M., Stein, B., Barrón, A., Rosso, P.: An evaluation framework for plagiarism detection. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), pp. 997–1005. Association for Computational Linguistics (August 2010)
 - 17. Raghavan, S., Kovashka, A., Mooney, R.: Authorship attribution using probabilistic context-free grammars. In: Proceedings of the ACL 2010 Conference, Short Papers, pp. 38–42. Association for Computational Linguistics (ACL), Uppsala (2010)

18. Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys* 34(1), 1–47 (2002)
19. Stamatatos, E.: A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology* 60(3), 538–556 (2009)
20. Tearle, M., Taylor, K., Demuth, H.: An algorithm for automated authorship attribution using neural networks. *Literary and Linguist Computing* 23(4), 425–442 (2008)

A New General Grammar Formalism for Parsing

Gabriel Infante-Lopez and Martín Ariel Domínguez

Grupo de Procesamiento de Lenguaje Natural
Universidad Nacional de Córdoba - Argentina
{gabriel,mdoming}@famaf.unc.edu.ar

Abstract. We introduce Probabilistic Constrained W-grammars (PCW-grammars), a two-level formalism capable of capturing grammatical frameworks used in three different state of the art grammar formalism, namely Bilexical Grammars, Markov Rules, and Stochastic Tree Substitution Grammars. For each of them we provide an embedding into PCW-grammars, which allows us to derive properties about their expressive power and consistency, and relations between the formalisms studied.

Keywords: Parsing, PCW-grammars, Bilexical Grammars.

1 Introduction

State of the art statistical parsers, e.g., [5,8,3,15] are procedures for extracting the syntactic structure hidden in natural language sentences. Usually, statistical parsers have two clearly identifiable main components. One has to do the nature of the set of syntactic analysis the parser can provide. It is usually defined using a grammatical framework, such as probabilistic context free grammars (PCFGs), bilexical grammars, etc. The second component concerns the way in which the different parts in the grammatical formalism are learned. For example, for supervised parsers, PCFGs can be read from tree-banks and their probabilities estimated using maximum likelihood [16]. In contrast, for unsupervised dependency parsing [14,1,18], the grammar formalism is inferred from a non-annotated corpus of sentences.

Clearly, the grammatical framework underlying a parser is a key component in the overall definition of the parser. Important characteristics of the parser are determined, either directly or indirectly, through its grammatical formalism; among others, the grammatical framework defines the set of languages the parser can potentially deal with, an upper bound for the parser's complexity, and the type of items that should be learned by the second component. Hence, a thorough understanding of the grammatical framework on which a parser is based, provides a great deal of information about the parser itself. We are particularly interested in the following properties: (1) The expressive power of a grammar formalism. (2) Conditions under which the probability distribution defined over the set of possible syntactic analysis is consistent: if this is the case, the probabilities associated to an analysis can be used as meaningful probabilistic indicators both for further stages of processing [16] and for evaluation [10]. (3) The relation to other grammatical frameworks; this provides insights on the assumptions made by the various frameworks.

Since building a parser is a time consuming process, formal properties of the underlying grammatical framework are not always a priority. Also, comparisons between parser models are usually based on experimental evidence. In order to establish formal properties of parsers and to facilitate the comparison of parsers we believe that a unifying grammatical framework, of which different parsers' grammars can be obtained as instances, is instrumental. Our main contribution is the introduction of a grammatical framework capable of capturing three state of the art grammatical formalisms, namely Bilexical Grammars [9,7,18], Markov Rules [5], and Stochastic Tree Substitution Grammars [3,1]. Our framework is based on so-called W-grammars, due originally to [19]. We constrain W-grammars, to obtain CW-grammars, which are more suitable for statistical natural language parsing than W-grammars. PCW-grammars extend CW-grammars with probabilities. For each the three formalisms, we provide an embedding in PCW-grammars and use this embedding to derive results on expressive power, consistency, and relations among the formalisms.

Section 2 presents our grammatical framework and proves expressive power and conditions to induce consistent distributions. In Section 3 we capture the models mentioned above in our framework, and derive consequences of the embeddings. In Section 4 we discuss the results and conclude.

2 Grammatical Framework

In this section we describe the grammatical framework we will be working with. We introduce constrained W-grammars, then present a probabilistic version, and introduce technical notions needed in later sections.

2.1 Constrained W-Grammars

A *constrained W-grammar (CW-grammar)* is a 6-tuple $(V, NT, T, S, \xrightarrow{m}, \xrightarrow{s})$ such that:

- V is a set of symbols called *variables*. Elements in V are denoted with calligraphic characters, e.g., $\mathcal{A}, \mathcal{B}, \mathcal{C}$.
- NT is a set of symbols called *non-terminals*; elements in NT are noted with upper-case letters, e.g., X, Y, Z .
- T is a set of symbols called *terminals*, denoted with lower-case letters, e.g.: a, b, c , such that V, T and NT are pairwise disjoint.
- S is an element of NT called *starting symbol*.
- \xrightarrow{m} is a finite binary relation defined on $(V \cup NT \cup T)^*$ such that if $x \xrightarrow{m} y$ then $x \in V$. The elements of \xrightarrow{m} are called *meta-rules*.
- \xrightarrow{s} is a finite binary relation on $(V \cup NT \cup T)^*$ such that if $r \xrightarrow{s} s$ then $r \in NT$, $s \neq \epsilon$ and s does not have any variable appearing more than once. The elements of \xrightarrow{s} are called *pseudo-rules*.

PCW-Grammars are rewriting devices, and as such they consist of rewriting rules. They differ from usual rewriting systems in that the rewriting rules do not exist a-priori. Pseudo-rules and meta-rules provide a mechanisms for building ‘real’ rules that can be used in the rewriting process. The rewriting rules are denoted by \xrightarrow{w} . These rules are

built by first selecting a pseudo-rule, and then using meta-rules for instantiating all the variables the pseudo-rule might contains.

For example, let $W = (V, NT, T, S, \xrightarrow{m}, \xrightarrow{s})$ be a CW-grammar where $V = \{ADJ\}$, $NT = \{S, Adj, Noun\}$, $T = \{ball, big, fat, red, green, \dots\}$, while \xrightarrow{m} and \xrightarrow{s} are given by the following table:

meta-rules	pseudo-rules
$ADJ \xrightarrow{m} \mathcal{ADJ} Adj$	$S \xrightarrow{s} \mathcal{ADJ} Noun$
$\mathcal{ADJ} \xrightarrow{m} Adj$	$Adj \xrightarrow{s} big$
	$Noun \xrightarrow{s} ball$
	\vdots

Suppose now that we want to build the rule $S \xrightarrow{w} Adj\ Adj\ Noun$. We take the pseudo-rule $S \xrightarrow{s} \mathcal{ADJ} Noun$ and instantiate the variable \mathcal{ADJ} with $Adj\ Adj$ to get the desired rule. The rules defined by W have the following shape: $S \xrightarrow{w} Adj^* Noun$. Trees for this grammar are flat, with a main node S and all the adjectives in it as daughters; see Figure 1.

The string language $L(W)$ generated by a CW-grammar W is the set $\{\beta \in T^+ : S \xrightarrow{w^*} \beta\}$. In words, a string β belongs to the language $L(W)$ if there is a way to instantiate rules \xrightarrow{w} that derive β from S .

A *w-tree* yielding a string l is defined as the \xrightarrow{w} derivation producing l . A w-tree ‘pictures’ the rules (i.e., pseudo-rules + variable instantiations) that have been used for deriving a string; Figure 1(a) has an example. The way in which the rule has been obtained from pseudo-rules or the way the variables have been instantiated remains hidden. The *tree language* generated by CW-grammar W is the set $T(G)$ defined by all w-trees generated by W yielding a string in $L(G)$.

Theorem 1. CW-Grammars are weakly equivalent to context-free grammars.

Proof. Let $W = (V, NT, T, S, \xrightarrow{m}, \xrightarrow{s})$ be a CW-grammar. Let $G_W = (NT', T', S', R')$ be a context-free grammar defined as follows (to avoid confusion we denote the rules in R by \rightarrow): $NT' = (V \cup NT)$; $T' = T$; S' is the starting symbol in W ; and $X \rightarrow \alpha \in R$ iff $X \xrightarrow{m} \alpha$ or $X \xrightarrow{s} \alpha$. It can be shown that G_W is well-defined and generates the same language as W .

Given a CW-grammar W , the *context-free grammar underlying* W , notation $CFG(W)$, is the grammar G_W defined in the proof Theorem 1. In Figure 1 we show a derivation in W and the corresponding one in $CFG(W)$.

Lemma 1. Let W be a CW-grammar and let $G = CFG(W)$. For every τ in $T(G)$ there is a unique tree $v \in T(W)$.

Proof. We sketch the proof using Figure 1. From the picture it is clear that there is a unique way to hide meta-derivations. The proof follows from applying this simple idea to all trees in $T(G)$.

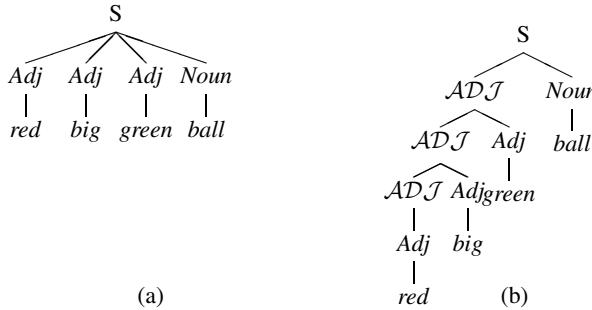


Fig. 1. (a) A tree generated by W . (b) The same tree with meta-rule derivations made visible

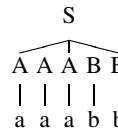


Fig. 2. A derivation tree for the string “aaabb”

A derivation produced by the CFG underlying W has all meta and pseudo derivations visible. Clearly for every derivation in the CFG underlying v , there is a unique $\stackrel{w}{\Rightarrow}$ derivation, and it is obtained by replacing all internal nodes corresponding to head symbols of meta-rules by the yield of the meta-rule production.

Next, we give an example to show that CW-grammars are *not* strongly equivalent to context-free grammars. In other words, trees generated by CW-grammars are different from trees generated by context-free grammars.

Example 1. Let $W = (V, NT, T, S, \xrightarrow{m *}, \xrightarrow{s *})$ a CW-grammar with $V = \{\mathcal{A}, \mathcal{B}, \mathcal{S}\}$, $NT = \{A, B\}$, $T = \{a, b\}$, $\xrightarrow{m} = \{\mathcal{A} \xrightarrow{m} \mathcal{A}A, \mathcal{A} \xrightarrow{m} A, \mathcal{B} \xrightarrow{m} \mathcal{B}B, \mathcal{B} \xrightarrow{m} B\}$, and $\xrightarrow{s} = \{A \xrightarrow{s} a, B \xrightarrow{s} b, \mathcal{S} \xrightarrow{s} \mathcal{A}\mathcal{B}\}$.

The grammar W generates the language $\{a^*b^*\}$ through instantiations of the variables A and B to strings in A^* and B^* respectively. The derivation \xrightarrow{w} for a string $aaabb$ is as follows: $S \xrightarrow{w} AAABB \xrightarrow{w} aAABB \xrightarrow{w} aaABB \xrightarrow{w} aaaBB \xrightarrow{w} aaabb \xrightarrow{w^*} aaabb$. The tree representing this derivation, pictured in Figure 2, only has one internal level (labeled $AAABB$), and its leaves form the accepted string.

No context-free grammar can generate the kind of flat structures displayed in Figure 2 since any context-free grammar producing the same language as W will have more than one intermediate level in its derivation trees.

2.2 Probabilistic CW-Grammars

Probabilistic CW-grammars (PCW-grammars, for short) are CW-grammars where the rules are augmented with a probability value, such that the probabilities belonging to rules sharing the same left-hand side sum up to one. More formally, in a probabilistic CW-grammar $(V, NT, S, \xrightarrow{m}, \xrightarrow{s})$ we have that

- $\sum_{x \xrightarrow{m} p y} p = 1$ for all meta-rules $x \xrightarrow{m} p y$ having x in the left-hand side.
- $\sum_{x \xrightarrow{s} p y} p = 1$ for all pseudo-rules $x \xrightarrow{s} p y$ having x in the left-hand side.

Next, we need to define how we assign probabilities to derivations, rules, and trees. To start with the former, if $\alpha' \xrightarrow{m^*} \alpha$ then there are $\alpha_1, \dots, \alpha_k$ such that $\alpha_i \xrightarrow{m} \alpha_{i+1}$, $\alpha_1 = \alpha'$ and $\alpha_k = \alpha$. We define the probability $P(\alpha' \xrightarrow{m^*} \alpha)$ of a derivation $\alpha' \xrightarrow{m^*} \alpha$ to be $\prod_{i=1}^{k-1} P(\alpha_i \xrightarrow{m} \alpha_{i+1})$

Now, let $X \xrightarrow{w} \alpha$ be a rule. The probability $P(X \xrightarrow{w} \alpha)$ is defined as the product of $P(\alpha' \xrightarrow{m^*} \alpha)$ and $\sum_{\alpha' \in A} P(X \xrightarrow{s} \alpha')$, where $A = \{\alpha' \in (V \cup NT \cup T)^+: X \xrightarrow{s} \alpha', \alpha' \xrightarrow{m^*} \alpha\}$. I.e., the probability of a ‘real’ rule is the sum of the probabilities of all meta derivations producing it.

The *probability of a tree* is defined as the product of the probabilities of the rules making up the tree, while the *probability of a string* $\alpha \in T^+$ is defined as the sum of the probabilities assigned to all trees yielding α .

Theorem 2. *For every PCW-grammar there is a PCF grammar assigning the same probability mass to all strings in the language.*

Proof. Let $G = (NT', T', S', R')$ be a PCFG with NT', T', S' as defined in the proof of Theorem 2 and R' such that $X \rightarrow \alpha \in R$ iff $X \xrightarrow{m} \alpha$ or $X \xrightarrow{s} \alpha$. Note that a \xrightarrow{w} derivation τ might be the product of many different derivations using rules in R' (G -derivations for short); let us call this set $D(\tau)$. From the definitions it is clear that $p(\tau) = \sum_{v \in D(\tau)} p(v)$. To prove the theorem we need to show (1) that for τ and τ' two \xrightarrow{w} derivations of the string α , it holds that $D(\tau) \cap D(\tau') = \emptyset$, and (2) that for every G -derivation v there is a \xrightarrow{w} derivation τ such that $v \in D(\tau)$. Both results follow from Lemma 1.

For a given PCW-grammar W , the PCFG defined in the proof of Theorem 2 is called *the PCFG underlying W* . As an immediate consequence of the construction of the PCFG given in Theorem 2 we get that a PCW-grammar is *consistent* iff its underlying PCFG is consistent.

2.3 Learning CW-Grammars from Tree-Banks

PCW-grammars are induced from tree-banks in almost the same way as PCFG are. The main difference is that the former requires an explicit decision on the nature of the hidden derivations. As we will see, the three different approaches we present, differ substantially in the assumptions they made in this respect.

2.4 Some Further Technical Notions

In later sections we will be using PCW-grammars to “capture” models underlying a number of state of the art parsers. The following will prove useful. Let F and G be two grammars with tree languages $T(G)$ and $T(F)$ and languages $L(F)$ and $L(G)$, respectively. We say that F is *f-equivalent* to G if $L(G) = L(T)$ and there is a bijective function $f : T(F) \rightarrow T(G)$. Given two grammatical formalisms A and B , we say that A is *f-transformable* to B , if for every grammar F in A there is a grammar G in B such that F is f -equivalent to G .

3 Capturing State of the Art Parsers

In this section we use PCW-grammars to capture the models underlying a number of state of the art parsers.

3.1 Bilexical Grammars

Bilexical grammars [8,9,7,18] is a formalism in which lexical items, such as verbs and their arguments, can have idiosyncratic selectional influences on each other. They can be used for describing bilexical approaches to dependency and phrase-structure grammars, and a slight modification yields link grammars.

Background. A *split unweighted bilexical grammar* B is a 3-tuple $(W, \{r_w\}_{w \in W}, \{l_w\}_{w \in W})$ where:

- W is a set, called the (terminal) *vocabulary*, which contains a distinguished symbol ROOT
- For each word $w \in W$, a pair of regular grammars l_w and r_w , having starting symbols S_{r_w} and S_{l_w} . Each grammar accepts some regular subset of W^* .

A *dependency tree* is a tree whose nodes (internal and external) are labeled with words from W ; the root is labeled with the symbol ROOT. The children ('dependents') of a node are ordered with respect to each other and the node itself, so that the node has both *left children* that precede it and *right children* that follow it. A dependency tree T is *grammatical* if for every word token w that appears in the tree, l_w accepts the (possibly empty) sequence of w 's left children (from right to left), and r_w accepts the sequence of w 's right children (from left to right).

Example 2. Let $B = (W, \{l_w\}_{w \in L}, \{r_w\}_{w \in L})$ be a split regular grammar defined as follows: $W = \{a, b\}$, $l_a = b^*$, $r_a = \epsilon$, $l_b = \epsilon$, $l_{ROOT} = a$, $r_{ROOT} = \epsilon$ and $r_b = (a|b)^*$. This grammar accepts the string "bbabaa" with the analysis tree pictured in Figure 3. Because l_{ROOT} accepts a , l_a accepts "bbb", l_a accepts ϵ , l_b accepts ϵ , and r_b accepts "a".

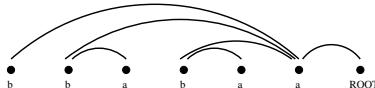


Fig. 3. An example of a dependency tree

Bilexical Grammars as CW-Grammars

Definition 1. Let $B = (W, \{l_w\}_{w \in L}, \{r_w\}_{w \in L})$ be a split bilexical grammar. Let $W_B = (V, NT, T, S, \xrightarrow{m}, \xrightarrow{s})$ be the CW-grammar defined as follows:

- The set of variables V is given by the set of starting symbols S_{l_w} and S_{r_w} from regulars grammars l_w and r_w respectively, and w in W .
- The set of non-terminals NT is some set in 1-1-correspondence with W , e.g., it can be defined as $NT = \{w' : w \in W\}$.
- The set of terminals T is the set of words W .

- The set of meta-rules is given by the union of $\{w' \xrightarrow{m} w : w \in W\}$ and the rules in all of the right and left regular grammars.
- The set of pseudo-rules is given by $X' \xrightarrow{s} S_{l_w^-} x S_{r_w}$ where l_w^- denotes the regular expression inverting (reading backwards) all strings in $L(l_w)$.

Below, we establish the (weak) equivalence between a bilexical grammar B and its CW-grammar counterpart W_B . The idea is that the set of meta-rules, producing derivations that would remain hidden in the tree, are used for simulating the regular automata. Pseudo-rules are used as a nexus between a hidden derivation and a visible one: For each word w in the alphabet, we define a pseudo-rule having w as a terminal, and two variables S_{l_w} and S_{r_w} marking the left and right dependents, respectively. These variables correspond to the starting symbols for the left and right automata l_w and r_w , respectively. Instantiating the pseudo-rule associated to w would use a left and a right derivation using the left and the right automata, respectively, via meta-rules. The whole derivation remains hidden in the \xrightarrow{w} derivation, as in bilexical grammars.

Lemma 2. *Bilexical grammars are f -transformable to CW-grammars.*

Proof. We have to give a function $f : T(B) \rightarrow T(W_B)$, where B is a bilexical grammar and W_B the grammar defined in Definition 1, such that f is invertible. A bilexical tree yielding the string $s = w_1, \dots, w_n$ can be described as a sequence u_1, \dots, u_n of 3-tuples $\langle \alpha_i, w_i, \beta_i \rangle$ such that l_{w_i} accepts α_i and r_{w_i} accepts β_i . The desired function f transforms a dependency tree in a w-tree by transforming the sequence of tuples into a \xrightarrow{w} derivation. We define f as $f(\langle \alpha, w_i, \beta \rangle) = W_i \xrightarrow{w} \alpha w_i \beta$. The rule corresponding to $\langle \alpha, w_i, \beta \rangle$ is the one produced by using the pseudo rule $W'_i \xrightarrow{s} S_{l_w^-} x S_{r_w}$ and instantiating $S_{l_w^-}$ and S_{r_w} with α and β respectively. Since the sequence of tuples forms a dependency tree, the sequence of w-rules builds up a correct w-tree.

Weighted bilexical grammars are like unweighted bilexical grammars but all of their automata assign weights to the strings they generate. Lemma 2 implies that weighted bilexical grammars are a subclass of PCW-grammars.

Properties. By Lemma 2 bilexical grammars are weakly equivalent to context free grammars. Moreover, Example 1 can be used to show that bilexical grammars are not strongly equivalent to CFGs; hence, the probabilistic version of bilexical grammars cannot be captured by PCFGs.

As a consequence of Lemma 2, learning bilexical grammars is equivalent to learning PCW-grammars, and since every PCW-grammar has an underlying PCFG, learning a PCW-grammar is equivalent to learning its underlying PCFG. [8] assumed that all hidden derivations were produced by Markov chains. Under the PCW-paradigm, his methodology is equivalent to transforming all trees in the training material by making all their hidden derivations visible, and inducing the underlying PCFG from the transformed trees. Variables in the equivalent PCW-grammar are defined according to the level degree of the Markov chain (we assume that the reader is familiar with Markov models and Markov chains [13]). In particular, if the Markov chain used is of degree one, variables are in one-to-one correspondence with the set of words, and it can be shown that the resulting bilexical grammar is consistent. The result follows from the

fact that inducing a degree one Markov chain in a bilexical grammar is the same as inducing the underlying PCFG in the equivalent PCWG using maximum likelihood, plus the fact that using maximum likelihood for inducing PCFG produces consistent grammars [11,4].

3.2 Markov Rules

In this subsection we capture one of the models presented by Collins, his so-called first model. The main idea behind [5,6] is to extend what he calls a “simple” CFG to a lexicalized backed-off grammar.

Background. Collins’ first model may be viewed as a way to describe the probabilities assigned to CF-like rules. A rule has the following shape:

$$(1) \quad P(h) \rightarrow L_n(l_n) \dots L_1(l_1) H(h) R_1(r_1) \dots R_m(r_m),$$

where H is the head-child of the phrase, which inherits the head word h from its parent P , and where $L_n(l_n), \dots, L_1(l_1)$ and $R_1(r_1), \dots, R_m(r_m)$ are left and right modifiers of H , respectively. Either or both of n and m may be zero, and $n = m = 0$ for unary rules. Figure 4 shows a tree with its respective rules.

Collins defines the probability of a rule (such as (1)) as the probability of its right-hand side, conditioned on the probability of its left-hand side, which is then decomposed as follows:

$$\begin{aligned} & \mathcal{P}(L_n(l_n) \dots L_1(l_1) H(h) R_1(r_1) \dots R_m(r_m) | P(h)) = \\ & \mathcal{P}_h(H | P(h)) \times \prod_{i=1, \dots, n+1} \mathcal{P}_l(L_i(l_i) | L_1(l_1), \dots, L_{i-1}(l_{i-1}), P(h), H) \times \\ & \prod_{j=1, \dots, m+1} \mathcal{P}_r(R_j(r_j) | L_1(l_1), \dots, L_{n+1}, R_1(r_1), \dots, R_{j-1}(r_{j-1}), P(h), H). \end{aligned}$$

Collins approximates the probabilities using Markov independence assumptions for each order. In particular, the generation of the right-hand side of a rule as in (1), given the left-hand side, is decomposed into three steps:

1. Generate the head constituent label of the phrase, with probability $\mathcal{P}_H(H | P, h)$.
2. Generate modifiers to the left of the head with probability

$\prod_{i=1, \dots, n+1} \mathcal{P}_L(L_i(l_i) | P, h, H)$, where $L_{n+1}(l_{n+1}) = \text{STOP}$. The STOP symbol is a non-terminal, and the model stops generating left modifiers as soon as it has been generated.



Fig. 4. A lexicalized parse tree and the rules it contains; POS tags omitted for brevity

3. Generate modifiers to the right of the head with probability

$$\prod_{i=i \dots m+1} \mathcal{P}_R(R_i(r_i) | P, h, H). R_{m+1}(r_{m+1}) \text{ is defined as STOP.}$$

We can think of the probabilities $\mathcal{P}_R(R_i(r_i) | P, h, H)$ and $\mathcal{P}_L(L_i(r_i) | P, h, H)$ as being produced by a zero order Markov chain.

Markov Rules as CW-grammars. For capturing blexical grammars, we first described the formalism using regular language and later added probabilities. To capture Collins' first model we proceed in the opposite direction. We use the zero-order Markov model he builds to define regular languages and use these to build a CW grammar corresponding to Collins' model.

Independent of their order, *Markov models* or *Markov chains* describe a regular language. Let $M = (S, P, F, I)$ be a Markov chain, where S is a sequence of states, P the transition matrix, $F \subseteq S$ the set of absorbing states, and I the initial distribution of probabilities. We can transform the Markov model into an automaton A_M directly by taking S as the states of the automaton, F as the set of final states, and the initial states as the state that receives an initial probability mass.

Let NT be the set of possible phrase names, e.g., NP , PP , etc.; let W be the set of words in the lexicon; we assume that both sets are finite. It can be seen from the definition of rules, that for each pair $(H, w) \in NT \times W$ there are two Markov chains $r_{(H,w)}, l_{(H,w)}$.

A bit of notation: following Collins, we rewrite rules such as (1) as follows:

$$(P, h) \rightarrow (L_n, l_n) \dots (L_1, l_1)(H, h)(R_1, r_1) \dots (R_m, r_m)$$

The probability of strings $(H, h)(L_1, l_1) \dots (L_n, l_n)$ and $(H, h)(R_1, r_1) \dots (R_m, r_m)$ is given by the probabilities assigned to paths $(L_1, l_1) \dots (L_n, l_n)$ STOP and $(R_1, r_1) \dots (R_m, r_m)$ STOP in the Markov chains $l_{(H,h)}$ and $r_{(H,h)}$ respectively.

Definition 2. Let $B = (NT, W, \{l_{(H,w)}\}_{w \in W, h \in NT}, \{r_{(H,w)}\}_{w \in W, h \in NT})$ be a grammar based Markov rules. Let $W_B = (V, NT, T, S, \xrightarrow{m}, \xrightarrow{s})$ be the CW-grammar defined as follows:

- The set of variables V is given by the set of starting symbols $S_{l_{(H,w)}}$ and $S_{r_{(H,w)}}$ from the regular grammars $l_{(H,w)}$ and $r_{(H,w)}$ respectively, and w in W .
- The set of non-terminals NT is some set in 1-1-correspondence with W , e.g., it can be defined as $NT = \{w' : w \in W\}$.
- The set of terminals T is the set of words W .
- The set of probabilistic meta-rules is given by the union of the rules in each of the right and left regular grammars (i.e., the set given by $A \xrightarrow{m} \alpha$ iff $A \rightarrow \alpha \in l_{(H,w)}$ or $A \rightarrow \alpha \in r_{(H,w)}$ for some H and some w) plus the set $\{w' \xrightarrow{m} w : w \in W\}$.
- The set of pseudo-rules is given by $(P, h) \xrightarrow{s} \mathcal{P}_H(H|(P, h)) S_{l_{(H,h)}}(H, h) S_{r_{(H,h)}}$

An important technical detail: a pair (H, h) should not be viewed as separate variables, but as a single entity; consequently, they are treated as one for replacing.

Lemma 3. *Markov rules as used in Collins' first model are f -transformable to CW-grammars.*

Properties. By Lemma 3 Collins' first model is weakly equivalent to context free grammars. Moreover, Example 1 can be used to show that the model is not strongly equivalent to CFGs; consequently, its probabilistic version can not be captured using PCFGs. As a consequence of Lemma 3, learning Markov rules-based grammars is equivalent to learning PCW-grammars and since every PCW grammar has an underlying PCFG, learning a PCW-grammar is equivalent to learning its underlying PCFG.

[5] assumed that all hidden derivations were produced by Markov chains. Under the PCW-paradigm, his methodology is equivalent to transforming all trees in the training material by making all their hidden derivations visible, and inducing the underlying PCFG from the transformed trees. Variables in the equivalent PCW-grammar are defined according to the level degree of the Markov chain. If the Markov chain used is of degree zero, there is only one variable (the Markov chain contains a unique state), and the induced Markov rules-based grammar is consistent. The result follows from the fact that inducing a degree zero Markov chain is the same as inducing the underlying PCFG in the equivalent PCW-grammar using maximum likelihood, plus the fact that using maximum likelihood for inducing PCFGs produces consistent grammars [11,4].

The embedding of Collins' and Eisner's model into CW-grammars shows that even though the first deals with phrase structures and the second with dependency structures, they only differ in the number of variables they assume to exist in the underlying PCW-grammars.

3.3 Stochastic Tree Substitution Grammars

Data-oriented parsing (DOP) is a memory-based approach to syntactic parsing. The basic idea is to use the subtrees from a syntactically annotated corpus directly as a stochastic grammar. The DOP-1 model [2] was the first version of DOP, and most later versions of DOP are variations on it. Recently, in [1] the authors used Tree Substitution Grammars in Unsupervised dependency parsing and they obtained the state of the art results. The underlying grammatical formalism in all previous models is Stochastic Tree Substitution Grammars (STSG), the grammatical formalism we capture here.

Background. The model itself is extremely simple and can be described as follows: for every sentence in a parsed training corpus, extract every subtree. Now we use these trees to form an Stochastic Tree Substitution Grammar.

A *Stochastic Tree-Substitution Grammar* (STSG) G is a 5-tuple $\langle V_N, V_T, S, R, P \rangle$ where

- V_N is a finite set of nonterminal symbols.
- V_T is a finite set of terminal symbols.
- $S \in V_N$ is the distinguished symbol.
- R is a finite set of elementary trees whose top nodes and interior nodes are labeled by nonterminal symbols and whose yield nodes are labeled by terminal or nonterminal symbols.
- P is a function which assigns to every elementary tree $t \in R$ a probability $P(t)$. For a tree t with a root node symbol $root(t) = \alpha$, $P(t)$ is interpreted as the probability of substituting t on a node α . We require, therefore, for a given α that $\sum_{\{t:root(t)=\alpha\}} P(t) = 1$ and that $0 < P(t) \leq 1$ (where t 's root node symbol is α).

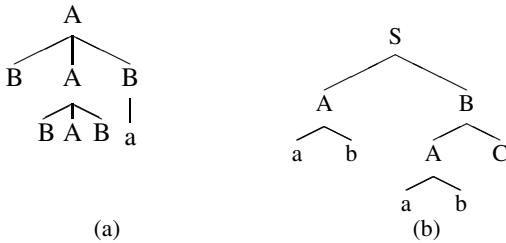


Fig. 5. (a) A derivation tree. (b) An elementary tree.

If t_1 and t_2 are elementary trees such that the leftmost non-terminal frontier node symbol of t_1 is equal to the root node symbol of t_2 , then $t_1 \circ t_2$ is the tree that results from substituting t_2 in this leftmost non-terminal frontier node symbol in t_1 . The partial function \circ is called *leftmost substitution* or simply *substitution*. Trees are derived using left most substitution.

STSGs as CW-grammars. DOP-1 is not quite a context-free grammar. The main difference, and the hardest to capture in a CFG-like setting, is the way in which probabilities are computed for a given tree. The probability of a tree is given by the sum of the probabilities of all derivations producing it. CW-grammars offer a similar mechanism: the probability of the body of a rule is the sum of the probabilities of all meta-derivations producing the rule body. The idea of the equivalence is to associate to every tree produced by DOP a ‘real’ rule of the PCW-grammar in such a way that the body of the rule codifies the whole tree.

To implement this idea, we need to code up trees as strings. The simplest way to achieve this is to visit the nodes in a depth first left to right order and for each inner node use the applied production, while for the leaves we type the symbol itself if the symbol is a terminal and a primed version of it if the symbol is a non-terminal. For example the derivation describing tree the in Figure 5.(a) is $(A, BAB)B'a(A, BAB)(B, a)a$.

We start capturing STSG by building rules capturing elementary trees using the notation just introduced. Specifically, let t be an elementary tree belonging to a STSG. Let S be its root and α its string representation. The CF-like rule $S' \rightarrow \alpha$ is called the *elementary rule* of t . Elementary rules store all information about the elementary tree. They have primed non-terminals where a substitution can be carried out. For example, let t be the elementary tree pictured in Figure 5.(b), its corresponding elementary rule is $S' \rightarrow (A, B)(A, ab)ab(B, AC)(A, ab)abC'$. Note the primed version of C in the frontier of the derivation.

Definition 3. Let $H = (V_N, V_T, S, R, P)$ be a STSG. Let $W_H = (V, NT, T, S', \xrightarrow{s}, \xrightarrow{m})$ be the following CW-grammar:

- V is the primed version of V_T .
- (A, α) is in NT iff $(A, \alpha) \rightarrow \epsilon$ appears in some elementary tree.
- T is exactly as V_T .
- S' is a new symbol.
- The set of meta-rules is built by transforming each elementary tree to its corresponding elementary rule.

- The set of pseudo-rules is given by $(A, \alpha) \xrightarrow{s} \epsilon$ if $A \rightarrow \alpha$ appears in a elementary tree, plus rules $S' \xrightarrow{s} S$.

Two remarks. First, all generative capacity is encoded in the set of meta-rules. In the CW-world, the body of a rule (i.e., an instantiated pseudo-rule) encodes a derivation of the DOP-1 model. Second, the probability of a ‘real’ rule is the sum of the probabilities of meta-derivations yielding the rule’s body.

Lemma 4. Let $H = (V_N, V_T, S, R, P)$ be a STSG and let W_H be the CW-grammar given in Definition 3. There is a one-to-one correspondence between derivations in H and meta-rule derivations in W_H .

Proof. Let t be a tree produced by H . We prove the lemma using induction on the length of the derivation producing t . If t has length 1, there is an elementary tree t_1 such that S is the root node and yields α , which implies that there is a meta-rule obtained from the elementary rule corresponding to the elementary tree t_1 . The relation is one-to-one as, by definition, meta-rules are in one-to-one correspondence with elementary trees.

Suppose the lemma is true for derivation lengths less than or equal to n . Suppose t is generated by a derivation of length $n + 1$. We can assume that there are trees t_1, t_2 such that $t_1 \circ t_2 = t$. By definition there is a unique meta-rule r_1 corresponding with t_1 and by inductive hypothesis there is a unique derivation for t_2 .

Lemma 5. Let $H = (V_N, V_T, S, R, P)$ be an STSG, and let W_H be the CW-grammar given in Definition 3. Then W_H accepts the same set of strings as H , i.e., STSGs and PCW-grammars are weakly equivalent.

Proof. Let α be a string in $L(H)$. There is at least one tree derivation $t_1 \circ \dots \circ t_k$ yielding α . From Lemma 4 we know that there is a CW-Rule $S' \xrightarrow{w} \alpha$ such that after applying rules $(A, \beta) \xrightarrow{s} \epsilon$, α is obtained.

Corollary 1. Let $H = (V_N, V_T, S, R, P)$ be an STSG, and let W_H be the CW-grammar given in Definition 3. There is a one-to-one correspondence between derivations in H and W_H .

Corollary 2. STSGs are f -transformable to CW-grammars.

Lemma 6. Let $H = (V_N, V_T, S, R, P)$ be an STSG, and let W_H be the CW-grammar given in Definition 3. Both grammars assign the same probability mass to trees related through the one-to-one mapping described in Corollary 1.

Proof. A tree has a characteristic W-rule, defined by its shape. I.e., the probability of a W-Rule according to the definition of PCW-grammars is given by the sum of the probabilities of all derivations producing the rule’s body, i.e., all STSG derivations producing the same tree. As a consequence, a particular STSG tree, identified from the body of the corresponding w-rule, has the same probability assigned by the equivalent CW-Grammar.

Properties. By Corollary 2, STSGs are weakly equivalent to context free grammars. The consistency of an STSG depends on the methodology used for computing the probabilities assigned to its elementary trees. DOP-1 is one particular approach to computing these probabilities. Under it, a tree t contributes all its possible subtrees to a new tree-bank from which the probabilities of elementary trees are computed. Probabilities of an elementary tree are computed using maximum likelihood. Since the events in the new tree-bank are not independently distributed the resulting probabilities are inconsistent and biased [12]. Solutions that take into account the dependence between trees in the resulting tree-banks have been suggested [17].

Even though consistency conditions cannot be derived for the DOP-1 estimation procedure given that it does not attempt to learn the underlying PCFG, our formalism suggests that probabilities should be computed differently. By our embedding, a tree t in the tree-bank corresponds to the body of a pseudo-rule instantiated through meta-derivations; t is the final “string” and does not have any information on the derivation that took place. But viewing t as a final string changes the problem definition completely. Now, we have as input a set of elementary rules and a set of accepted trees. The problem is to compute probabilities for these rules: an unsupervised problem that can be solved using any unsupervised technique. The resulting STSG’s consistency depends on the consistency properties the unsupervised method.

4 Discussion and Conclusion

We introduced probabilistic constrained W-grammars, a grammatical framework capable of capturing a number of models underlying state of art parsers. We provided expressive power properties for three formalisms (Bilexical Grammars, Markov Rules, and Stochastic Tree Substitution Grammars) together with some conditions under which the inferred grammars are consistent.

We showed that, from a formal perspective, Bilexical Grammars and Markov rules do not differ in a principled way: both are based on approximating bodies of rules using Markov models. We also found that STSGs and Markov rules also have certain similarities. Markov rules suppose that rule bodies are obtained by collapsing hidden derivations. That is, for Collins a rule body is a regular expression (equivalent to Markov models). Similarly, Bod’s DOP model takes this idea to the extreme by taking the whole sentence to be the yield of a hidden derivation. PCW-grammars suggest that we explore intermediate levels of abstraction, levels in between having everything hidden and only having productions hidden.

References

1. Blunsom, P., Cohn, T.: Unsupervised induction of tree substitution grammars for dependency parsing. In: Proceedings of EMNLP 2010 (2010)
2. Bod, R.: Enriching linguistics with statistics. Ph.D. thesis, Univ. Amsterdam (1995)
3. Bod, R.: Beyond Grammar—An Experience-Based Theory of Language. Cambridge University Press (1999)
4. Chi, Z., Geman, S.: Estimation of probabilistic context-free grammars. Compu. Ling. 305, 24:299–24:305 (1998)

5. Collins, M.: Three generative, lexicalized models for statistical parsing. In: ACL 1997 (1997)
6. Collins, M.: Head-driven statistical models for natural language parsing. Ph.D. thesis, Univ. Pennsylvania (1999)
7. Domníquez, M.A., Infante-Lopez, G.: Searching for Part of Speech Tags That Improve Parsing Models. In: Nordström, B., Ranta, A. (eds.) GoTAL 2008. LNCS (LNAI), vol. 5221, pp. 126–137. Springer, Heidelberg (2008)
8. Eisner, J.: Three new probabilistic models for dependency parsing: An exploration. In: Proceedings of International Conference on Computational Linguistics (COLING-1996), Copenhagen (1996)
9. Eisner, J.: Bilexical grammars and their cubic-time parsing algorithms. In: Advances in Probabilistic and Other Parsing Technologies, pp. 29–62 (2000)
10. Infante-Lopez, G., de Rijke, M.: Comparing the ambiguity reduction abilities of probabilistic context-free grammars. In: Proc. LREC 2004 (2004)
11. Joan-Andreu, S., Bened, J.M.: Consistency of stochastic context-free grammars from probabilistic estimation based on growth transformations. IEEE Trans. on Pattern Analysis and Machine Intelligence (1997)
12. Johnson, M.: The dop estimation method is biased and inconsistent. Comp. Ling 76, 28:71–28:76 (2002)
13. Jurafsky, D., Martin, J.: Speech and Language Processing. Prentice Hall (2000)
14. Klein, D., Manning, C.: Corpus-based induction of syntactic structure: Models of dependency and constituency. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (2004)
15. Koo, T., Carreras, X., Collins, M.: Simple semi-supervised dependency parsing. In: Proc. ACL/HLT (2008)
16. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press (1999)
17. Sima'an, K., Buratto, L.: Backoff Parameter Estimation for the DOP Model. In: Lavrač, N., Gamberger, D., Todorovski, L., Blokceel, H. (eds.) ECML 2003. LNCS (LNAI), vol. 2837, pp. 373–384. Springer, Heidelberg (2003)
18. Spitkovsky, V.I., Alshawi, H., Jurafsky, D., Manning, C.D.: Viterbi training improves unsupervised dependency parsing. In: Proc. CoNLL 2010 (2010)
19. van Wijngaarden, A.: Orthogonal design and description of a formal language. In: Technical Report MR76, Mathematisch Centrum, Amsterdam (1965)

Contextual Semantic Processing for a Spanish Dialogue System Using Markov Logic

Aldo Fabian¹, Manuel Hernandez¹, Luis Pineda², and Ivan Meza^{2,*}

¹ Instituto de Computación

Universidad Tecnológica de la Mixteca (UTM)

<http://www.utm.mx/>

² Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS)

Universidad Nacional Autónoma de México (UNAM)

eMoya801@gmail.com, manuelhg@mixteco.utm.mx, luis@leibniz.iimas.unam.mx,

ivanvladmir@turing.iimas.unam.mx

<http://www.iimas.unam.mx/>

Abstract. Semantic processing is vital in a dialogue system for the language understanding stage. Recent approaches of semantic processing rely on machine learning methods to perform the task. These are more robust to errors from the speech recogniser. Although these approaches are built on the domain of the dialogue system they do not incorporate contextual information available in the dialogue system. In this paper, we explore the use of contextual information in the form of expectations of a dialogue system to perform semantic processing in a *Spoken Dialogue System*. We show the benefits on doing so, and propose a Markov Logic model which incorporates such information.

Keywords: Automatic Speech Recognition, Children Speech.

1 Introduction

Semantic processing is the stage of a dialogue system at the understanding level which is in charge of assigning a semantic representation to a spoken interaction of a user. For instance, if the user tell the system *my name is John* the semantic processing module could produce the following logical form `name(john)` which represents the meaning of the utterance. This semantic representation is used by the dialogue system to identify its next action; for instance, in this case the system could reply *hello John*.

The developers of dialogue systems have the options of using rule-based or data-oriented approaches. For the rule-based approaches the developer can collect a set of keywords or propose a grammar for the syntactic analysis. For the data-oriented approach, a machine learning technique can be used to learn the mapping from utterance to semantic representation based on examples. In this

* We thank the enthusiastic support of the members of the Golem Group at IIMAS. We also gratefully thank the support of grants CONACYT 81965 and PAPPIT-UNAM IN-115710.

paper, we explore both options in the light of contextual information. For the case of rule-based, we use the word spotting approach enhanced with regular expressions. For the case of data-oriented, we use the Markov Logic.

Context is a complex concept in dialogue systems. It can mean from a simple state of the conversation to the history of the conversation. Previous work has been successful in exploiting the user generated context [9]. Other has put attention to the state information of system [5]. In this work we focus on what the system expects from the user. This information encodes the current system situation and synthesizes much of the previous history. For instance if the system just said *What is your name?* then the system will expect a name instance as an answer. We use this expectation information to guide the semantic processing.

The structure of the paper is as follows. Section 2 reviews the dialogue modeling we use to extract the expectations of the system. Section 3 presents a summary of both of the techniques used in this paper: word spotting and Markov Logic. Section 4 presents the main characteristics of the corpus. Section 5 presents the experiments and their results. Finally, section 7 presents the main conclusions and future work.

2 Dialogue Models

We base our contextual information on dialogues models as defined by the Interaction-Oriented Cognitive Architecture (IOCA) [7]. In this approach, dialogue models are representations of conversational protocols which are defined in advance through analysis. A particular application is defined as a set of *dialogue model* units. These units are then assembled dynamically during the interaction producing rich and natural conversations. The main tenet of this approach is that dialogue acts are expressed and interpreted in relation to a conversational context that is shared between the speaker and hearer.

IOCA models practical dialogues in which the conversational partners “visit” conversational *situations* with highly structured expectations about what can be expressed by the interlocutor, not only about the spoken language but any modality. These expectations are called expected intentions or simply *expectations*. Once an expectation is satisfied then a particular *action* will be performed. Situations, expectations and actions of an application domain are encoded through dialogue models. A dialogue model can be represented as a directed graph (cycles are permitted). Situations are represented as nodes and edges are labeled with expectation and action pairs. Figure 2 presents a single transition between two situations. If the expectation of an edge is satisfied by the current interpretation, then the corresponding action is performed. Situations can have one or more input and output expectation-action pairs.

Expected intentions and actions are expressed through abstractions that are independent of the expression used by the interlocutor and of the actual patterns that appear on the visual field of the system. These abstractions allow to capture a wide range of possible concrete communication behaviors. Expectations are expressed through a declarative notation representing speech acts and actions.



Fig. 1. Specification of a situation’s transition

Actions are also specified declaratively through *Multimodal Rhetorical Structures* (MRS); these are lists of basic rhetorical acts, defined along the lines of the *Rhetorical Structure Theory* (RST) [3]. Although the specification of MRS is also modality independent, the basic rhetorical acts are associated to an output modality. Accordingly, a MRS is thought of as “paragraph” in which some of its sentences are rendered through speech, but others may be rendered visually, as texts, pictures, animations and video. The specification of speech acts and rhetorical acts can be expressed through concrete expressions (e.g., constants and grounded predicates), but also these can be expressed through propositional functions.

The notation for transitions is illustrated in Figure 2, where the situation s_f is reached from s_i if the corresponding expectation is satisfied; during this transition the MRS is performed by the system. For instance, Figure 2 represents the situation when the system expects a name and once this is satisfied, it executes the `say_name` MSR which repeats such name and arrives to a new situation, which for instance could ask for the age.

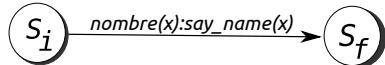


Fig. 2. Example of an dialogue model transition with an expectation

In this work, we take advantage of the expectations of the system to guide the semantic processing that will produce the interpretation that will satisfy at least one of such expectations.

3 Semantic Processing

In this work we explore two different approaches for semantic processing. In the case of rule-based approaches, we use an enhance version of the *word spotting* approach where instead of keywords it uses regular expressions. In the case of data-oriented we use the *Markov Logic Network* (MLN) framework [2].

3.1 Word Enhanced Spotting

The word spotting we use in this research is pretty straightforward. There is a list of expectations with their corresponding regular expressions. Therefore, if the

Algorithm 1. Word enhanced spotting (WES)

```

Require: exp2res;
Require: expectations;
Require: test;
1: for exp  $\in$  expectations do
2:   re  $\leftarrow$  exp2res[exp]
3:   if match(re,text) then
4:     res  $\leftarrow$  resolve(re, text)
5:   end if
6: end for
7: if not res then
8:   res=error
9: end if
10: return res

```

semantic processing receives an utterance transcription with a set of expectation it will check if any of the regular expressions associated to any of the expectation matches. If that the case, it returns the such expectation otherwise it will return an error code. The basic algorithm is presented in Algorithm 1.

3.2 Markov Logic

Markov Logic is a Statistical Relational Learning language based on First Order Logic and Markov Networks. It can be seen as a formalism that extends First Order Logic to allow formulae that can be violated with some penalty. From an alternative point of view, it is an expressive template language that uses First Order Logic formulae to instantiate Markov Networks of repetitive structure.

Let us describe ML by considering the semantic processing task. In ML we can model this task by first introducing a set of logical predicates such as *repr(Feature, Value)* or *word(Position, Orthography)*. The first predicate defines the semantic representation, in this case it consists of a *feature* and a *value*. On the other hand, the second predicate defines the tokens of a utterance, based on its position and the orthography of the word. With these two predicates, we can specify a set of weighted first order formulae that define a distribution over sets of ground atoms of these predicates (or so-called possible worlds). Ideally, the distribution we define with these weighted formulae assigns high probability to possible worlds where the semantic representation predicates are correctly identified and a low probability to worlds where this is not the case. For example, a suitable set of weighted formulae would assign a high probability to the world:

{*word(1, my), word(2, name), word(3, is), word(4, john), repr(name, john)*}
and a low one to:

{*word(1, my), word(2, name), word(3, is), word(4, john), repr(name, is)*}

In the first case, the semantic representation for the utterance *my name is john* is *name(john)* which is right. In the second case is *name(is)* which is incorrect.

As an example we present the formulae which relates each token of the utterance with its semantic representation:

$$\text{word}(Pos, Ortho) \wedge \text{repr}(Feat, Val)$$

This formula establishes a relation between any word (with position Pos and orthography $Ortho$) and the semantic representation (with feature $Feat$ and value Val). This FOL becomes a Markov Logic Network when it is associated to a weight. Figure 3 shows the factor graph representation resulting of relating the formula to weights. In the Figure, we observe the abstract representation and one possible grounding for the utterance *my name is John*.

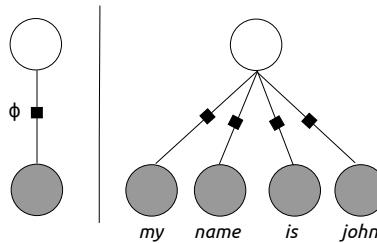


Fig. 3. Graphical representation of a MLN. Left side is the abstract graph of a formula. The right side it the grounded formulae for such formula and the utterance *my name is jonh*

In summary, to model a problem in Markov Logic Network (MLN) it is necessary to identify the formulae which defines the model. Following, we present the formulae used in this research:

- Relation between any word and the semantic representation (base)
 $\text{word}(Pos, Ortho) \wedge \text{repr}(Feat, Val)$
- Relation between any POS tag and the semantic representation (POS)
 $\text{pos}(Pos, Tag) \wedge \text{repr}(Feat, Val)$
- Relation between any word and its previous word and the semantic representation (*bigram*)
 $\text{word}(Pos, Ortho_1) \wedge \text{word}(Pos - 1, Ortho_2) \wedge \text{repr}(Feat, Val)$
- Relation between any POS tag and a POS tag of a window (*window_n*)
 $\text{pos}(Pos, Tag) \wedge \text{word}(Pos_n, Ortho) \wedge \text{repr}(Feat, Val)$
- Relation expectation (context) and the semantic representation (*contextual*)
 $\text{expectation}(Exp) \wedge \text{repr}(Feat, Val)$

Different models can be build using different combinations of such formulae.

Assuming that we have an MLN, a set of weights and a given sentence then we need to predict the semantic representation with *maximal a posteriori probability* (MAP). To this end we apply a method that is both exact and efficient: Cutting Plane Inference (CPI) [8] with Integer Linear Programming (ILP) as base solver. We learn the weights associated with each MLN using 1-best *MIRA Online solver*.

4 Corpus

We based this research on the “Guess the card” real users corpus [6]. The “Guess the card” is a working system which is constantly collecting the interactions with its users. A subset of these interactions conforms this corpus. The corpus is composed by 132 interactions for development and 40 interactions for testing. All of these interactions are real interactions between different users and the system which were manually augmented with orthographic and semantic transcriptions. At the end of the manual labelling, we are left with both versions of the system: the one using manual transcriptions and the one using automatic transcriptions from the ASR module from the system. Table 1 summarises the main characteristics of this corpus.

Table 1. “Guess the card” real users corpora main properties

	Development	Testing
Interactions	132	40
Utterances	1,806	516
Utts. per interaction	13.7	12.9
Avg. Length	1.73	1.97
Token vocabulary	3,140	1,021

5 Experiments

We have defined two type of experiments to measure the effect of the contextual information in the semantic processing. The *baseline* experiments establish the baseline without using the contextual information. The *context* experiments show the performance with the contextual information. In both types of experiments we try the word spotting and the Markov Logic approaches.

Another point to explore is the relation of the semantic processing with the ASR system present in a dialogue system. The ASR module usually makes errors during the transcription of the spoken interaction of the user. A good semantic process should be robust enough to resist those errors. During the experiment, we test both the manual and automatic transcriptions.

For all experiments, we measure the *accuracy* of recovering a complete semantic representation. This is a strict metric because if a part of the semantic representation is missing there are not points in favor of the performance. In order to have a more relaxed measure which takes into account parts of the semantic representation we also measure the *f*-score of retrieving the right parts of the semantic representation. The *f*-score is a weighted average of the precision and recall for a information retrieval task, in this case the semantic processing retrieves parts of the semantic representation.

5.1 Exploratory and Baseline Experiments

To identify the right MLN model, first it was necessary to identify the formulae to build such a MLN model. We experimented with different combinations using the development part of the corpus and a 10-part cross validation. The best two models plus the *base* model consisting only of the *base* formula were the models used in this work (see section 3.2 for the formulae). We call this experiments the *exploratory* experiments. In all these cases, the formulae which included the contextual information was not included.

After identifying the models with a good performance, we performed the *baseline* experiments. These have the goal to establish the performance without using the contextual information. We perform two main experiments, first using the rule-based approach. For this case, we modified the version presented in subsection 3.1 to try with all of the possible regular expressions for each input utterance. For both approaches, we experiment with the manual and automatic transcriptions.

5.2 Context Experiments

The *context* experiments have the goal to measure the effect of using the context. For the rule-based case we follow the method presented in subsection 3.1. On the other hand, for the data-oriented approach we include the expectation rule with the three best models from the *baseline* experiments.

Also for these experiments, for both approaches, we experimented with the manual and automatic transcriptions.

6 Results

Table 2 presents the results of the *exploratory* experiments. We found that from the possible combinations the *base+bigram* (BB) and the *base + bigram + POS + window₁* (BB+) performed the best. However, when compared with the performance of the base model (B) we found that they had a lower performance. We believe this is because, for the case of bigrams, there are not many instances of them, since the utterances in this corpus are rather short.

Table 3 presents the results for the performance of the semantic processing without contextual information. First, notice that the data-oriented approaches have a better performance than the rule based. For the case of manual transcriptions it is 16.24% better and 4.64% for the automatic. Also notice the fall in the performance from manual to automatic transcriptions for all the approaches and models. For instance, the base model (B) drops 44.72%. This speaks of the challenge for automatic transcriptions in real life applications.

Table 4 presents the results for the performance of the semantic processing with contextual information. For these results we also observe a drop in the performance from manual to automatic transcriptions. An unexpected result was the great performance that the rule based approach has when dealing with manual transcriptions using contextual information. However, this performance suffers a 39.00% drop when dealing with automatic transcriptions. For the case

Table 2. Results of the exploration of models using the ML

	Accuracy	<i>f</i> -score
base (B)	85.7%	89.07%
<i>base+POS</i>	84.33%	88.07%
<i>base+POS+window₋₁</i>	81.29%	84.84%
<i>base+POS+window₊₁</i>	84.28%	87.91%
<i>base+POS+window_{-1,+1}</i>	84.11%	87.71%
base+bigram (BB)	84.61%	88.35%
<i>base+bigram+POS</i>	84.50%	88.31%
<i>base+bigram+POS+window₋₁</i>	83.73%	86.87%
base+bigram+POS+window₊₁(BB+)	84.66%	88.0%
<i>base+bigram+POS+window_{-1,+1}</i>	84.05%	87.89%

Table 3. Evaluation of ruled and data-oriented approaches **without** contextual information. Both, using manual and automatic transcriptions.

Model	Accuracy	<i>f</i> -score
Manual transcriptions		
Word enhanced spotting	74.80%	74.80%
MLN B	86.95%	91.11%
MLN BB	85.65%	89.59%
MLN BB+	84.88%	88.06%
Automatic transcriptions		
Word enhanced spotting	45.93%	45.04%
MLN B	48.06%	53.51%
MLN BB	45.15%	51.48%
MLN BB+	45.73%	53.23%

of automatic transcriptions the superiority of the data-oriented approaches can be noticed. The base model outperforms by 22.37% rule base approach.

Comparing Tables 3 and 4 we notice the advantage of using contextual information. In all the cases, the use of the expectation to guide the semantic processing improves the performance. In the case of automatic transcriptions the base model outperforms 33.73% of the performance of the same model but without contextual information. This result represents what we will expect with a functioning dialogue system.

7 Conclusions

In this paper we have explored the use of contextual information in the semantic processing task. In particular, we use the expectations of the system about what is going to be said by the user of a dialogue system in order to guide the semantic processing.

We found that the use of contextual information improves the performance of the task. We tested for both rule and data-oriented based approaches, and for

Table 4. Evaluation of ruled and data-oriented approaches with contextual information. Both, using manual and automatic transcriptions.

Model	Accuracy	<i>f</i> -score
Manual transcriptions		
Word enhanced spotting	85.46%	88.22%
MLN B	86.24%	89.69%
MLN BB	84.49%	88.20%
MLN BB+	85.85%	89.26%
Automatic transcriptions		
Word enhanced spotting	52.13%	60.37%
MLN B	62.79%	70.86%
MLN BB	61.62%	67.76%
MLN BB+	60.65%	66.83%

both cases there was an improvement. We also found, that a simple word spotting technique enhanced with regular expression is good option when dealing with manual transcriptions. However, the data-oriented approach is more adequate to deal with automatic transcriptions.

In the case of the data-oriented approach, we used Markov Logic to model the task. We found that the simple model, the *base* was the best option. We believe this is the case since the average length of the utterance of the corpus used was to small. As future work, we will aim to use larger utterances. We are also interested into test the portability of this approach to different domains.

References

1. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research* 3, 951–991 (2003)
2. Richardson, M., Domingos, P.: Markov Logic Networks. *Machine Learning* 62, 107–136 (2006)
3. Mann, W.C., Thompson, S.: Rhetorical structure theory: Towards a functional theory of text organization. *Text* 8(3), 243–281 (1988)
4. Miller, S., Stallard, D., Bobrow, R., Schwartz, R.: A fully statistical approach to natural language interfaces. In: Proc. of the Association for Computational Linguistics (1996)
5. Meza, I., Lemmon, O.: Using dialogue context to improve parsing performance in dialogue systems. In: Proceedings of IWCS-6, pp. 21–27 (2005)
6. Meza, I.V., Salinas, L., Venegas, E., Castellanos, H., Chavarría, A., Pineda, L.A.: Specification and Evaluation of a Spanish Conversational System Using Dialogue Models. In: Kuri-Morales, A., Simari, G.R. (eds.) IBERAMIA 2010. LNCS, vol. 6433, pp. 346–355. Springer, Heidelberg (2010)
7. Pineda, L.A., Meza, I., Salinas, L.: Dialogue Model Specification and Interpretation for Intelligent Multimodal HCI. In: Kuri-Morales, A., Simari, G.R. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 20–29. Springer, Heidelberg (2010)
8. Riedel, S.: Improving the accuracy and efficiency of map inference for markov logic. In: UAI 2008: Proceedings of the Annual Conference on Uncertainty in AI (2008)
9. Zettlemoyer, L., Collins, M.: Learning Context-dependent Mappings from Sentences to Logical Form. In: Proceedings of ACL-IJCNLP (2009)

A Statistics-Based Semantic Textual Entailment System

Partha Pakray¹, Utsab Barman¹, Sivaji Bandyopadhyay¹, and Alexander Gelbukh²

¹ Computer Science and Engineering Department,
Jadavpur University, Kolkata, India

² Center for Computing Research, National Polytechnic Institute,
Mexico City, Mexico
{parthapakray, utsab.barman.ju}@gmail.com,
sbandyopadhyay@cse.jdvu.ac.in
www.gelbukh.com

Abstract. We present a Textual Entailment (TE) recognition system that uses semantic features based on the Universal Networking Language (UNL). The proposed TE system compares the UNL relations in both the text and the hypothesis to arrive at the two-way entailment decision. The system has been separately trained on each development corpus released as part of the Recognizing Textual Entailment (RTE) competitions RTE-1, RTE-2, RTE-3 and RTE-5 and tested on the respective RTE test sets.

Keywords: textual entailment; Universal Networking Language; Recognizing Textual Entailment data sets.

1 Introduction

Recognizing Textual Entailment (RTE) is one of recent challenges in Natural Language Processing (NLP) [1]. Textual Entailment is defined as a directional relationship between two text expressions, denoted by T, which is the entailing “Text” and H, the entailed “Hypothesis”. T entails H if the meaning of H can be inferred from the meaning of T, as would typically be interpreted by people.

Textual Entailment has many applications in NLP tasks. For example, in Summarization (SUM), a summary should be entailed by the text; Paraphrases (PP) can be seen as mutual entailment between two texts; in Information Extraction (IE), the extracted information should also be entailed by the text; in Question Answering (QA), the answer obtained for a question after the Information Retrieval (IR) process must be entailed by the supporting snippet of text.

To date there have been six Recognizing Textual Entailment (RTE) competitions: RTE-1 [2] in 2005, RTE-2 [3] in 2006 and RTE-3 [4] in 2007, RTE-4 [5] in 2008, RTE-5 [6] in 2009 and RTE-6 [7] in 2010. In every new competition several new features of RTE were introduced. In 2010, Parser Training and Evaluation using Textual Entailment [8] was organized at SemEval-2. Our work has been tested on the data sets released in those competitions (except RTE-4, where no separate development data was released, and the most recent RTE-6).

Our system is based on the ideas of [9] but has been optimized for the entailment YES/NO decision using the development set. Our results are better than in [9].

The paper is organized as follows. Related works are described in Section 2. Section 3 explains the UNL Expressions. Section 4 presents our semantic based RTE system architecture. The experiments carried out on the development and test data sets, as well as their results, are described in Section 5. Section 6 includes discussion and error analysis. Conclusions are drawn in Section 7.

2 Related Work

In the various RTE Challenges, several methods have been applied to the textual entailment task. Most of these systems use some sort of lexical matching. A number of systems represent the texts as parse trees (e.g., syntactic or dependency trees) before the actual task. Some of the systems use semantic relations (e.g., logical inference, Semantic Role Labeling) for solving the text and hypothesis entailment problem.

The VENSES system [10] (Venice Semantic Evaluation System) is organized as a pipeline of two subsystems: the first is a reduced version of GETARUN, their system for Text Understanding. The output of the system is a flat list of head-dependent structures with Grammatical Relations and Semantic Roles labels. The version [11] presented at RTE-3 uses a linguistically based approach for semantic inference.

A syntax-driven semantic analysis system is presented in [12] and uses the notion of atomic proposition as its main element for entailment recognition. The idea is to find the entailment relation in the sentence pairs by comparing the atomic propositions contained in the text and hypothesis sentences.

The GROUNDHOG system [13] for recognizing textual entailment utilizes a classification-based approach to combine lexico-semantic information derived from text processing applications along with a large collection of paraphrases automatically acquired from the WWW.

A baseline system for modeling textual entailment is presented in [14] that combines deep syntactic analysis with structured lexical meaning descriptions in the FrameNet paradigm. Textual entailment is approximated by degrees of structural and semantic overlap of text and hypothesis, which they measure in a match graph.

A Machine Learning approach with Support Vector Machines and AdaBoost to deal with the RTE challenge is presented in [15]. Its authors perform a lexical, syntactic and semantic analysis of the entailment pairs. From this information they compute a set of semantic based distances between sentences.

The system presented in [16] generates paraphrases of semantically labeled input sentences using the semantics and syntax encoded in FrameNet, a freely available lexico-semantic database. The algorithm generates a large number of paraphrases with a wide range of syntactic and semantic distances from the input.

The system presented in [17] maps premise and hypothesis pairs into an abstract knowledge representation (AKR) and then performs entailment and contradiction detection on the resulting AKRs.

The system reported in [18] presents a new data structure, termed compact forest, which allows efficient generation and representation of entailed consequent represented as parse trees.

The system presented in [19] performs semantic interpretation of the sentence pairs. It tries to determine if the logic for the H sentence subsumes some inference-elaborated version of the T sentence, using WordNet and the DIRT paraphrase database as its sources of knowledge.

The Monte Carlo Pseudo Inference Engine for Text system [20] addresses the RTE problem in a new theoretic framework for robust inference and logical pattern processing based on integrated deep and shallow semantics.

The Boeing Language Understanding Engine [21] can be viewed as comprising of three main elements: parsing, WordNet and DIRT, built on top of a simple baseline of bag-of-words comparison.

A joint syntactic-semantic representation to better capture the key information shared by the T-H pair is proposed in [22]. The system applies a co-reference resolver to group cross-sentential mentions of the same entities together.

In [23], the entailment recognition is attempted by computing shallow lexical deductions and richer inferences based on semantics. The system works on WordNet, detection of negation terms, named entity recognition, verbs implications and frame semantic analysis.

3 Universal Networking Language

Universal Networking Language (UNL) [24] is an artificial language that expresses information or knowledge in the form of semantic network with hyper-nodes. The applications of UNL have been found in the domains of Machine Translation, Information Retrieval and Multilingual Document Generation [26].

UNL consists of a set of Universal Words (UW), relations and attributes. Universal Words are concepts. The binary relationships among the universal words in the sentence are specified as relations. Attributes are properties of the Universal Words. UNL semantic network includes a set of binary relations and each binary relation relates the two Universal Words that hold the relation. A binary relation of UNL has the following format:

$$\langle \text{Relation} \rangle (\langle \text{UW1} \rangle, \langle \text{UW2} \rangle).$$

The process of representing natural language sentences in UNL graphs is called *enconverting*, and the process of generating natural language sentences out of UNL graphs is called *deconverting*. An EnConverter is a language independent parser, which provides a framework for morphological, syntactic, and semantic analysis synchronously. A DeConverter is a language independent generator, which provides a framework for syntactic and morphological generation synchronously.

4 System Description

Our semantic-based textual entailment system accepts pairs of text snippets (T and H) in the input and outputs a binary value: “YES” if the text T entails H and “NO” otherwise. It is a point-based scoring system, which takes decision based on the scores

of the UNL relations of a T-H pair. The score of the T-H pair is compared with a threshold value (which we calculated empirically by observation of the results) to obtain the entailment decision.

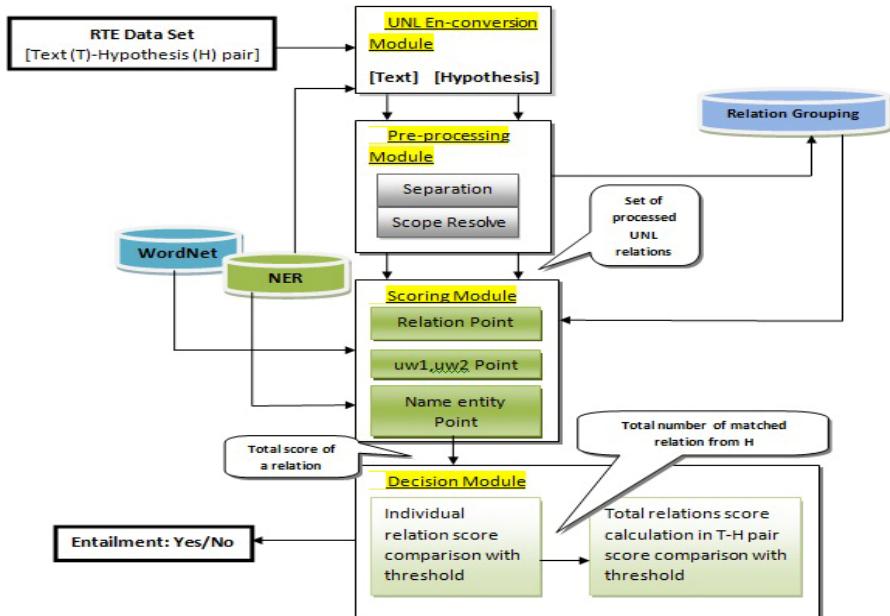


Fig. 1. Semantic Textual Entailment System

4.1 UNL En-Conversion Module

The T-H pairs are converted into UNL expression using the UNL En-Converter (www.unl.ru). An example of UNL expression for a hypothesis from RTE-5 Development data is shown in Figure 2.

```

[S:00]
{orgen} Pfizer is accused of murdering 11 children {/org}
{unl}
obj(accuse(icl>do,equ>charge,cob>abstract_thing,agt>person,obj>person).@entry
.@present,pfizer.@topic)
qua:01(child(icl>juvenile>thing).@pl,11)
obj:01(murder(icl>kill>do,agt>thing,obj>living_thing).@entry,child(icl>juvenile
>thing).@pl)
cob(accuse(icl>do,equ>charge,cob>abstract_thing,agt>person,obj>person).@entr
y.@present,:01)
{/unl} [/S]
  
```

Fig. 2. Example of a hypothesis of RTE-5 Development Data in UNL

4.2 Pre-processing Module

Separation. From the different UNL graphs of T and H, individual UWs are extracted using regular expressions. The regular expressions that are used to extract the individual UWs are as follows:

For UW1: [#] + [-a-z0-9R:._-&=*=^"~\\"\\ + [\s]] + [\W(.)]

For UW2: [,] + [-a-z0-9R:._- &=*=^"~\\"\\ + [\s]] + [\W(.#]

The relation name, scope ID, constraint list and attribute list are separated from a single UNL relation graph. All relations are put up into a logical set in some specific format as per our system requirement:

[Relation Name] [Relation Scope ID] {[UW1][UW1 Scope id], [UW2][UW2 Scope id]}

Extraction of different components from a single UNL relation graph is done by using regular expressions.

Scope Resolution. The specific task at this step is to resolve the scope ID of UNL relations. For example, consider a hypothesis of the RTE-5 development set in UNL relation format shown in Figure 2. In the fourth relation Cob we find a scope ID ‘:01’ in the place of UW2 that specifies the relation between the present UNL graph and the other UNL graph specifying UW2. In the sentence the main subject / noun in focus is ‘Pfizer’ and the other noun ‘children’ in the predicate part has less focus. However, the second noun is directly affected by the action of first one that has occurred in parallel. The UNL specification defines the relation Cob as “defines a thing that is directly affected by an implicit event done in parallel or an implicit state in parallel.” The result is shown in Figure 3.

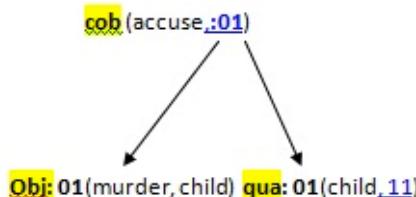


Fig. 3. UNL graph of Cob relation

Relation Grouping. Relation grouping step groups UNL relations that are semantically identical [25]. UNL relations are hyper semantic network and UW1, UW2 are two nodes of the graph having a relation specified by the relation name in the UNL graph. The strategy is based on the thematic roles of different relations. Table 1 shows different relation groups and the set of relations in each group.

4.3 Scoring Module

The scoring module calculates the score between a pair of T-H UNL relations. The module assigns points to each relation pair using certain set of rules. The rules are Relation grouping rule, UW Rule, and Name Entity Rule.

Table 1. UNL Relation grouping

Group Name	Relations
Agent	<i>Agt, cag, aoj, cao, ptn</i>
Object	<i>Obj, cob, opl, ben</i>
Place	<i>Plc, plf, plt</i>
Instrument	<i>Ins, met</i>
State	<i>Src, gol, via</i>
Time	<i>Tim, tmf, tmt, dur</i>
Manner	<i>Man, bas</i>
Logical	<i>And, or</i>
Concept	<i>Equ, icl, iof</i>
Cause	<i>Con, pur, rsn</i>
Sequence	<i>Coo, seq, cnt, mod, nam, per, pof, pos, qua</i>

Relation grouping rule checks whether the two UNL relations or UNL graphs, one from the text and another from the hypothesis, are in same relation group. If so, it is considered as a match and one point is assigned to the relation pair.

UW rule Checks whether UWs in the two matched UNL relations or graphs are same or they belong to the same synset, i.e., refer to same meaning. We have used the riWordNet (www.rednoise.org/rita/wordnet/documentation/index.htm) for synset matching. One point is considered as the score for each UW match. In case of the presence of scope id in the place of any UW the comparison will be done from the UW list created by the scope resolve module.

Name Entity Rule: If there are n Named Entities in H and m Named Entities in T and k be the number of Named Entities that are present in both H and T, then the point for Named Entity will be calculated as the fraction of the Named Entities in Hypothesis that match, i.e., k/n .

The composite score of a T-H pair is calculated as follows:

$$\begin{aligned} \text{Total Score (TS)} &= \text{Relation Match Point (RMP)} + \\ &\quad \text{UW1 point (UW1)} + \text{UW2 point (UW2)} + (k/n) \end{aligned} \quad (1)$$

4.4 Decision Module

Individual Relation Pair decision: The total score of individual relation pair is calculated using the equation (2). The maximum value of the total score (TS_{\max}) for each individual relation pair is calculated as (4). This gives for our example:

$$TS = RMP + UW1 + UW2 + (k/n) \quad (2)$$

$$TS_{\max} = RMP_{\max} + UW1_{\max} + UW2_{\max} + (k/n)_{\max} \quad (3)$$

$$RMP_{\max} = Uw1_{\max} = Uw2_{\max} = (k/n)_{\max} = 1 \quad (4)$$

$$\text{Hence } TS_{\max} = 4 \quad (5)$$

The minimum value of the total score (TS_{\min}) for each individual relation pair has been observed as 3.5 from the training sets of the various RTEs. If the total score of a relation pair falls between 3.5 and 4, the relation pair is considered as a match.

Final Decision by Total Relations Score Calculation: Let H_n be the numbers of UNL relations in hypothesis and T_n , the numbers of UNL relations in text. Then the number of matched relation pairs (M_n) is identified. The final score (FS) for the T-H pair is calculated as (M_n / H_n) . It has been observed from the training sets of the various RTEs that the minimum value of FS is 0.96 for the T-H pair to be entailed. Hence, if the FS score for a T-H pair is 0.96 or above then the T-H pair is considered as entailed.

5 Experimental Results

We used the following data sets: RTE-1 development set and test set, RTE-2 development set and test set, RTE-3 development set and test set, RTE-4 test set and RTE-5 main development set and test set to deal with the two-way classification task.

Table 2. Evaluation Results

RTE Data	Entailment decision	Entailments in Gold Standard	Correct entailments by our system	Total entailments by our system	Precision (%)	Recall (%)	F-Score (%)
RTE-1 Dev. Set (1)	YES	143	63	102	0.44	0.61	0.51
	NO	144	104	185	0.72	0.56	0.63
RTE-1 Test set	YES	400	172	346	0.43	0.49	0.46
	NO	400	262	454	0.65	0.57	0.60
RTE-2 Dev. Set	YES	400	170	272	0.42	0.62	0.50
	NO	400	297	528	0.74	0.56	0.64
RTE-2 Test Set	YES	400	262	460	0.65	0.56	0.60
	NO	400	212	340	0.53	0.62	0.57
RTE-3 Dev. Set	YES	412	189	278	0.45	0.67	0.54
	NO	388	299	522	0.77	0.57	0.65
RTE-3 Test Set	YES	410	285	409	0.69	0.69	0.69
	NO	390	262	391	0.67	0.67	0.67
RTE-4 Test Set	YES	500	352	522	0.67	0.70	0.68
	NO	500	320	478	0.66	0.64	0.65
RTE-5 Dev. Set	YES	300	176	300	0.58	0.58	0.58
	NO	300	175	300	0.58	0.58	0.58
RTE-5 Test Set	YES	300	169	318	0.56	0.53	0.54
	NO	300	150	282	0.50	0.53	0.51

The RTE-1 has two development sets, one consisting of 287 text-hypothesis pairs and another consisting of 287 text-hypothesis pairs. The RTE-1 test set, the RTE-2, RTE-3, and RTE-5 development sets and test sets consist of 800 text-hypothesis pairs each. At RTE-4, no development set was provided, as the pairs proposed were very similar to the ones contained in the RTE-3 development and test sets. The RTE-4 test set consisted of 1000 text-hypothesis pairs.

Four applications, i.e., IE, IR, QA and SUM, were set as the contexts for the pair generation. The length of the H's was the same as in the past data sets (RTE-3); however, the T's were generally longer.

6 Error Analysis

The system has some limitations in the matching of universal words (UW). Two UWs may belong to different layers of geographical hierarchy, one may be a named entity and another may be a description of the named entity itself or one may be an anaphor and the other may be the antecedent. In all such cases, the current level of UW match will not succeed. Some related Text-Hypothesis pairs with the above situations are shown below.

Geographical Hypothesis:

T: *He lives in West Bengal.*

H: *He lives in India.*

Here all the UNL relations are same, West Bengal and India both are in plc relation but the system doesn't have enough capabilities to identify the West Bengal is in India.

Knowledge of Name Entities Hypothesis:

T: *Madona has 3 children.*

H: *The rock star has 3 children.*

The system also does not have enough resource so that it can identify Madona is a rock star.

Pronoun Replacement Hypothesis:

T: *Albert Einstein discovered theory of relativity.*

H: *He discovered the theory of relativity.*

If the pronoun in text is replaced with proper noun/name entity then the system will increase its performance.

7 Conclusions

Our results show that a semantic-based approach appropriately tackles the textual entailment problem. Experiments have been initiated for a semantic and syntactic based RTE task.

Acknowledgements. The work was done under partial support of the DST India—CONACYT Mexico project “Answer Validation through Textual Entailment”, European Project WIQ-EI 269180, IPN (COFAA, SIP 20100773 & 20113295),

CONACYT (SNI and Sabbatical program), and Mexico City Government project ICYT PICCO10-120.

References

1. Ledeneva, Y., Sidorov, G.: Recent Advances in Computational Linguistics. *Informatica. International Journal of Computing and Informatics* 34(2010), 3–18 (2010)
2. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the First PASCAL Recognizing Textual Entailment Workshop (2005)
3. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy (2006)
4. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The Third PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic (2007)
5. Giampiccolo, D., Dang, H.T., Magnini, B., Dagan, I., Cabrio, E.: The Fourth PASCAL Recognizing Textual Entailment Challenge. In: TAC 2008 Proceedings (2008), <http://www.nist.gov/tac/publications/2008/papers.html>
6. Bentivogli, L., Dagan, I., Dang, H.T., Giampiccolo, D., Magnini, B.: The Fifth PASCAL Recognizing Textual Entailment Challenge. In: TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland, USA (2009)
7. Bentivogli, L., Clark, P., Dagan, I., Dang, H.T., Giampiccolo, D.: The Sixth PASCAL Recognizing Textual Entailment Challenge. In: TAC 2010 Notebook Proceedings (2010)
8. Yuret, D., Han, A., Turgut, Z.: SemEval-2010 Task 12: Parser Evaluation using Textual Entailments. In: Proceedings of the SemEval-2010 Evaluation Exercises on Semantic Evaluation (2010)
9. Pakray, P., Poria, S., Bandyopadhyay, S., Gelbukh, A.: Semantic Textual Entailment Recognition using UNL. *Polibits* (43), 23–27 (2011)
10. Delmonte, R., Tonelli, S., Boniforti, M.A.P., Bristot, A., Pianta, E.: VENSES – a Linguistically-Based System for Semantic Evaluation. In: Proceedings of the First PASCAL Recognizing Textual Entailment Workshop (2005)
11. Delmonte, R., Bristot, A., Boniforti, M.A.P., Tonelli, S.: Entailment and Anaphora Resolution in RTE3. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic (2007)
12. Akhmatova, E.: Textual Entailment Resolution via Atomic Propositions. In: Proceedings of the First PASCAL Recognizing Textual Entailment Workshop (2005)
13. Hickl, A., Bensley, J., Williams, J., Roberts, K., Rink, B., Shi, Y.: “Recognizing Textual Entailment with LCC’s GROUNDHOG System”. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy (2006)
14. Burchardt, A., Frank, A.: Approaching Textual Entailment with LFG and FrameNet Frames. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy (2006)
15. Ferres, D., Rodriguez, H.: Machine Learning with Semantic-Based Distances Between Sentences for Textual Entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic (2007)

16. Ellsworth, M., Janin, A.: Mutaphrase: Paraphrasing with FrameNet. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic (2007)
17. Bobrow, D.G., Condoravdi, C., Crouch, R., de Paiva, V., Karttunen, L., King, T.H., Laird, R., Price, L., Zaenen, A.: Precision-focused Textual Inference. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic (2007)
18. Bar-Haim, R., Berant, J., Dagan, I., Greental, I., Mirkin, S., Shnarch , E., Szpektor, I.: Efficient Semantic Deduction and Approximate Matching over Compact Parse Forests. The Fourth PASCAL Recognizing Textual Entailment Challenge. In: TAC 2008 Proceedings (2008), <http://www.nist.gov/tac/publications/2008/papers.html>
19. Clark, P., Harrison, P.: Recognizing Textual Entailment with Logical Inference. The Fourth PASCAL Recognizing Textual Entailment Challenge. In: TAC 2008 Proceedings (2008), <http://www.nist.gov/tac/publications/2008/papers.html>
20. Bergmair, R.: Monte Carlo Semantics: MCPIET at RTE4. In: TAC 2008 Proceedings (2008), <http://www.nist.gov/tac/publications/2008/papers.html>
21. Clark, P., Harrison, P.: An Inference-Based Approach to Recognizing Entailment. The Fifth PASCAL Recognizing Textual Entailment Challenge. In: TAC 2009 Workshop Notebook, Gaithersburg, Maryland, USA (2009)
22. Wang, R., Zhang, Y., Neumann, G.: A Joint Syntactic-Semantic Representation for Recognizing Textual Relatedness. The Fifth PASCAL Recognizing Textual Entailment Challenge. In: TAC 2009 Workshop Notebook, Gaithersburg, Maryland, USA (2009)
23. Fernandez, O., Munoz, R., Palomar, M.: Alicante University at TAC 2009: Experiments in RTE. The Fifth PASCAL Recognizing Textual Entailment Challenge. In: TAC 2009 Workshop Notebook, Gaithersburg, Maryland, USA (2009)
24. UNDL Foundation. Universal networking language (unl) specifications, edition 2006 (August 2006), <http://www.udl.org/unlsys/unl/unl2005-e2006/>
25. Ishizuka, M.: A Solid Foundation of Semantic Computing toward Web Intelligence. School of Information Science and Technology, http://www.jst.go.jp/sicp/ws2010_austria/presentation/presentation_14.pdf
26. Cardeñosa, J., Gelbukh, A., Tovar, E. (eds.): Universal Networking Language: Advances in Theory and Applications. Research on Computing Science, vol. 12, p. 443 (2005), [http://www.cicling.org/2005/UNL-book/..](http://www.cicling.org/2005/UNL-book/)

Semantic Model for Improving the Performance of Natural Language Interfaces to Databases

Rodolfo A. Pazos R., Juan J. González B.,
and Marco A. Aguirre L.

Instituto Tecnológico de Ciudad Madero, Mexico
r_pazos_r@yahoo.com.mx, jjgonzalezbarbosa@hotmail.com,
marco@marcoaguirre.com.mx

Abstract. Despite the fact that since the late 60s many Natural Language Interfaces to Databases (NLIDBs) have been developed, up to now many problems continue, which prevent the translation process from natural language to SQL to be totally successful. Some of the main problems that have been encountered relate to 1) achieving domain independence, 2) the use of words or phrases of different syntactic categories for referring to tables and columns, and 3) semantic ellipsis. This paper introduces a new method for modeling databases that includes relevant information for improving the performance of NLIDBs. This method will be useful for solving many problems found in the translation from natural language to SQL, using a database model that contains linguistic information that provides more semantic information than that found in conventional database models (such as the extended entity-relationship model) and those used in previous NLIDBs.

Keywords: Natural Language Interfaces to Databases, semantic modeling.

1 Introduction

In the last decades information has played an important role in our daily life, most people request information before making some important decision. Currently, the largest sources of information are stored in databases.

In order for a user to obtain information from a database, he/she needs formulate a query in such a way that the computer interprets and generates the correct answer (usually in a query language such as SQL). Unfortunately, only computing professionals can formulate queries in such way.

The normal way in which people request information is through questions in natural language, but computers can not directly understand this kind of language. Therefore, natural language interfaces to databases (NLIDBs) emerge as an alternative to several problems that occur in systems for obtaining information, since they permit users accessing information stored in databases through a query in natural language.

The need for NLIDBs has become important nowadays, because many users request accessing information from different types of systems [1] [2]. Most of the

NLIDBs that have achieved good results (around 95% of queries correctly translated) have been those that are domain dependent; i.e., those that can be used to query only one database. Since the 90s much effort has been devoted to developing domain independent NLIDBs; i.e., those that can be used for different databases; however, the limitations and deficiencies they have are worth noticing, which are reflected in a success rate of 80-90%, which is unsatisfactory for critical applications.

Some commercial NLIDBs exemplify this situation: for example, LanguageAccess (developed by IBM) was discontinued, English Query (developed by Microsoft) was included for the last time in SQL Server ver. 8.0, released in 2000, and English Wizard (developed by Linguistic Technology Corporation) was discontinued several years ago [3].

The NLIDBs developed up to now have several problems. A survey of the literature on NLIDBs as well as the tests of some prototypes, reveal that the systems developed up to now have limitations in two aspects [1]: it is difficult to port them from one domain to another (i.e., from a database to another), and they fail with some queries, either because they misinterpret the queries or because they are unable to respond even though there exists an answer to the query.

Understanding a query in natural language is undoubtedly a hard task for a computer due to several problems, such as semantic ellipsis and the use of words of different syntactic categories (nouns, verbs, adjectives and prepositions) for referring to tables and columns of the databases.

In this paper we propose a new semantic model for databases based on an analysis of queries to three databases (ATIS [4], Northwind and Pubs), from which many problems present in those queries were identified and classified (Section 3). The modeling methodology on which our model is based is called semantically enriched model (SEM), and it aims at including semantic information (besides structural information of the database) that helps the semantic analysis of queries. This new model will constitute a key element for designing a data dictionary for a new version of a NLIDB developed by us (whose description can be found in [5] [6]).

2 Background

The use of natural language interfaces for accessing databases dates back to the late 60s and the early 70s. Though many dozens of NLIDBs have been developed since then, in this survey we consider only the most recent NLIDBs that include/exclude three characteristics: domain independence, database modeling and learning capacity.

CoBase [7] generates candidate queries based on the user input through a search algorithm in a semantic graph. This search is organized according to a metric of probabilistic information. This system uses an incremental method that helps users to formulate complex queries through a series of simple queries.

NLPQC [8] is a NLIDB that is used in the virtual library CINDI. This system is constituted by a preprocessor and a run-time module. The preprocessor constructs a conceptual knowledge base of the database schema using Wordnet, which is used at runtime for a semantic analysis of the input according to several predefined templates and for constructing the SQL expression.

WYSIWYM [9] uses a semantic graph inspired by the one described in [7]. The NLIDB helps users to formulate queries using predefined frames through a query interface where the user completes the frame through a list of search values for the columns involved in the query.

C-PHRASE [10] has an authoring tool for editing the NLIDB data dictionary of the NLIDB for customizing it for a specific domain. The query analysis centers mainly in recognizing nominal phrases. C-PHRASE uses a semantic grammar based on rules and patterns, which are used for translating the query to SQL.

In [11] a NLIDB that uses Sequence and Tree Kernels (STKs) and some variants is presented. In this system sets of data are constructed that consist of pairs of queries in natural language and SQL expressions. The NLIDB represents these pairs through syntactic trees, and uses kernel functions and support vector machines for carrying out the translation from natural language to SQL.

From the surveyed literature, it can be observed that most of the NLIDBs have been tested with one or a few databases, which does not fully prove their domain independence. On the other hand, only a few NLIDBs use dialogue managers for clarifying queries and learning capabilities, which are features that may improve the performance of the NLIDB.

Most of the NLIDBs developed up to now have focused on solving existing complex problems; however, for solving them developers have implemented NLIDBs that involve some degree of difficulty in their use, and very few have taken care in defining a formal model of the database that helps achieve a better translation processing.

Table 1 shows a comparison of the NLIDBs described above with respect to three characteristics: domain independence, use of a formal DB modeling process, learning capability, dialogues for clarifying queries and success rate (i.e., the percentage of correctly translated queries). The second column shows the characteristics that the new version of our NLIDB aims at.

Table 1. Comparison of characteristics of important NLIDBs

Characteristics	NLIDBs					
	New version of our NLIDB	(1999) CoBase	(2005) NLPQC	(2006) WYSIWYM	(2008) C-PHRASE	(2010) STK
Domain independence	✓	?	L	L	L	✓
DB modeling	✓	✓	L	L	L	✗
Learning capability	✓	✗	✗	✗	✓	✓
Clarification dialogues	✓	✗	✗	✗	✗	✗
Success rate	95%	?	?	?	86%	76%

Key: ✓ Includes feature, ? No evidence, L Limited, and ✗ Lacks this feature.

At this point it is important to mention that the current version already has the following characteristics: domain independence, clarification dialogues and a success rate of 79-89% [6]. With the inclusion of the new features we aim at increasing the success rate to 95%.

The preceding and current versions of our NLIDB are described in [5] and [6]. The main characteristic of the system described in [5] consists of dealing with domain independence. In order to achieve this, it uses a preprocessor that automatically generates a domain dictionary and a translation technique that involves nouns, prepositions and conjunctions. It is important to mention that such version did not have a data dictionary with semantic information (such as the one proposed here). The second version, described in [6], includes domain-independent dialogue processes, which were designed for their use with any relational database, and were based on a typification of problems in queries that involves most of the cases found. In order to solve the problems present in those two versions, a new database model has been proposed that includes information necessary for solving the remaining problems (those described in Section 3), which is complemented with an architecture based on functional layers.

Despite the competitive performance of our NLIDB (compared with other state-of-the-art systems), it still does not successfully deal with some problems (such as those mentioned in the following section), which made it necessary to think in a major overhaul of the system based on a new data dictionary and a layered architecture.

3 Problems in Queries

From an analysis carried out on query corpora involving three databases, four general types of problems (i.e., those that usually are found in queries for most databases) were identified and classified:

1. The use of words or phrases of different syntactic categories (such as nouns, verbs, adjectives, and prepositions) for referring to tables or columns of the database.
2. Semantic ellipsis that occurs when words that are necessary for clearly understanding the query are omitted.
3. Covering of the capabilities of SQL, such as involving several tables of the database and the use of aggregate functions.
4. Other type of problems related to human errors, such as nonexistent information in the database, words that indicate imprecise values, etc.

The classification obtained and some examples are shown in Table 2.

Table 2. Types of problems that occur in queries

Cases	Problems
1	Use of words or phrases of different syntactic categories
1.1	Use of nouns or nominal phrases for referring to tables or columns. Example: List number of seats on D9S.
1.2	Use of verbs or verbal phrases for referring to tables or columns. Example: What time does flight 102136 leave ATL to DFW?

Table 2. (*continued*)

- 1.3 Use of prepositions or prepositional phrases for referring to tables or columns.
 Example: Give me an economy class flight **from** DFW **to** BWI one-way.
- 1.4 Use of adjectives or adjectival phrases for referring to tables or columns.
 Example: How **fast** can the Concorde fly?
- 1.5 Use of temporal adverbs.
 Example: List fares for all flights leaving **after** twelve o'clock noon from BOS to BWI.
- 1.6 Use of conjunctions.
 Example: Flights exiting Fort Worth **and** entering Dallas.
- 2 Semantic ellipsis**
- 2.1 Lacking information of tables or columns.
 Example: List **fares** for all flights leaving after twelve o'clock noon from BOS to BWI.
 Note: there exist two columns related to the word "fare": "one_way_cost" and "rnd_trip_cost", thus it is not clear which of these columns is being referred to.
- 2.2 Lacking information of tables or columns referred to by some value.
 Example: How much is **Delta** flight **539**?
- 2.3 Lacking information of tables or columns about the information requested.
 Example: **All flights** from ATL to SFO on Delta first class.
- 3 Covering of the capability of SQL**
- 3.1 Queries that involve several tables.
 Example: Give me an **economy** class **flight** from DFW to BWI **one-way**.
 Note: the columns referred to by "economy", "flight" and "one-way" belong to different tables.
- 3.2 Queries that involve aggregate functions.
 Example: Which flight from Philadelphia to Dallas has the **cheapest** fare?
- 4 Other type of problems**
- 4.1 Search values that involve two or more columns.
 Example: Give me the hire date of the employee **Margaret Peacock**.
 Note: the value "Margaret Peacock" involves two columns: "FirstName" and "LastName."
- 4.2 Search values constituted by two or more words.
 Example: Give me the postal code and city of the supplier "Exotic Liquids."
- 4.3 Incomplete search values.
 Example: What is the name of the store where is "**the busy**."
- 4.4 Inexistent tables or columns.
 Example: Get me a **date** on flight 294 leaving ATL to Washington.
 Note: the table or column "date" does not exist in the database.
- 4.5 Spacing, punctuation and formatting mistakes.
 Example: Flights between SFO and Dallas between noon and **5:00 P.M.**
 Note: the ATIS database uses the military format for time.
- 4.6 Imprecise search values.
 Example: Show me the Atlanta to Dallas flights in the **morning**.
-

It is important to mention that the classification carried out according to the problems found in the corpora can be applied to most databases. The proposed model is based on including the information needed for solving the aforementioned problems in order to improve the performance of NLIDBs.

4 Semantically Enriched Database Modeling

Existing data models (such as the Extended Entity-Relationship model (EER), the Unified Modeling Language (UML), etc.) include information about the real world and provide graphical techniques that are used for database design. Unfortunately, for some applications such as NLIDBs, their usefulness is insufficient because they do not include enough semantic information for an effective translation of queries from natural language to SQL.

Some of the main problems that have been found in NLIDBs relate to domain independence (system portability), grammatical problems and semantic ellipsis (concerning query translation). When a NLIDB is ported from one domain to another, it needs different domain information and grammatical structures; therefore, it must acquire somehow this information for constructing a domain dictionary that is necessary for the translation process.

Considering the limitations of the data models and the difficulty for solving the problems faced by NLIDBs, we propose the design of a software architecture based on a new database model. This new model does not intend to substitute existing database models, it rather intends to enrich their semantic information that permits to solve the problems faced by NLIDBs in query translation.

The *semantically enriched model* (SEM) represents the knowledge of a set of data in a specific domain. It is formally defined as,

$$SEM = (C, L),$$

where C is a set of concepts that belong to the database schema (entities, attributes and relationships) and L is the set of links among such concepts.

4.1 Grammatical Descriptors

In the analysis carried out with three query corpora, it has been observed that words that are used in queries for referring to entities or attributes belong to four syntactic categories; therefore, the inclusion of this information in the model will benefit the translation process. The categories that have been detected are the following:

- Nouns are used mainly to refer to entities or attributes.
- Verbs are used mainly to refer to relationships among entities.
- Prepositions are used to refer to attributes (mainly those that indicate locations or times).
- Adjectives qualify nouns (in Spanish they can occasionally be used instead of nouns).

As a result of this observation, a grammatical descriptor has been defined, which permits representing semantic information classified according to syntactic categories.

Grammatical descriptors G are syntactic categories that can be used for representing some of the schema concepts (entities or attributes). These tags can be of four types:

- Verbs or verbal phrases, V.
- Nouns or nominal phrases, N.
- Prepositions or prepositional phrases, P.
- Adjectives or adjectival phrases, Adj.

4.2 Entities

Like in the EER model, an *entity* (*e*) is defined as a concrete or abstract object that identifies an animated or unanimated being that exists in the real world. Unlike the EER model, in our model the representation of an entity includes a grammatical tag of type N, which defines the noun or nominal phrase that is usually used in queries for referring to the entity.

4.3 Attributes

Like the EER model, an *attribute* (*a*) is defined as a descriptive property of an entity. Each entity class E_i has a set of attributes $A_i = (a_{i1}, a_{i2}, \dots, a_{im})$. In the proposed model attributes are denominated using nouns and are associated to the entity, and they are referred to by a grammatical descriptor of type N. It is important to mention that both entities and attributes may optionally have grammatical tags of types V, P and Adj.

Attributes have a domain, which is defined as the set of possible values that an attribute may adopt, $v_{ij} \in Dom_{ij}$, where v_{ij} is the value of attribute a_{ij} , and Dom_{ij} is the set of atomic values permitted for the attribute.

In the set of attributes of each entity class E_i , there exists at least one attribute a_{ij} that is the identifier (or primary key) of the entity, which will be denoted by pk_i .

From the EER model only the following types of attributes will be included in our model: simple attributes, composite attributes, univalued attributes and derived attributes.

An attribute can be simple or composite. A *simple attribute* is one that can not be divided into smaller components, and conversely, a *composite attribute* is one that can be divided into several simple attributes. Most of the attributes are *univalued*, which means that they have a single value. The values of some attributes may be *derived*, which means that they are calculated from the values of other attributes.

4.4 Relationships

A *relationship* is an association between a pair of entity classes that is established through a link R . R is defined as the set of links r_i , where each one of these associates a pair of entities $[e_i, e_j]$. Typically, relationships are referred to by a grammatical descriptor of type V.

At this point it is important to remark that unlike the EER model, in our model relationships are defined differently: in the first model relationships may involve n entities (where $n = 1, 2, 3, \dots$); however for the purposes of our model relationships are defined only for pairs of entities.

A *role* is the function played by an entity in a relationship. According to this concept, entities can be of two types: *Agent entity* is an entity that normally plays the role of the subject in a sentence (in active voice) that describes the interaction between the two entities involved. *Patient entity* is an entity that normally plays the role of the object in a sentence (in active voice) that describes the interaction between the two entities involved. Generally, entities that represent people are considered agent entities; while those that represent unanimated things are considered patient entities.

Unlike the EER model, in our model relationships always have direction. Normally a relationship implies a *forward direction* (i.e., where the relationship points to) defined by the role played by its entities: from the agent entity to the patient entity. Additionally, for every direction between a pair of entities there exists an *inverse direction* that goes from the patient entity to the agent entity. Associated to such inverse direction, a sentence in passive voice can be defined that describes the interaction between the two entities involved. Occasionally, in a relationship there might exist two patient entities, and in this case one has to be chosen as agent for defining a forward direction.

A recursive relationship is an association between an entity class and itself. In this case the entity class plays different roles.

A relationship that associates a pair of entity classes $[E_i, E_j]$ involves a primary key of entity E_i and some corresponding attribute of entity E_j . This attribute is called *foreign key* and is denoted by fk_j . E_i is called referenced entity and its primary key is called referenced attribute; while E_j is called referencing entity and its foreign keys are called referencing attributes.

Similarly to the EER model, we define the notion of *cardinality*, which is defined as the number of entities that may participate in a relationship. According to cardinality, relationships are divided into three types: one to one (1:1), one to many (1:N), and many to many (N:M).

4.5 Specialization and Generalization

The definitions of specialization and generalization are defined similarly to those in the EER model, the only difference being in its representation. In our model, specialization is related to its entities using the verbal phase "it is a", and generalization is related to its entities using the verbal phase "it is a type of".

5 SEM Representation

This section defines the graphical representation of the main concepts of our proposed semantic modeling method.

An entity is graphically represented by a rectangle with two sections: the upper section is used for indicating the entity name, and the lower section indicates the attributes names. Each entity must have a unique name that distinguishes it from the other entities. Entities must have one grammatical descriptor, which will usually be of type N, and optionally grammatical descriptors of other types.

An attribute represents some property of the entity that occurs in all the instances of the class. Each attribute, must have one descriptor of type N, and optionally it may have grammatical descriptors of other types. In the graphical representation, names of attributes that are primary keys are written with underscoring; while those of attributes that are foreign keys are written with double underline. Names of derived attributes are written with segmented underscoring. The names of composite attributes are written in italics, and the constituent attributes are written as a list, each preceded by a bullet. Figure 1 shows an example of the graphical representation of entities and attributes.

Relationships are represented by a line that joins the entities involved with one arrow head that points to the referenced entity. There exist two alternatives for considering the direction of a relationship: one defined by the agent and patient entities involved and the other defined by the referenced entity and the referencing entities. In our model we use the first alternative because it is more useful for the semantic analysis of queries. The relationship description includes a box that contains a grammatical descriptor of type V and an arrow that indicates the forward direction of the relationship, and another grammatical descriptor of type V with an arrow for the inverse direction. Additionally, there is an arrow that links the box to its corresponding relationship. Finally, the referencing and the referenced attributes of the relationship are written on the sides of the box and they are linked to it by lines. Figure 2 shows an example of the representation of a relationship.

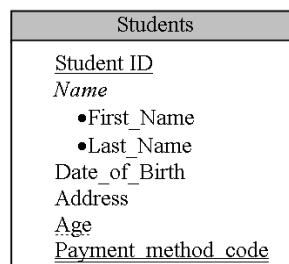


Fig. 1. Example of representation of entities and attributes

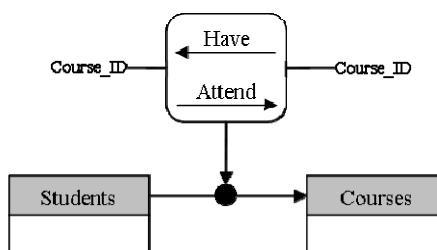


Fig. 2. Example of representation of a relationship

The cardinality of relations is represented as follows: a one to one (1:1) relationship is represented writing number one at each end of the line that represents the relationship, a one to many (1:N) relationship is represented writing number one at the end of the line that has the arrow head and a letter N at the other end, and a many to many (N:M) relationship is represented writing the letter N at one end of the line that represents the relationship and a letter M at the other end. Figure 3 shows examples of representations of relationships with these types of cardinalities.

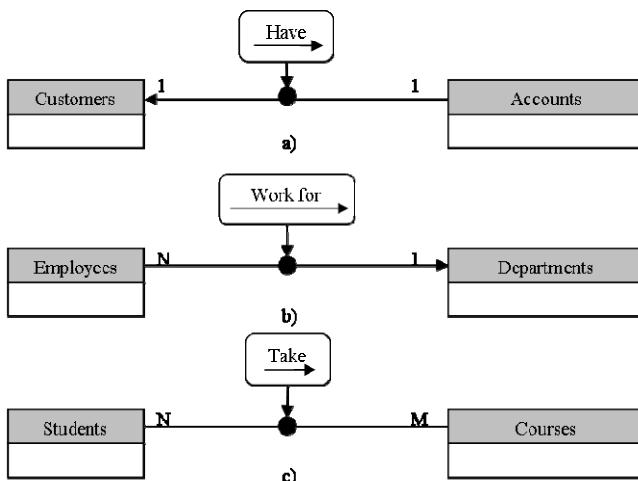


Fig. 3. Representation of cardinalities: a) one to one, b) one to many, and c) many to many



Fig. 4. Representation of descriptors: a) verbal, b) nominal, c) prepositional and d) adjectival

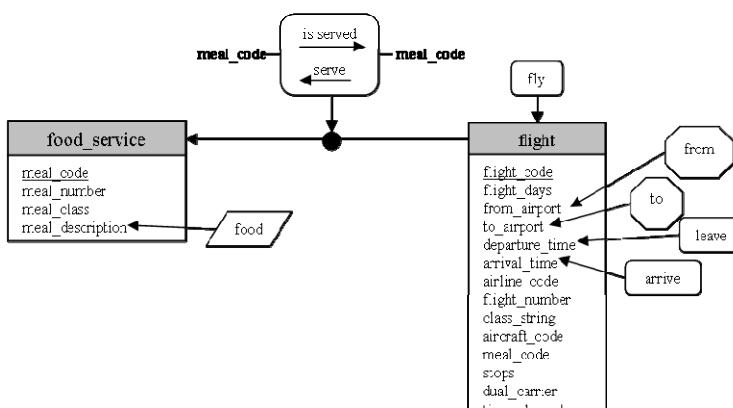


Fig. 5. Fragment of the SEM diagram for the ATIS database

Grammatical descriptors are one of the main components of the new semantic model. The representation of the four types of descriptors is shown in Figure 4.

Grammatical descriptors provide information to the NLIDB, which is useful during the semantic analysis of a query, since it permits to relate query words to the database columns and tables. Figure 5 shows a fragment of the ATIS database schema that exemplifies the use of grammatical descriptors.

6 Case Study

In order to show the usefulness of the SEM model, the following paragraphs describe its application to three of the most common problems found in the corpus for the ATIS database.

Case #1. Use of words or phrases of different syntactic categories. Though in queries, tables and columns are usually referred to by nouns; however, our study of query corpora revealed that a significant percentage of queries include verbs, prepositions and adjectives for referring to tables and columns. The following query is an example of this type of problem: *How much does it cost to fly from Boston to Oakland in one-way?* The analysis of this query shows that words and phrases such as “cost”, “to fly from”, “to”, and “one-way” refer to columns of table “Fare”. Figure 6 shows the semantic information included (modeled by suitable descriptors) for table “Fare”. These descriptors contain the information required during the semantic analysis of the query for identifying the columns and table referred to in the query.

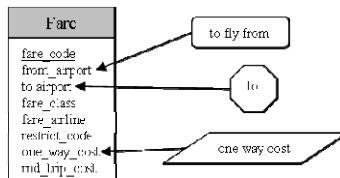


Fig. 6. Grammatical descriptors for columns of table “Fare”

Case #2. Semantic Ellipsis. In natural language communication, people usually omit words that might be crucial for the semantic interpretation of a query. The following query shows an example of this problem: *All flights and fares from ATL to SFO on 539*. This query has three types of problems: regarding “All flights” the query does not specify the information requested, “fares” might refer to two columns, and “539” is a search value for some unspecified column. For the first problem, from the model, it follows that “flights” refers to table “Flight”, from which the NLIDB can display a list of the columns of table “Flight” so the user can select the ones he/she wishes. Concerning the second problem, “fares” refers to two columns (“one_way_cost” and “rnd_trip_cost”) of table “Fare” (Fig. 6), which can be displayed to the user so he/she can choose one or both. Concerning the third problem, the model contains information about the data type of each column; with this information the NLIDB can

display a list of the columns of table “Flight” that contain numeric data, so the user can indicate the unspecified column. It is important to mention that semantic ellipsis is an extremely complex problem; and therefore, the intervention of the user is necessary for choosing among the different alternative interpretations that arise during the semantic analysis. For example, in the preceding query, there is no way that the NLI can determine if the user refers to one-way cost or round-trip cost, unless the NLI receives this information from the user through a dialogue.

Case #3. Covering of the capability of SQL. The following query shows an example that involves an aggregate function: *Find the cheapest one-way fare from Pittsburgh to Oakland first class*. In addition to the structures mentioned in Sections 4 and 5, the model includes a structure for this purpose that contains information on the aggregate functions supported by the NLIDB (Figure 7). Each of the aggregate functions has a link to each column it can be applied to, and the link has a descriptor that specifies the word or phrase for referring to such function in the wording of a query. This information is useful during the semantic analysis for identifying the aggregate function referred to by the word “cheapest” in the query.

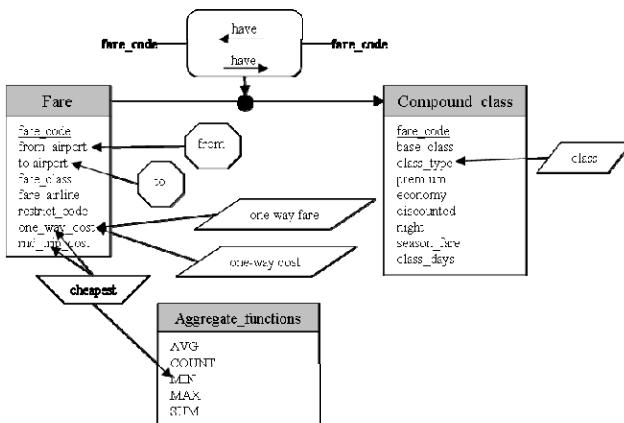


Fig. 7. Information for associating aggregate functions to columns

7 Final Remarks and Future Work

Despite the fact that since the late 60s many NLIDBs have been developed, up to now many problems continue that prevent the translation process to be totally successful, i.e., error free. Most of the NLIDBs developed have focused on solving existing complex problems, but have overlooked a key aspect of NLIDB design: a formal and comprehensive model that includes enough semantic information so as to facilitate the semantic analysis of queries for achieving success rates close to 100%.

From the experience acquired during the development of the previous version of our NLIDB, we realized about the extreme complexity of the problems involved in

translating correctly from natural language to SQL, which prevents attaining success rates above 90% by domain-independent NLIDBs.

Thus, we concluded that for solving these problems, it is necessary to apply design techniques for highly complex systems. From database management systems (DBMS), we borrowed the idea that a NLIDB should have a powerful data dictionary (similar to those found in relational DBMSs), with the difference that our data dictionary, besides the structural information of the database, should be enriched with enough semantic information in order to facilitate the translation process from natural language to SQL.

This paper presents a new modeling method for databases that includes relevant information in order to improve the performance of NLIDBs. The new model is based on the analysis of problems that occur in the corpora of three databases (ATIS, Northwind and Pubs), from which it is intended to obtain all the information needed by NLIDBs to solve such problems. We are currently designing the data dictionary for the new version of our NLIDB, based on the SEM modeling method presented in this paper.

We claim that in order to implement a successful translation process for a NLIDB, it is necessary to design an architecture for the translation process applying another design technique used in complex systems: an architecture based on functionality layers (similar to the OSI model for communications systems). We have already designed a layered architecture for the new version of our NLIDB, which will be published elsewhere.

We think that in order to attain success rates close to 95%, our NLIDB should have learning capability. This feature will permit the NLIDB to learn from its interaction with users (via an extended version of our dialogue module [6]) in order to modify information stored in its data dictionary for inserting new semantic information or correcting possible inaccuracies.

We are currently implementing the new version of the NLIDB. For evaluating the performance of the new system, it will be tested with a corpus that includes queries that involve the problems mentioned in Table 2. The results of these tests will be compared with those from the best state-of-the-art systems such as C-PHRASE [10] and STK [11], as well as ELF [3] which is one of the best existing commercial NLIDBs.

References

1. Cimiano, P., Haase, P., Heizmann, J.: Porting Natural Language Interfaces Between Domains: an Experimental User Study with the ORAKEL System. In: Proc. 12th International Conference on Intelligent User Interfaces, Honolulu, Hawaii, USA, pp. 180–190 (2007)
2. BBC News. Bill Gates Says: Mouse is Out, Touch Screen and Natural Language Interface Are In,
http://news.bbc.co.uk/player/nol/newsid_7170000/newsid_7174300/7174330.stm?bw=bb&mp=wm&asb=1&news=1&bmcws=1
3. ELF Software, <http://www.elf-software.com/FaceOff.htm>

4. DARPA Air Travel Information System (ATIS0),
[http://www.ldc.upenn.edu/Catalog/
readme_files/atis/sdtd/trn_prmp.html](http://www.ldc.upenn.edu/Catalog/readme_files/atis/sdtd/trn_prmp.html)
5. Pazos R., R.A., Pérez O., J., González B., J.J., Gelbukh, A., Sidorov, G., Rodríguez M., M.J.: A Domain Independent Natural Language Interface to Databases Capable of Processing Complex Queries. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 833–842. Springer, Heidelberg (2005)
6. Pazos R., R.A., Rojas P., J.C., Santaolaya S., R., Martínez F., J.A., Gonzalez B., J.J.: Dialogue Manager for a NLIDB for Solving the Semantic Ellipsis Problem in Query Formulation. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS (LNAI), vol. 6277, pp. 203–213. Springer, Heidelberg (2010)
7. Zhang, G., Chu, W.W., Meng, F., Kong, G.: Query Formulation from High-Level Concepts for Relational Databases. In: Proc. User Interfaces to Data Intensive Systems, pp. 64–74 (1999)
8. Stratica, N.: Using Semantic Templates for a Natural Language Interface to the CINDI Virtual Library. *Data & Knowledge Engineering* 55, 4–19 (2005)
9. Hallett, C.: Generic Querying of Relational Databases using Natural Language Generation Techniques. In: Proc. INLG 2006, Nottingham, United Kingdom, pp. 88–95 (2006)
10. Minock, M., Olofsson, P., Näslund, A.: Towards Building Robust Natural Language Interfaces to Databases. In: Kapetanios, E., Sugumaran, V., Spiliopoulou, M. (eds.) NLDB 2008. LNCS, vol. 5039, pp. 187–198. Springer, Heidelberg (2008)
11. Giordani, A., Moschitti, A.: Semantic Mapping between Natural Language Questions and SQL Queries via Syntactic Pairing. In: Horacek, H., Métais, E., Muñoz, R., Wolska, M. (eds.) NLDB 2009. LNCS, vol. 5723, pp. 207–221. Springer, Heidelberg (2010)

Modular Natural Language Processing Using Declarative Attribute Grammars

Rahmatullah Hafiz and Richard A. Frost

School of Computer Science, University of Windsor
401 Sunset Avenue Windsor, ON N9B 3P4 Canada
`{hafiz,richard}@uwindsor.ca`

Abstract. A system based on a general top-down parsing algorithm has been developed which allows language processors to be created as executable specifications of arbitrary attribute grammars. Declarative notation of attribute grammars allows modular construction of executable language definitions. Syntax is defined through general context-free grammar rules, and meaning is defined by associated semantic rules with arbitrary dependencies. An innovative technique allows parses to be pruned by arbitrary semantic constraints. This new technique is useful in modelling natural-language phenomena by imposing unification-like restrictions, and accommodating long-distance and cross-serial dependencies, which cannot be handled by context-free rules alone.

Keywords Top-down parsing, Attribute grammars, Lazy evaluation, Constraint-based Formalism, Compositional Semantics.

1 Introduction

Top-down analysis of languages showed early potential by being modular, and by accommodating a wide range of linguistic phenomenon that are essential for deep analysis. Modular top-down analysis allows construction of larger language processors by piece-wise combination of smaller components. This type of construction is especially useful when processors (e.g., natural language interfaces) compute meanings using compositional semantics such as Montagovian semantics. Specifying syntax and semantics to describe formal languages using denotational notation of attribute grammars (AGs) has been practiced extensively. However, very little work has shown the usefulness of declarative AGs for computational models for natural language. Previous work falls short in accommodating ambiguous CFGs with left-recursive rules, and providing a declarative syntax-semantics interface that can take full advantages of dependencies between syntactic constituents to model linguistically-motivated cases.

In this paper we show that a general top-down parsing approach can be used to build a system in a purely-functional language Haskell [1] where application developers can specify syntactic and semantic descriptions of natural languages using a general notation of AGs as directly executable specifications. The underlying top-down analysis method parses ambiguous sentences with general

CFGs efficiently and allows coupling semantic rules with syntax declaratively. We show that this parsing method can be used not only to produce compact representation of highly ambiguous parses efficiently and to compute meanings of sentences using semantics, but also to model linguistic properties that may require more restricted generative formalisms than context-free grammars. Some of these properties we demonstrate by only using upward propagating synthesized attributes include characteristics of unification-based formalisms (e.g., subject-verb agreements), unbounded syntactic dependencies in sentences for some form of disambiguation, and generative formalisms that can process cross-serial dependencies and duplicate languages.

Similar formalisms exist that can impose unification-like restrictions while parsing with CFGs (see definitions of definite clause grammar (DCG) [2] and Affix grammar [3] formalisms). However, our emphasis is to build modular and executable language processors by integrating declarative compositional semantics with context-free rules. Our technique relies on 1) employing a set of higher-order functions that allow the construction of directly-executable specifications of syntax and semantics rules, 2) using a context and depth-aware top-down parsing algorithm for general CFGs and a variation of explicit memorization for time and space efficiency, and 3) the lazy-evaluation procedure to impose constraints to model targeted languages. Our declarative notation allows designers to specify *what* the description of the language is, *what* relationships exist between components, and *what* restrictions are needed to be imposed, rather than *how* these procedures actually work.

2 General Notation

Operators for Syntactic Analysis: consider the following CFG (in BNF) for a segment of English that contains sentences such as **bob saw a nightingale**:

<i>sent</i>	$::=$	<i>tp vp</i>	<i>tp</i>	$::=$	<i>pnoun det np</i>
<i>vp</i>	$::=$	<i>verb tp</i>	<i>np</i>	$::=$	<i>noun</i>
<i>pnoun</i>	$::=$	<i>'bob'</i>	<i>noun</i>	$::=$	<i>'nightingale'</i>
<i>verb</i>	$::=$	<i>'saw'</i>	<i>det</i>	$::=$	<i>'a'</i>

Example CFG 1

In the above grammar, a nonterminal sentence (*sent*) is a term phrase(*tp*) followed by a verb phrase (*vp*). A *tp* can be expanded using two alternative rules - one rewrites *tp* to a proper noun *pnoun*, and in the other one *tp* is formed by a determiner (*det*) followed by a noun phrase (*np*). The alternatives are separated with a *|* and terminals are in single quote. When *sent* is applied to **bob saw a nightingale**, it will unambiguously produce a single parse tree.

In our system for executable specifications, we use a set of higher order functions or combinators to modularly denote syntactic descriptions similar to the above example. We use the combinators *<|>* and **>* to denote alternating and sequencing of syntax symbols, and *term* to represent a terminal. We will gradually increase this set of combinators or operators to accommodate other linguistic

properties. Using these basic combinators, we can represent the CFG 1 as the following executable specification:

<code>sent = memoize Sent (tp *> vp)</code>	<code>tp = memoize Tp (pnoun < > det *> np)</code>
<code>vp = memoize Vp (verb*> tp)</code>	<code>np = memoize Np noun</code>
<code>pnoun = term "bob"</code>	<code>noun = term "nightingale"</code>
<code>verb = term "saw"</code>	<code>det = term "a"</code>

In this executable specification, each expression is a function for a non-terminal that maps an index representing the current input position (*Start*) to a parse-tree of the recursive data-type *PTree* (see below) that indicates the structure of the current parse that successfully ends at the position *End*. The final result of complete parsing is a list of these *PTrees* with their starting and ending positions. The embedded (*Start*, *End*) pairs in trees work as pointers to indicate *where to go next*, which allows the trees in the result set to be one-level-depth branches, sub-nodes or leaves. The trees that start and end at identical positions are shared between all non-terminal entries (see [4] for the formal definition of the *shared forest* in parsing). Note that each non-terminal's functional definition is *memoized* (a technique similar to [5])) with a wrapper function *memoize* that ensures that a non-terminal (identified by a unique label e.g., `Sent`) is executed at an input position at most once. This reuse of previously-computed results guarantees $O(n^3)$ and $O(n^4)$ worst-case time complexities for non left-recursive and left-recursive CFGs respectively.

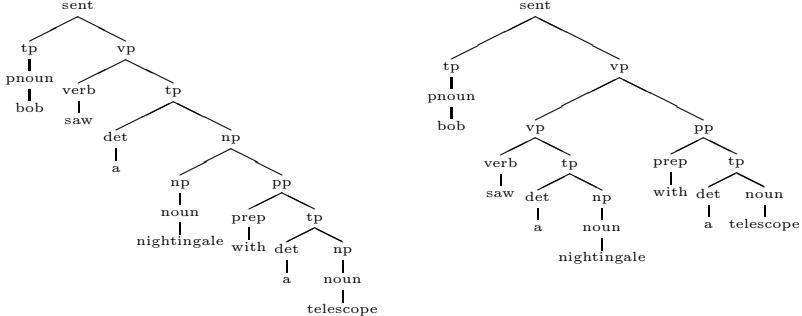
```
type Start/End = Int
data PTree Label = Leaf Label | Branch[PTree Label]
                  | SubNode(Label, (Start, End))
type Result = [(Start, End), [PTree Label]]
```

The compact and memoized result for all successful parses is systematically threaded through parser-executions. As pure functional combinators do not have *side effects*, the threading of the memoized table is done with a *state monad* [6]. We can extend CFG 1 by adding rules for prepositional phrases *pp* so that we can parse a wide range of sentences such as `bob saw a nightingale with a telescope`. We can also add more terminals according to the target language:

<code>sent ::= tp vp</code>	<code>tp ::= pnoun det np</code>	<code>pp ::= prep tp</code>
<code>vp ::= vp pp verb tp</code>	<code>np ::= np pp noun</code>	<code>pnoun ::= 'bob'</code>
<code>noun ::= 'nightingale' 'telescope'</code>		<code>prep ::= 'with' 'on' etc.</code>
<code>verb ::= 'saw' 'see' etc.</code>		<code>det ::= 'the' 'an' 'a'</code>

Example CFG 2

This extended CFG 2 now contains left-recursive rules (e.g., `vp ::= vp pp`), and it can accommodate ambiguity. For example, when we parse the sentence `bob saw a nightingale with a telescope` starting from the non-terminal `sent` using a general parsing algorithm we get the following two parses:



As our system allows piecewise integration of smaller components to form a larger description, we can seamlessly extend our previous executable specification of syntax to model CFG 2 (as shown below). This form of modular construction also allows testing and editing of individual components separately without affecting other parts. Hence our system is loosely coupled yet highly cohesive in nature.

```

sent = memoize Sent (tp *> vp)
tp = memoize TP (pnoun <|> det *> np)
pp = memoize PP (prep *> tp)
vp = memoize VP (verb *> tp <|> vp *> pp)
np = memoize NP (noun <|> np *> pp)
noun = memoize "noun" term "nightingale" <|> term "telescope"
pnoun = memoize "pnoun" term "bob"           det = memoize "det" term "a" <|> etc.
prep = memoize "prep" term "with" <|> etc.   verb = memoize "verb" term "saw" <|> etc.
  
```

This specification can be executed directly following a general top-down parsing process ([7]) to accommodate a subset of English that contains ambiguity. When the starting nonterminal **sent** is applied at the first token position of the sentence **bob saw a nightingale with a telescope**, it produces two parses that are embedded in the following pretty-printed compact format. Two different parses can be extracted if we follow the (*Start*, *End*) pointing pairs starting from the **Sent**'s entry. Notice that ambiguities arise from two of the **vp**'s entries where the verb phrase starts at position 2 and ends at position 8 for the substring **saw a nightingale with a telescope** using two different sub-tress. Each entry in this compact representation is shared i.e., they are entered exactly once in the result set. For example, the prepositional phrase **pp** that stars at position 5 and ends at position 8, is shared by the **vp** and the **np**. All ambiguous entries are grouped under common nodes that have referencing pointers to construct complete parse trees whenever needed (similar to [8] and the more general proposal due to Bernard Lang [9]).

```

"Sent" (Start: 1,End: 8) Branch [SubNode ("tp", (1,2)),SubNode ("vp", (2,8))]
"VP"   (Start: 2,End: 8)[Branch [SubNode ("verb", (2,3)),SubNode ("tp", (3,8))]
                      ,Branch [SubNode ("vp", (2,5)),SubNode ("pp", (5,8))]]
          (Start: 2,End: 5) Branch [SubNode ("verb", (2,3)),SubNode ("tp", (3,5))]
"TP"   (Start: 1,End: 2) SubNode ("pnoun", (1,2))
          (Start: 3,End: 8) Branch [SubNode ("det", (3,4)),SubNode ("np", (4,8))]
          (Start: 3,End: 5) Branch [SubNode ("det", (3,4)),SubNode ("np", (4,5))]
          (Start: 6,End: 8) Branch [SubNode ("det", (6,7)),SubNode ("np", (7,8))]
"NP"   (Start: 4,End: 8) Branch [SubNode ("np", (4,5)),SubNode ("pp", (5,8))]
"PP"   (Start: 5,End: 8) Branch [SubNode ("np", (5,6)),SubNode ("pp", (6,8))]
  
```

```
"pnoun" (Start: 1,End: 2) Leaf "bob"      "verb"  (Start: 2,End: 3) Leaf "saw"
"det"   (Start: 3,End: 4) Leaf "a"        "noun"   (Start: 4,End: 5) Leaf "nightingale"
"prep"  (Start: 5,End: 6) Leaf "with"     "det"    (Start: 6,End: 7) Leaf "a"
"noun"  (Start: 8,End: 9) Leaf "telescope"
```

Operators for Semantic Analysis: *Computing Meaning.* Our system is ideally-suited to incorporate compositional semantics that have a one-to-one correspondence between the rules defining syntax and the rules stating how the meaning of phrases are constructed from the meanings of their constituents. Richard Montague, who was one of the first to develop a compositional semantics for English, suggested to treat natural language semantics similar to formal language semantics as purely-functional lambda(λ) expressions where larger expressions are constructed by composing smaller expressions. As the original proposal is computationally intractable, we use an efficient set-theoretic version [10].

Alongside with piecewise syntactic extension, we can glue together semantics rules for corresponding syntax rules to compute meanings of a larger set of languages. We extend the set of combinators to integrate semantics with syntax as directly-executable specifications of general attribute grammars. Consider a subset of CFG 2 $sent ::= tp vp$, $vp ::= verb tp$, $tp ::= pnoun|det noun$ that can parse the sentence **bob saw a nightingale**. We define an AG where this CFG's rules are accompanied by sets of semantic rules, which are Montague-style compositional semantics as (λ) expressions:

$sent(S_0)$	$::= tp(T_0) vp(V_0)$
$\{S_0.VAL$	$\uparrow = (\lambda p\ p\ T_0.VAL \uparrow) V_0.VAL \uparrow\}$
$tp(T_0)$	$::= pnoun(P_0)$
$\{V_0.VAL$	$\uparrow = P_0.VAL \uparrow\}$
	$ det(D_0) noun(N_0)$
$\{T_0.VAL$	$\uparrow = (\lambda p\ D_0.VAL \uparrow\ p)\ N_0.VAL \uparrow\}$
$vp(V_0)$	$::= verb(V_1) tp(T_1)$
$\{V_0.VAL$	$\uparrow = (\lambda p\ V_1.VAL \uparrow\ p) T_1.VAL \uparrow\}$
$pnoun(P_0)$	$::= 'bob'$
$\{P_0.VAL$	$\uparrow = (\lambda p\ 'bob' \in p)\ \{\text{human}\}\}$
$noun(N_0)$	$::= 'nightingale'$
$\{N_0.VAL$	$\uparrow = \{\text{nightingales}\}\}$
$verb(V_0)$	$::= 'saw'$
$\{V_0.VAL$	$\uparrow = (\lambda z\ z(\lambda x\lambda y\ applytransvp(y, x)))\}$
$det(D_0)$	$::= 'a'$
$\{D_0.VAL$	$\uparrow = \lambda p\lambda q(p \wedge q) \neq \phi\}$

Attribute Grammar (AG) 1

In AG 1, non-terminals are identified with labels (e.g. V_0 , N_0 etc.), and these non-terminals have synthesized attributes (e.g. $V_0.VAL \uparrow$) that propagate upward. If inherited attributes are needed that propagate downward then we can mark them with \downarrow s. The semantics are correctly-typed pure functions, and some of them are higher-order functions. E.g.,, the semantic rule for the transitive verb **saw**, which is identified with the $V_0.VAL$, is a function that receives two functions - one defines the term phrase **a nightingale** and the other defines the proper noun **bob** as input arguments. The semantic for **saw** eventually passes arguments into a user-defined function *applytransvp* that determines the authenticity of the transitive verb's relation between the subject and the object.

We extended our set of combinators to accommodate AGs like the above, which are described in section 4. We have built combinators `rule_s` and `rule_i` to form synthesized and inherited semantic rules, wrapper functions `parser` and `nt` to form a complete AG expressions, and we refer to syntactic constructs by their unique labels (e.g., `V1`, `P1` etc.) in order to use them as semantic functions' arguments. By allowing any syntactic construct's attributes that are available in a syntax rule as a semantic functions' arguments, we can form arbitrary dependencies between these constructs that are necessary for many linguistic phenomena. The above AG can be described declaratively with our notation in Haskell's syntax to form a directly-executable specification. Below we show only a few AG rules (the rules for nonterminals `sent`, `tp`, and `vp`) to introduce our notation for integrating semantic rules with syntax rules:

```

1. sent = memoize Sent
2.   (parser (nt tp T0 *) nt vp V0)
3. [rule_s val OF LHS EQ applytp [synthesized VAL OF V0, synthesized VAL OF T0]]
4. tp = memoize TP
5.   (parser (nt det D0 *) nt noun N0)
6. [rule_s val OF LHS EQ applydet [synthesized VAL OF D0, synthesized VAL OF N0]]
7. <|> parser (nt pnoun P0)
8. [rule_s Ref OF LHS EQ copy [synthesized Ref OF P0])
9. vp = memoize VP
10. (parser (nt verb V1 *) nt tp T1)
11. [rule_s val OF LHS EQ applyvp [synthesized VAL OF V1, synthesized VAL OF T1]]

```

In the above specification, lines 2, 5, 7, and 10 are syntax rules for `sent`, `tp`, and `vp`, and the rest are sets of associated declarative semantics. E.g., the synthesized `val` attributes of non-terminals (which are identified by LHSs) are computed with user functions (e.g. `applytp`) by supplying the required syntactic constructs' attributes (such as `synthesized VAL OF V0`). These user functions perform similar tasks that are mentioned as λ expressions in the AG 1.

Using Semantics for Disambiguation: Here we provide a simple example to show an innovative use of our declarative semantic notation to perform one form of natural language disambiguation. The sentence `bob saw a nightingale with a telescope` that we parsed using CFG 2, which produced two syntax trees, can also be interpreted in more than one way. In a prepositional phrase `pp`, a preposition demonstrates an adjectival or adverbial property by quantifying either the noun phrase `np` or the verb phrase `vp`. According to the first parse, the `np`'s head noun `nightingale` is quantified by the preposition `with`, which can be misinterpreted as “`a telescope carrying nightingale`”. But in the second parse, the `with` quantifies the verb `saw` from the `vp`, which is semantically more meaningful. Based on this argument, we can make sure that the first parse is discarded while generating only the second parse by using the following attribute grammar:

<code>sent(S0)</code>	$::= tp(T_0) vp(V_0)$
<code>tp(T0)</code>	$::= det(D_0) np(N_0) pnoun(N_1)$
<code>pp(P0)</code>	$::= prep(P_1) tp(T_0) \{P_0.Ref \uparrow = T_0.Ref \uparrow\}$
<code>vp(V0)</code>	$::= vp(V_1) pp(P_0)$
$\{V_0.Ref$	$\uparrow = V_1.Ref \uparrow$
$, V_0.Kill$	$\uparrow = V_1.Ref \uparrow \in P_0.Ref \uparrow$
	$ verb(V_1) tp(T_1))$
$\{V_0.Ref$	$\uparrow = V_1.Ref \uparrow\}$
$np(N_0)$	$::= np(N_1) pp(P_0)$

$\{N_0.\text{Ref}$	$\uparrow = N_1.\text{Ref} \uparrow$
$,N_0.\text{Kill}$	$\uparrow = N_1.\text{Ref} \uparrow \in P_0.\text{Ref} \uparrow\}$
	$ \text{noun}(N_1)$
$\{N_0.\text{Ref}$	$\uparrow = N_1.\text{Ref} \uparrow\}$
$p\text{noun}(N_0)$	$::= \text{bob}$
$\text{noun}(N_0)$	$::= \text{nightingale} \quad \{N_0.\text{Ref} \uparrow = \text{bird}\}$
	$ \text{telescope} \quad \{N_0.\text{Ref} \uparrow = \text{used to see}\}$
$\text{verb}(V_0)$	$::= \text{saw} \quad \{N_0.\text{Ref} \uparrow = \text{to see}\}$
$\text{prep}(V_0)$	$::= \text{with} \quad \text{det}(D_0) ::= a$

Attribute Grammar (AG) 2

In the above AG, we added semantic rules for agreements with the verb phrase *vp* and the noun phrase *np* that result in rejection of the sub-parse that produces an attribute *Kill* of the value *true*. According to our approach, when a sub-parse is rejected, then the entire parse is rejected too. These rejection rules check to see whether prepositional phrase *pp* is meaningfully quantifying the neighboring verb phrase and noun phrase by matching their upward propagating *Ref* attribute values. In this artificial example, we wanted to enforce the requirement that the noun *telescope* has a meaningful match with the verb *saw* (i.e., the *saw*'s attribute belongs to the *telescope*'s attribute), but not with the noun *nightingale*. In subsequent sections we describe the working mechanism of our approach. The above AG can be described modularly with our notation to create the following declarative specification (we only show the rule for verb phrase *vp*):

```
vp = memoize VP (parser (nt vp V1 *> nt pp P1)
  [rule_s Ref OF LHS EQ copy [synthesized Ref OF V1]
   ,rule_s Kill of LHS EQ notElemOf [synthesized Ref OF, synthesized Ref OF P1]]
  <|> parser (nt verb V1 *> nt tp T1)
  [rule_s Ref OF LHS EQ copy [synthesized Ref OF V1],rule_s Kill of LHS EQ False ])
```

Here the second semantic rule for *vp* can discard a parse by checking whether the *Ref OF V1* is a member of *Ref OF P1* with the function *notElemOf*. When this *vp* rule is placed with other rules to form the complete executable AG, and when the root non-terminal *sent* is applied to our example sentence, then it will only generate one compactly-represented parse tree similar to the 2nd parse tree on the 2nd page.

3 Underlying Foundation

3.1 Grammars, Top-Down Parser and Combinators

In an attribute grammar (AG, [11]), syntax rules of context-free grammars (CFGs)[12] are augmented with semantic rules to describe the meaning of the sentences of a context-free language. Although different definitions are given in [13,14,15], we define a general AG by imposing minimal restrictions on attribute dependencies. A synthesized attribute is an attribute for the LHS non-terminal of a production rule, and it can be computed using any attributes from the RHS terminals/non-terminals. Whereas an inherited attribute is associated with a terminal/non-terminal that resides at the RHS of the production rule, which can be computed attributes of terminals/non-terminals that are on the right or on the

left of the current non-terminal. Inherited attributes propagate downwards and synthesized attributes propagate upwards. From the viewpoint of pure functions and lazy-evaluation, semantics rules and attributes are interchangeable. When we refer to an attribute we either refer to a value - *static semantics*, or to an unevaluated expression - *dynamic semantics*.

Parser combinators have been used extensively ([16,17] etc.) to prototype top-down backtracking recognizers and parsers to provide modular and piecewise construction of executable grammars that can accommodate ambiguity. In basic recursive-descent recognition, syntax rules are constructed as mutually-recursive functions, each of which maps an starting *input* position to a set of indices corresponding to a set of ending positions at which the parser successfully finished recognizing a sequence of *input* tokens. An empty result set indicates that the recognizer has failed. After an alternative rule has been applied, the recognizer backtracks to try another rule. The result for an ambiguous input contains one or more ending indices. In parsing, indices in the result set are replaced by tree structures to show successful parsing structures. However, recognizers constructed with basic combinators do not terminate for left-recursive grammars, and when extended to parsers they require exponential time and space for ambiguous input in the worst case. Frost et al. addressed these problems [7] using a technique that restricts the depth of left-recursive calls, curtails a parse when left-recursive call exceeds the number of remaining input tokens, and that tracks curtailed indirect left-recursive non-terminals to determine the context at which the result has been constructed.

3.2 Declarative and Executable Attribute Grammars

Forming Arbitrary Dependency: In [18], authors demonstrated how semantics rules can be integrated with CFGs' syntax in top-down parsing by describing the implementation of combinators in the purely functional language Haskell to enable the formation of modular language specifications. The new parser (from now on we shall refer non-terminals of AGs as parsers) now has a set of attributes, which are computed from other terminals/non-terminals' attributes that belong to the current parser's syntax definition(s).

Even if a parser contains syntax rules as well as semantic rules, the result of a parser's execution is still a mapping from a start position of input to a set of parse-tree structures with end points. A recursive parse-tree can be a single leaf (to represent a terminal), a sub-node (to represent a non-terminal), or a branch (to represent combinations of terminals and non-terminals, which models more complex syntax). The primary change in the tree data structure is how attributes are represented and threaded so that they are available for dependencies that are specified in the semantic rules. These attributes are essentially purely-functional lazy expressions. E.g., the `BinaryOP` attribute in the example below is a function that computes an integer by applying a binary operation to two input integers.

```
type Start/End = (Int, [Attributes])
data Attributes = MaxVal {getAVAL :: Int}
                | BinaryOP {getBOP :: (Int → Int → Int)}
```

Synthesized and inherited semantic rules associated with a parser are mapped onto the starting and ending positions respectively in the parser's result-set. This facilitates the overall syntax-directed evaluation and allows semantic rules to be denoted in terms of potentially unevaluated attributes from the environment of the current parser, its predecessor, successors, and sibling parsers. E.g., when a parser p_i with a syntax rule $p_i = p_m \star> p_n$ is applied to position 1 and successfully ends at position 5, one of p_i 's input/output attribute relations might be:

$$\begin{array}{c}
 p_i(\text{inh}_{p_i1}, \text{syn}_{p_i5}) \\
 \swarrow \quad \searrow \\
 p_m(\text{inh}_{p_m1}, \text{syn}_{p_m3}) \quad p_n(\text{inh}_{p_n3}, \text{syn}_{p_n5})
 \end{array}$$

where, assuming p_m starts parsing at 1 and ends at 3, p_n starts parsing at 3 and ends at 5, inh_{xy} and syn_{xy} represent inherited and synthesized attributes of parser x at position y respectively. From this structure, semantic functions with arbitrary attribute dependencies can acquire input arguments whenever required while they are integrated with the syntax rule.

Construction of Combinators for Executable AGs: The alternative combinator $\langle | \rangle$'s inputs p and q are alternative syntax with lists of respective semantic rules. The associated semantic rules include synthesized rules for the parent parser (i.e. *LHS*) and inherited rules for parsers that are in the associate syntax rule. p and q are executed with the current *start* position and a *context* (for left-recursive calculation), and they pass down their ids and inherited rules so that they can be used in succeeding parsers' semantics. Threading appropriate semantic rules to the appropriate syntax symbol is done with the combinator *parser*. All results from alternative parsers are merged together at the end.

Terminals or non-terminals (say p and q) can be sequenced with the combinator $\star>$ to form a compound syntax. $\star>$ enables p to be applied to the current *start* position and *context*, and to compute its inherited attributes using the combinator *nt* from an environment that consists of p 's precursor's attributes, and the *result* of the current syntax. This *result* (a tree structure) contains all sequencing parsers' synthesized and inherited attributes. The next parser q is then applied to the set of end positions returned by p , and computes inherited attributes from the same environment using the combinator *nt*. A result from p is united with all subsequent results from q to form new branches.

The higher-order function *nt* forms a non-terminal of an AG that enables parsers to pass down their own identification and a list of inherited semantic rules so that they can be used in subsequent AG definitions. These rules are evaluated by applying them on an environment that consists of the predecessor's and surrounding parsers' attributes. This function ultimately maps the associated parser's inherited attributes to the starting point of its result so that other inquiring parsers know where to look if needed in the future. The wrapper function *parser* forms a complete AG rule by mapping the current parser's synthesized rules to the ending points of the result, and by assisting each parser in the current syntax to have an access to a common environment for their own semantic rules' future needs.

Static synthesized attributes are provided within the definitions of the AG combinators *terminal* and *empty* that define the terminals in the AG rules, and these attributes are passed upward with the end positions in a tree of type *leaf* only if the terminal successfully consumes an input token. Declarative synthesized and inherited rules are constructed with the help of the function *rule* by applying a desired user operation to a list of synthesized and/or inherited expressions from surrounding parsers.

4 Modeling Unification-Based Formalisms

As with CFGs, unification-based formalisms [19] are language describing machines with added capabilities to impose linguistically motivated restrictions. These formalisms use an information-domain of *feature structures*, which are partial functions from features to their values where the values can be atomic or feature structures. A NL phrase is associated with a feature structure to guarantee its category and properties. *Unification* is a declarative process of restrictive combinations of information from two structures to form a new structure, and if the combination fails then the formalism does not recognize the current input.

The general notation of AGs that we introduced earlier can be utilized for modular and declarative modeling of unification. We can compare *feature structures* with a set of synthesized or inherited attribute equations. The unification takes place by comparing evaluated values of these attributes according to the restriction that is needed to model linguistic correctness. We introduce a special purpose synthesized attribute *Kill* of type *bool* for all syntax symbols in an AG. The one-pass syntax and semantic analysis algorithm is modified so that when the set of synthesized expressions for a non-terminal has an expression dedicated for the *Kill* attribute, and when this attribute is evaluated to *True* then the entire current result is discarded. This step is required to ensure that if a condition or restriction in the semantics fails then the current parse should not be a member of the result. When the expression for *Kill Bool* is mapped onto the current result, a predefined operation *nomatch* tries to determine whether all input argument expressions are evaluated to the same value or not:

```

nomatch(e1 : e2 : inputExps, environment)
= do let att1 ← e1(environment), att2 ← e2(environment)
     if att1/ = att2 then return Kill True
     else return nomatch(inputExps, environment)

nomatch([e1, e2], environment)
= do let att1 ← e1(environment), att2 ← e2(environment)
     if att1/ = att2 then return Kill True else return Kill False

```

These input expressions form restricted dependencies that can be used to impose linguistically motivated conditions in conjunction with compositional semantics. The following simple directly executable AG shows that phrases such as *moons that spin* is recognized as a grammatically correct sentence, but if we test the input is the phrase *moons that spins* then the parse fails because of the grammatical disagreement that sets *Kill True* for the sentence.

```

sent = memoize Sent
(parser (nt termph S1 *> nt relpro S2*> nt vbph S3)
 [rule_s Kill OF LHS EQ nomatch [synthesized VAL OF S1, synthesized VAL OF S3]
 ,rule_s SENT_VAL OF LHS EQ apply_termphrase
 [synthesized TERMPH_VAL OF S1, synthesized RELPRO_VAL OF S2, synthesized VERBPH_VAL OF S3]])
termph
= memoize Termph (terminal (term "moons") [TERMPH_VAL set_of_moons, VAL "plural"]
 <|> terminal (term "planets") [TERMPH_VAL set_of_planets, VAL "plural"])
vbph
= memoize Vbph (terminal (term "spin") [VERBPH_VAL set_of_spin, VAL "plural"]
 <|> terminal (term "spins") [VERBPH_VAL set_of_spin, VAL "singular"])
relpro
= memoize Relpro (terminal (term "that") [RELPRO_VAL intersect]
 <|> terminal (term "who") [RELPRO_VAL intersect])

```

Note that the other semantic rules are constructed based on the set-theoretic version of Montague’s compositional semantics. Our declarative notation for compositional semantics allows us to establish unbounded dependency that exists in relative clause sentences. According to Montague, relative pronouns *relpro* act as a conjunction (i.e. *and*) for two surrounding phrases *p* and *q* (i.e., *and* = $\lambda p \lambda q (\lambda z(p z \wedge q z))$). In the set-theoretic version, the *and* is defined as set intersection between two phrases, which are computed as sets too. In the above example, relative pronouns (*that*, *who*) have a synthesized attribute *intersect*, which is a function that expects two sets as input to perform the set intersection operation. This attribute is propagated upwards to the *sent*’s definition as *RELPRO_VAL*. The *sent* computes the final *SENT_VAL* through an *apply_termphrase* function that provides two sets (*TERMPH_VAL* and *VERBPH_VAL*) to *RELPRO_VAL* from two surrounding phrases - term phrase and verb phrase.

In [20] Correa systematically compared properties of attribute grammars and unification grammars, and demonstrated that attribute grammars are more general than unification grammars in terms of expressive power or computational efficiency. He also mentioned difficulties in implementing generalize attribute grammars because of problems related to attribute evaluations. As our parsing strategy is non-strict, our attribute evaluation order is demand driven and untangled by nature. Moreover, unlike unification-based notations, we can refer to unevaluated expressions as attributes in semantic dependencies, which enriches the overall declarativeness and modularity of the language description.

5 Accommodating Phenomena Beyond Context-Free

It has been well-documented that a few instances of natural language (e.g., some sentences from Dutch or Swiss German) may not be accommodated by context-free grammars; rather they need a stronger formalisms that are often referred to as mildly context-sensitive grammars (MCSGs) [21]. MCSGs are strictly stronger than CFGs and strictly weaker than context sensitive grammars in terms of generative power. Many formal formalisms strictly or weakly satisfy characteristics of MCSG such as tree adjoining grammar (TAG), combinatory categorial grammar (CCG), linear indexed grammar (LIG) etc. (see [22] for the proof of their equivalence). Common characteristics that define MCSG formalisms include 1)

that they contain all context-free languages, 2) that the membership problem is solvable in polynomial time, 3) that a member language grows constantly, and 4) that these formalisms contain the following non context-free languages [23] -

- L1. *multiple agreement* : $\{a^n b^n c^n | n \geq 0\}$
- L2. *crossed agreement* : $\{a^n b^m c^n d^m | n, m \geq 0\}$
- L3. *duplication* : $\{ww | w \in \{a, b\}^*\}$

We now show how to take advantage of arbitrary attribute dependencies to model above languages. As AGs generate languages that are founded on context free grammars, it is not directly possible to generate the above non context free languages, but it is possible to non-strictly filter out sentences that do not belong to the target language by imposing semantic restrictions that guarantee only acceptance of members of *L1*, *L2*, and *L3*. A CFG $S ::= As^* > Bs^* > Cs$, $As ::= As^* > a|a$, $Bs ::= Bs^* > b|b$, $Cs ::= Cs^* > c|c$ that generates a set of sentences (in the form of any number of *as* followed by any number of *bs* then any number of *cs*) can also potentially include sentences generated by *L1*. But, if we can enforce that the *As*, *Bs*, and *Cs* can be expanded or recursively re-write themselves exactly the same number of times (i.e., *n*) then this CFG is restricted to a grammar only for *L1*. This restriction is shown with the following example AG by checking *As*, *Bs*, and *Cs*'s *count* attribute value. Each time the *As*, *Bs*, or *Cs* accepts a token (e.g., *a*, *b*, or *c* respectively), their synthesized *count* attribute's values increase, and at the root non-terminal *S* these *counts* are compared with *nomatch*. If any mismatch is found then the *S*'s *Kill* attribute is set to *True*, and the current parse is entirely discarded.

```
s = memoize Ss (parser (nt as S0 *> nt bs S1 *> nt cs S2 )
  [rule_s Kill OF LHS EQ nomatch
   [synthesized Count OF S0, synthesized Count OF S1,synthesized Count OF S2]])
as = memoize As (parser (nt as S0 *> nt term_a S1 )
  [rule_s Count OF LHS EQ increment [synthesized Count OF S0]]
  <|> parser (nt term_a S1)
  [rule_s Count OF LHS EQ copy [synthesized Count OF S1]])
bs = memoize Bs (parser (nt bs S0 *> nt term_b S1 )
  [rule_s Count OF LHS EQ increment [synthesized Count OF S0]]
  <|> parser (nt term_b S1)
  [rule_s Count OF LHS EQ copy [synthesized Count OF S1]])
cs = memoize Cs (parser (nt cs S1 *> nt term_c S1 )
  [rule_s Count OF LHS EQ increment [synthesized Count OF S1]]
  <|> parser (nt term_c S1)
  [rule_s Count OF LHS EQ copy [synthesized Count OF S1]])
term_a = memoize TA (terminal (term "a") [Count 1])
term_b = memoize TB (terminal (term "b") [Count 1])
term_c = memoize TC (terminal (term "c") [Count 1])
```

Note that in the above example *increment* and *copy* are user-defined functions that increment and make a copy of an attribute value whenever demanded. Using a similar concept, an AG for *L2* can be constructed declaratively by using another user defined function *crossmatch* to impose cross serial dependencies. The *crossmatch*'s four argument expressions' *count* values are now used to decide whether the root non-terminal's *Kill* value is *True* or *False*:

```
crossmatch([e1,e2,e3,e4],environment)
= do let count1 ← e1(environment), count2 ← e2(environment)
     ,count3 ← e3(environment), count4 ← e4(environment)
if count1 == count3 and count2 == count4
then return Kill False else return Kill True
```

The application of *crossmatch* in AG the for the *L2* can be shown with the following partial executable specification:

```
s = memoize Ss (parser (nt as S0 **> nt bs S1 **> nt cs S2 **> nt ds S3 )
  [rule_s Kill OF LHS EQ  crossmatch [synthesized Count OF S0, synthesized Count OF S1
  , synthesized Count OF S2, synthesized Count OF S3]])..
```

To model the duplicate or 2-place copy language *L3* with our notation we can use a list-like data structure as an attribute type. The backbone CFG that potentially can contain members of *L3* along with other unwanted sentences can be represented as $S :: W:\ast>W$, $W ::= a\ast>W \mid b\ast>W \mid \epsilon$. This CFG satisfies generating $\{a, b\}^*$, but one W following another does not guarantee generating *L3*. If W had an attribute as a list of indicators representing all accepted *as* and *bs*, and if we can set the root *S*'s *Kill* value to *True* whenever two lists of indicators for two consecutive W s are not identical then unwanted sentences will be discarded. We can accomplish this objective with the partially-complete AG below. Assume that new user function *nomatchList* checks equality of two operand lists, and *addToList* attaches the first argument to the second, which is a list originated as an empty list from the alternative *empty*.

```
s = memoize Ss (parser (nt w S0  **> nt w S1)
  [rule_s Kill OF LHS EQ  nomatchList [synthesized List OF S0, synthesized List OF S1]])
w = memoize W (parser (nt term_a S0 **> nt W S1 )
  [rule_s Count OF LHS EQ addToList [synthesized Val OF S0, synthesized List OF S1]]
<|> parser (nt term_b S0 **> nt W S1 )
  [rule_s Count OF LHS EQ addToList [synthesized Val OF S0, synthesized List OF S1]])
<|> terminal (empty) [List []]....
```

The above example specifications of executable AGs for languages *L1*, *L2*, and *L3* could have been constructed differently by utilizing different variations of semantics constrains by introducing new inherited and/or synthesized attributes.

6 Concluding Comments

We have demonstrated how the declarative notation for general AGs can be used to create executable specifications to modularly prototype many NLP tasks. Our approach is founded on a top-down parsing technique that accommodates any form of CFGs coupled with declarative semantics. The overall evaluation technique is non-strict and treats attributes as unevaluated functions, and as a result arbitrary dependencies between syntactic components can be described in the semantics. We have shown that restrictions can be imposed using special-purpose attributes and customizable semantic functions to model natural language properties such as unification, and phenomena that cannot be accommodated with only CFGs. In the future we plan to investigate further incorporating dependencies involving inherited attributes along with synthesized attributes to construct general-purpose natural language interfaces where compositional semantics would compute meanings. We are intending to investigate many other forms of natural language disambiguation using our declarative restrictive capability. We also believe that our transparent syntax-semantic interface can be used effectively for syntax-based machine translation applications.

References

1. Hudak, P., Peterson, J., Fasel, J.: A gentle introduction to haskell 98. Technical report (1999)
2. Pereira, F.C.N., Warren, D.: Definite clause grammars for language analysis. Artificial Intelligence 13 (1980)
3. Koster, C.H.A.: Affix grammars for natural languages. In: Attribute Grammars, Applications, and Systems, pp. 469–484 (1991)
4. Billot, S., Lang, B.: The structure of shared forests in ambiguous parsing. In: 27th ACL, BC, Canada, pp. 143–151 (1989)
5. Norvig, P.: Techniques for automatic memoization with applications to context-free parsing. Computational Linguistics, 91–98 (1991)
6. Wadler, P.: Monads for functional programming. In: Jeuring, J., Meijer, E. (eds.) AFP 1995. LNCS, vol. 925, Springer, Heidelberg (1995)
7. Frost, R., Hafiz, R., Callaghan, P.: Parser combinators for ambiguous left-recursive grammars. In: ACM-PADL, pp. 167–181 (2008)
8. Tomita, M.: Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems. Kluwer Academic Publishers (1986)
9. Lang, B.: Deterministic techniques for efficient non-deterministic parsers. In: 2nd Colloquium on Automata, Languages and Programming, pp. 255–269 (1974)
10. Frost, R.A., Fortier, R.J.: An Efficient Denotational Semantics for Natural Language Database Queries. In: Kedad, Z., Lammari, N., Métais, E., Meziane, F., Rezgui, Y. (eds.) NLDB 2007. LNCS, vol. 4592, pp. 12–24. Springer, Heidelberg (2007)
11. Knuth, D.: Semantics of context-free languages. In: Theory of Computing Systems, vol. 2, pp. 127–145. Springer, New York (1968)
12. Aho, A.V., Ullman, J.D.: The Theory of Parsing, Translation, and Compiling. Parsing. Prentice Hall (1972)
13. De Moor, O., Backhouse, K., Swierstra, D.: First-class attribute grammars. In: 3rd Workshop on AGs and their Applications (2000)
14. Tienari, M.: On the definition of attribute grammar. In: Jones, N.D. (ed.) Semantics-Directed Compiler Generation. LNCS, vol. 94, Springer, Heidelberg (1980)
15. Vogt, H.H., Swierstra, S.D., Kuiper, M.F.: Higher order attribute grammars. In: PLDI, pp. 131–145. ACM (1989)
16. Wadler, P.: How to replace failure by a list of successes. In: FP Languages and Computer Architecture, vol. 201, pp. 113–128 (1985)
17. Frost, R., Launchbury, J.: Constructing natural language interpreters in a lazy functional language. The Computer Journal 32(2) (1989)
18. Hafiz, R., Frost, R.A.: Lazy combinators for executable specifications of general attribute grammars. In: ACM-PADL, pp. 167–182 (2010)
19. Shieber, S.: An introduction to unification-based theories of grammar. CSLI Lecture Notes Series. University of Chicago Press (1986)
20. Correa, N.: Attribute and unification grammar: A review and analysis of formalisms. Annals of Mathematics and AI 8, 73–105 (1993)
21. Joshi, A.K.: Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? NL Parsing, Cambridge Uni. (1985)
22. Vijay-Shanker, K., Weir, D.J.: The equivalence of four extensions of context-free grammars. Theory of Computing Systems 27(6), 511–546 (1994)
23. Kudlek, M., Mart'in-Vide, C., Mateescu, A., MitranaSource, V.: Contexts and the concept of mild context-sensitivity. Linguistics and Philosophy 26(6), 703–725 (2003)

EM Clustering Algorithm for Automatic Text Summarization

Yulia Ledeneva¹, René García Hernández¹, Romyna Montiel Soto²,
Rafael Cruz Reyes², and Alexander Gelbukh³

¹ Universidad Autónoma del Estado de México

Unidad Académica Profesional Tianguistenco,

Paraje el Tejocote San Pedro Tlantizapan, 52600, Estado de México

yledeneva@yahoo.com, renearnulfo@hotmail.com

² Laboratorio de Reconocimiento de Patrones, Instituto Tecnológico de Toluca

Av. Tecnológico s/n, Ex Rancho La Virgen, Metepec, 52140, México

romyna.montiel@yahoo.com.mx, tectoluca@hotmail.com

³ Centro de Investigación en Computación, Instituto Politécnico Nacional

Av. Juan de Dios Bátiz s/n, D.F., 07738, México

www.Gelbukh.com

Abstract. Automatic text summarization has emerged as a technique for accessing only to useful information. In order to known the quality of the automatic summaries produced by a system, in DUC 2002 (Document Understanding Conference) has developed a standard human summaries called gold collection of 567 documents of single news. In this conference only five systems could outperforms the baseline heuristic in single extractive summarization task. So far, some approaches have got good results combining different strategies with language-dependent knowledge. In this paper, we present a competitive method based on an EM clustering algorithm for improving the quality of the automatic summaries using practically non language-dependent knowledge. Also, a comparison of this method with three text models is presented.

Keywords: Automatic text summarization, extractive summarization, EM clustering algorithm, text models, n-grams, maximal frequent sequences.

1 Introduction

In the last two decades, we have experienced an exponential increase in the electronic text information available for being query. The best example is the immense and ever-growing collection of documents consulted most often: the Internet, with millions of digital documents, in websites. Fig. 1 shows a graph illustrating the increase of active sites from February 2003 to June 2011. The data appears in Netcraft's website [3]. To access so much information nowadays is used, among others, the Google search engine to make a query. The result is a list of links to thousands of web pages, of which the user must decide what is of interest, reading only a few words from the

documents associated with links that occur as a result of the query. Often this is not enough and you need to download and read each document to find the required information. This represents a waste of time.

According to Lee [4], with Internet use, the amount of information in public domains continues to grow, causing that much of this information is redundant. Therefore, we need new technologies to efficiently process information. The generation of document summaries is a key technology to overcome this obstacle. Given this, it is essential to develop automated methods that extract the most relevant information from a text, researched by Automatic Generation of Summaries (AGS) area [6, 8, 9, 12].

Automatic Text Summarization (ATS) is an active research area that deals with single- and multi-document summarization tasks. In single-document summarization, the summary of only one document is built, while in multi-document summarization the summary of a whole collection of documents (such as all today's news or all search results for a query) is built. While we believe that our ideas apply to either case, in this work we have experimented only with single-document summaries.

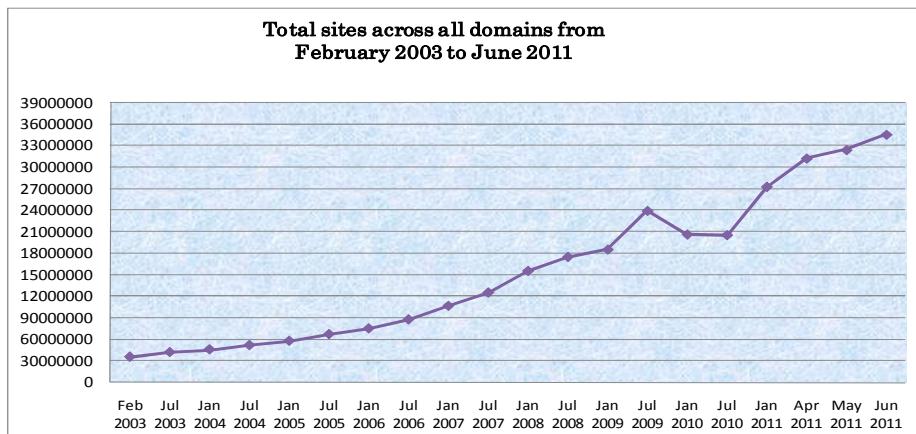


Fig. 1. Active websites from February 2003 to June 2011 [3]

Summarization methods can be classified into abstractive and extractive summarization. An abstractive summary is an arbitrary text that describes the contexts of the source document. Abstractive summarization process consists of “understanding” the original text and “re-telling” it in fewer words. Namely, an abstractive summarization method uses linguistic methods to examine and interpret the text and then to find new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original document. While this may seem the best way to construct a summary (and this is how human beings do it), in real-life setting immaturity of the corresponding linguistic technology for text analysis and generation currently renders such methods practically infeasible.

An extractive summary, in contrast, is composed with a selection of sentences (or phrases, paragraphs, etc.) from the original text, usually presented to the user in the same order—i.e., a copy of the source text with most sentences omitted. An extractive summarization method only decides, for each sentence, whether or not it will be included in the summary. The resulting summary reads rather awkward; however, simplicity of the underlying statistical techniques makes extractive summarization an attractive, robust, language-independent alternative to more “intelligent” abstractive methods. In this paper, we consider extractive summarization.

A typical extractive summarization method consists in preprocessing, term selection, term weighting, sentence weighting and sentence selection steps; at each of them different options can be chosen. We will assume that the units of selection are sentences (these could be, say, phrases or paragraphs). Thus, final goal of the extractive summarization process is sentence selection.

Usually, in the processing step the document is analyzed for removing words without meaning (stop-words) and for getting a canonical representation of each word by applying a stemming algorithm [7] in order to find relations between significant words. Moreover, some methods use more complex resources such as part-of-speech tagging, lemmatization (instead of stemming) [32, 33], weighted morphological dictionaries [34], key words, key phrases, etc.

Most of the language-independent methods employing the n -gram as the unit in term selection step which is composed by all the sequences of n words of the document, it is worth to say, than if $n = 1$ is also called bag of words model. One of the problems of the n -gram model is to determine the size of n since the dimensionality grows according to n is increased. A Maximal Frequent Sequence (MFS) model tries to select only the important terms according to the frequency without the need of determine n . A MFS text model can be defined in terms of grams as all the frequent grams (of any size) that are not subsequence of other frequent grams. For considering that a gram is frequent, it must be repeated at least a threshold times in the text, when the threshold it not specified it is assumed that is taken the lowest possible, i.e. two.

In the third step, an importance to the selected terms is given, for example, the presence or absence of a term can be used as Boolean weighting, but in this weighting it is not possible to know which term is more important. Also, the frequency of a term can be used as TF weighting, but a too frequent term is not always important since could be a stop word or a term that it is repeated in most of the sentences; therefore, it is important for the entire document and not for a single sentence. This problem can be solving if the inverse document frequency is used as IDF weighting, in this case the frequency of a term is divided by the number of sentences where the term is presented; it means a frequent term is more important if it appears in a single sentences instead of all the sentences. IDF weighting has the problem that a not frequent term is raise as important term since it is presented in few sentences.

The main problem for generating an extractive automatic text summary is to detect the most relevant information in the source document. Although, some approaches claim being domain and language independent, they use some degree of language knowledge like lexical information [2], key-phrases [3] or a golden sample for

supervised learning approaches [4-6]. Furthermore, training on a specific domain tends to customize the extraction process to that domain, so the resulting classifier is not necessarily portable. In our opinion, these works present a high dominion and language dependence degree.

The paper is organized as follows. Section 2 summarizes the state-of-the-art of text summarization methods. Section 3 describes the general scheme of the proposed approach. Section 4 presents the experimental settings followed for the experimentation. Section 5 compares our approach with those of existing methods. Section 6 concludes the paper.

2 Related Work

In 1958, Luhn presents the first work [8] in automatic summarization with an approach where the term selection is based on words and its weighting on its frequency. The sum of the terms of a sentence let to select the most important sentences. After that, other techniques have used the position of the text [10], Bayesian models as presented by Kupiec [12] and other approaches based on linguistic and statistical techniques as presented in [13-15].

TextRank [30] is a ranking algorithm based on graphs. A graph is built to represent the text, so that the nodes are words (or other text entities) interconnected by vertices with meaningful relationships. For the task of extracting sentences, the goal is to qualify whole sentences and sort highest to lowest rating. Therefore, a vertex is added to the graph for each sentence in the text. To establish connections (cycles) between sentences, define a relationship of similarity, where the relationship between two sentences can be seen as a process of "recommendation": a sentence that points to some concept in the text gives the reader a "recommendation" to refer to other sentences in the text that point to the same concepts and therefore a link can be established between any two sentences that share a common content.

Ledeneva [31] presents a method using MFS to generate extractive summaries from a single document based on statistics, which is independent of the domain and language. Ledeneva et al. [6, 31] experimentally shows that the words which are parts of bigrams (2-word sequences) which are repeated more than once in the text are good terms to describe the content of that text. This work also shows that the frequency of the term as ranking of terms gives good results (while only count the occurrences of a term in repeated bigrams). Ledeneva et al. applies a method which has 4 stages for generating the summary. These steps are term selection, term weighting, sentence weighting and sentence selection. In term selection step, SFMs, repetitive bigrams (must appear at least twice in the text), and unigrams (simply words) are extracted.

Recently, the generation of summaries from a single document or multiple documents has focused in to develop methods that use less linguistic resources. These methods can be classified as supervised and unsupervised learning methods. Usually, supervised learning methods for AGS task employ document-summary pairs for learning the sentences that are extracted from the original document [16-17]. In

unsupervised learning methods, the sentences that were clustering are extracted according to some criteria to conform the summary [1, 6, 18-23].

In [6], it is presented a 3 steps approach: term selection, term weighting and sentence selection; using an unsupervised learning algorithm. In this work, each document is preprocessed for removing stop-words and applying a stemming algorithm [8]. In the first step, the n -gram model was used. In the second step, the Boolean, TF, IDF and TF*IDF weighting were used. In the third step, the k-means algorithm is used to form groups of similar sentences. Then, for composing the summary, the most representative sentence is selected from each cluster. According to [6] resulting summaries obtained high levels of quality in comparison to other approaches. That approach, in contrast to supervised methods, does not require large sets of data as training sample. As in the work in [6], the work in [1] extends the comparative study of three text model representations: bag of words (1-grams), n -grams and Maximal Frequent Sequences (MFS's).

In [1] the same preprocessing is applied to the document. The results showed that 2-gram model (bi-grams) generated the better summaries than bag of words or SFM's models.

A great number of systems employ the ROUGE evaluation system since the results are reliable in AGS area, which let to access improvements to extend the usefulness of approaches [27].

In this paper we employ the same three text models like in [6], but instead of k-means we use the EM (*Expectation-Maximization*) algorithm in order to see if it is possible to outperform the quality of summaries.

3 General Scheme of the Proposed Approach

Our proposed method consist in term selection, term weighting and sentence selection steps. However, before these steps, the document is preprocessed for removing stop words and applying the stemming algorithm. In the first step, one of the three text model is extracted: bag of words, n -grams and MFS's. In the second step, the terms are weighting using Boolean, TF, IDF or TF*IDF scheme. In the third step, the EM algorithm is used to form groups of similar sentences in order to obtain a representative sentence from each cluster to conform the summary.

3.1 Term Selection

An n -gram is a sequence of n words. We say that an n -gram occurs in a text if these words appear in the text in the same order immediately one after another. If $n = 1$ the model is called bag of words, therefore one of the problem of this model is that the word lost its context and its actual meaning. 2-gram model tries to conserve more the context of the words but as n increases the dimensionality of the model grows.

We call an n -gram frequent (more accurately, β -frequent) if it occurs more than β times in the text, where β is a predefined threshold. Frequent n -grams—we will also call them frequent sequences (FSs)—often bear important semantic meaning:

they can be multiword expressions (named entities: *The United States of America*, idioms: kick the basket) or otherwise refer to some idea important for the text (the President's speech, to protest against the war). A Maximal Frequent Sequence is those that it is not subsequence of other FS.

3.2 Term Weighting

Boolean Weighting (BOOL): It is the easiest way to weight a term. It models the presence or absence of a term in the document, defined as:

$$w_i(t_j) = \begin{cases} 1, & \text{if the term } t_j \text{ appears in document } i \\ 0, & \text{other case} \end{cases}$$

Term Frequency (TF) was proposed in [25]. This weighting takes into account that a term that occurs in a document can better reflect the contents of document than a term that occurs less frequent. Therefore, the weighting TF assigns a greater relevance to terms with greater frequency and consists in evaluating the number of times the term appears in the document.

$$w_i(t_j) = f_{ij}, \text{ where } f_{ij} \text{ is the frequency of the term } j \text{ in document } i.$$

Inverse Document Frequency (IDF) was proposed by Salton [9] for improving information retrieval systems (IR). The problem of TF weighting in IR is that, when a term appears in almost all the documents in the collection; this term is useless for discriminating relevant documents. For example, the stop-word *and* could have a high TF, but it is useless for discriminating the relevant documents since tends to appear in most of the documents. IDF is defined as:

$$w_i(t_j) = \log\left(\frac{N}{n_j}\right), \text{ where } N \text{ is the number of documents in the collection and } n_j \text{ is the number of documents where the term } j \text{ appears.}$$

TF-IDF. The problem of IDF weighting in IR is that it is not possible distinguish between two documents with the same vocabulary (list of different words), even thought if the term is more frequent in a document. TF-IDF weighting gives more relevance to the terms that are less frequent in the collection but more frequent into the document.

$$w_i(t_j) = f_{ij} \times \log\left(\frac{N}{n_j}\right)$$

Note that in this paper we propose to use these term weights for single document summarization. Therefore, for applying these term weights we can consider the document as a collection of sentences instead of a collection of documents.

3.3 Selection of Sentences Using EM Algorithm

An unsupervised learning algorithm form groups of objects in order to achieve, in the one hand, the greatest possible similarity between objects of a group, in other hand, the greatest possible dissimilarity between objects of different groups.

In this step, we propose to use the unsupervised algorithm EM for discovering the groups of sentences with similar meaning. Then, we can select the most representative sentence from each group in order to compose the summary. The EM algorithm assumes that the number of clusters is previously known. Sometimes this characteristic is a disadvantage, however in our proposed approach is an advantage since it allows controlling the number of groups to create which lets, at the same time, estimate the number of words in the final summary. For example, if the average of words per sentence is 20 and a user desires a 100-word summary then EM must create 5 clusters, obviously this is only an estimation of the number of words in the final summary. Additionally, some tests were performed with the EM algorithm without specify the initial number of groups (-1) in order to EM will find the ideal number of clusters.

EM represents each sentence of the document in a vector space model. So, each document is represented as a vector of features, where the features correspond to the different terms in the document, in this case: bag of words, n-grams or MFS's.

EM algorithm belongs to a family of models known as finite mixture models, which can be used to segment data sets. A cluster is a probabilistic method. EM tries of getting the unknown Probability Density Function (PDF) which the set of data belong. This PDF can be approximated by a linear combination of NC components, defined by a lack of parameters

$$\{\theta\} = \{\theta_j \forall j = 1..NC\}, \quad P(x) = \sum_{j=1}^{NC} \pi_j p(x; \theta_j), \quad \sum_{j=1}^{NC} \pi_j = 1$$

Where π_j are the prior probabilities of each cluster that must sum 1, which also form part of the desired solution, $P(x)$ denotes the arbitrary FDP, NC is the number of clusters and $p(x; \theta_j)$ is the density function of the component j . Each cluster corresponds to the respective data samples belonging to each of the densities that are mixed. FDP can be estimated arbitrarily, using n-dimensional normal FDP, t-Student, Bernoulli, Poisson and log-normal.

4 Experimental Results

We used the standard DUC 2002 collection provided [26]. In particular, we used the data set of 567 news articles of different length and with different topics. Each document in the DUC collection is supplied with a set of human-generated summaries provided by two different experts.¹ While each expert was asked to generate summaries of different length, we used only the 100-word variants.

¹ While the experts were supposed to provide extractive summaries, we observed that the summaries provided in the collection were not strictly extractive: the experts considerably changed the sentences as compared with the original text.

Evaluation procedure. We used the ROUGE evaluation toolkit [24] which was found to highly correlate with human judgments [29]. It compares the summaries generated by the program with the human-generated (gold standard) summaries. For comparison, it uses n -gram statistics. Our evaluation was done using n -gram (1, 1) setting of ROUGE, which was found to have the highest correlation with human judgments, namely, at a confidence level of 95%. ROUGE lets to know the f-measure that is a balance (not an average) of recall and precision results.

Table 1. Results for 1-grams, 2-grams and MFS model with the BOOL,TF, IDF and TF-IDF weightings using the EM clustering algorithm with 5 initial seeds and without initial seeds

Initial seeds	Tex Model	Weighting	Recall	Precision	F-measure
5	1-grams	BOOL	0.46554	0.48067	0.47139
		TF	0.46916	0.46884	0.46870
		IDF	0.47304	0.47180	0.47213
		TF-IDF	0.46975	0.46866	0.46895
	2-grams	BOOL	0.46479	0.48123	0.47104
		TF	0.46609	0.46409	0.46487
		IDF	0.46975	0.46866	0.46895
		TF-IDF	0.46615	0.46412	0.46491
	MFS	BOOL	0.47191	0.48061	0.47495
		TF	0.47460	0.47985	0.47650
		IDF	0.47354	0.48012	0.47602
		TF-IDF	0.47545	0.48075	0.47742
-1+first	1-grams	BOOL	0.43517	0.48857	0.45357
		TF	0.46303	0.48256	0.47056
		IDF	0.46252	0.48253	0.47029
		TF-IDF	0.4628	0.48242	0.47038
	2-grams	BOOL	0.43615	0.48794	0.45325
		TF	0.46278	0.4826	0.47043
		IDF	0.46278	0.4826	0.47043
		TF-IDF	0.46294	0.48269	0.47056
	MFS	BOOL	0.44997	0.48434	0.46264
		TF	0.46058	0.47875	0.4676
		IDF	0.44997	0.48434	0.46264
		TF-IDF	0.45709	0.47478	0.46391

Table 1 shows the results obtained with ROUGE for different models and different term weights. We test for 1-grams, 2-grams and MFS's, and with EM algorithm with 5 seeds (100-word summary can be composed choosing 5 sentences) and without define the numbers of clusters (-1, as is denoted in WEKA software, if the summary does not complete 100 words it is completed with the *first* sentences of the text).

In table 1, it is possible to see that in general the F-measure results were similar between 0.45325 and 0.47742. Nevertheless, when the initial seed is 5 the best results were obtained by the MFS model followed by 1-gram and then by 2-grams model. In this case, contrary to the supposition, 2-grams get lower results than 1-grams although are similar. If there is not specified the number of initial seeds, the best results were obtained by 2-grams model followed much closed by 1-grams and then by MFS model. However, the best result was obtained by MFS model with 5 initial seeds.

5 Comparison

We compared our proposed approach with the following results:

- Baseline (random) [6]: This heuristic for which of the summaries are built from a set of sentences selected at random. The idea is to determine how significant the results can be achieved by other methods.
- TextRank [30] (see description above).
- Maximal Frequent Sequences (SFMs) [6, 31].
- Baseline (*first*): Selects the first sentences of the document until the desired size of the summary is reached [26]. This Baseline gives very good results on the kind of documents (news reports) that we experimented with, but would not give so good results on other types of texts.
- 2-grams, K-means: Montiel [1], we compare our methods with the best results.

The comparison of the best F-measure results of our proposed approach with the above state-of-the-art approaches is presented in Table 2. In this table, it is possible to observe that the difference of results, between the worst and best approach, is 0.08925. It difference was calculate in order to show that a centesimal or millesimal increase in F-measure is significant.

Table 2. Results of F-Measure with other methods

Method	F-measure
Baseline: <i>random</i> [6]	0.38817
TextRank: [30]	0.44320
SFM's (<i>k-best</i>)	0.45293
Baseline: <i>first</i>	0.47450
2-grams, K-means [1]	0.47570
SFM's (1best+first)	0.47634
Proposed (SFM's-EM-5)	0.47742

6 Conclusions

In this work, we proposed an extractive automatic text summarization approach by sentence extraction using algorithms contained in the machine learning package of WEKA. In particular, we use the EM algorithm. The results showed that the summaries generated by our proposed approach using the three models outperform those generated by the k-means algorithm with the same models as is shown in [1].

Applying the EM algorithm with 5 initial seeds, F-measure obtained the best results when the SFM model is used with BOOL weighting. Based on experimental results, we demonstrate that the proposed approach obtains favorably results than others state-of-the-art approaches.

Acknowledgments. Work done under partial support of Mexican Government (CONACyT, SNI, PROMEP, UAEM, PIFI). The authors thank at Autonomous University of the State of Mexico, Technological Institute of Toluca, and National Polytechnic Institute for their assistance.

References

1. Montiel, R., García, R.A., Ledeneva, Y., Cruz, R.: Comparación de Tres Modelos de Texto para la Generación Automática de Resúmenes. *Natural Language Processing Journal of Spain Society* 43, 303–311 (2009) ISSN 1135-5948
2. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Elsevier (2005) ISBN: 0-12-088407-0
3. NetCraft, June 2011 Web Server Survey, England (2011),
http://news.netcraft.com/archives/-web_server_survey.html
4. Lee, J.-H., Park, S., Ahn, C.-M., Kim, D.: Automatic Generic Document Summarization Based on Non-negative Matrix Factorization. In: *Information Processing and Management*, vol. 45, pp. 20–34. Elsevier (2009) ISSN 0306-4573
5. García, R., Ledeneva, Y., Gelbukh, A.: Keeping Maximal Frequent Sequences Facilitates Extractive Summarization. *Research in Computing Science* 28 (2008) ISSN 1870-4069
6. García-Hernández, R.A., Montiel, R., Ledeneva, Y., Rendón, E., Gelbukh, A., Cruz, R.: Text Summarization by Sentence Extraction Using Unsupervised Learning. In: Gelbukh, A., Morales, E.F. (eds.) *MICAI 2008. LNCS (LNAI)*, vol. 5317, pp. 133–143. Springer, Heidelberg (2008)
7. Porter, M.: The Porter Stemming Algorithm. Official home page for distribution of the Porter Stemming Algorithm (2006),
<http://tartarus.org/~martin/PorterStemmer/index.html>
8. Luhn, H.P.: The automatic creation of Literature abstracts. *IBM Journal of Research and Development* (1958)
9. Edmondson, H.P.: New Methods in Automatic Extraction. *Journal of the Association for Computing Machinery* (1969)
10. Brandow, R., Mitze, K., Rau, L.: Automatic condensation of Electronic publication by sentence selection. In: *Information Proc. and Management* (1995)
11. Kupiec, J., Pedersen, J.O., Chen, F.: A trainable document summarizer. In: *SIGIR 1995*, New York (1995)
12. Goldstein, J., Carbolell, J., Kantrowitz, M., Mittal, V.: Summarizing text documents: sentence and evaluation metrics. In: *22nd Int. ACM SIGIR Research and Development in Information Retrieval*, Berkley (1999)
13. Marcus, D.: The rhetorical parsing, summarization, and generation of natural language text, PhD Thesis, Dep. of Computer Science, University of Toronto (1998)
14. Marcus, D.: The Theory and practice of Discourse Parsing summarization. Institute of technology, Massachusetts (2000)
15. Yeh, J.Y., Ke, H.R., Yang, W.P., Meng, J.H.: Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing and Management* (2005)

16. Shen, D., Sun, J.T., Li, H., Yang, Q., Chen, Z.: Document summarization using conditional random fields. In: Proc. IJCAI 2007 (2007)
17. da Cunha, I., Fernández, S., Velázquez Morales, P., Vivaldi, J., SanJuan, E., Torres-Moreno, J.-M.: A New Hybrid Summarizer Based on Vector Space Model, Statistical Physics and Linguistics. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 872–882. Springer, Heidelberg (2007)
18. Radev, R., Jing, H., Stys, M., Tam, D.: Centroid-based summarization for multiple documents. 1st Int. Journal Information Processing and Management (2004)
19. García, R., Ledeneva, Y., Gelbukh, A., Gutierrez, C.: An assessment of Word Sequence Models for Extractive Text Summarization. Research in Computing Science (38), 253–262 (2008)
20. García, R., Ledeneva, Y., Mendoza, G., Hernandez, A., Chavez, J., Gelbukh, A., Tapia, L.: Comparing commercial tools and state-of-the-art methods for generating text summaries. In: Eighth Mexican International Conference on Artificial Intelligence, México, pp. 92–96 (2009)
21. Ledeneva, Y., Sidorov, G.: Recent advances in Computational Linguistics. Informatica, International Journal of Computing and Informatics 3871(34), 3–18 (2010) ISSN: 1854-3871
22. Ledeneva, Y., García, R., Gelbukh, A.: Multi-document summarization using Maximal Frequent Sequences. Research in Computer Science 47 (2010) ISSN 1870-4069
23. Lin, C.: ROUGE: A package for automatic evaluation of summaries. In: Proceedings of the Association for Computational Linguistics, Workshop, on Text Summarization, pp. 74–81 (2004)
24. Lin, C., Hovy, E.: Manual and Automatic evaluation of summaries, In: Proceedings of the Workshop on Automatic Summarization (including DUC 2002), vol. I, pp. 71–78. Association for Computational Linguistics on Human Language Technology (2002)
25. DUC Document Understanding Conference 2002 (2002),
<http://www-nlpir.nist.gov/proyect/duc>
26. Garcia, R., Martinez, F., Carrasco, A.: Finding maximal sequential patterns in text document collections and single documents. Informatica, International Journal of Computing and Informatics (34), 93–101 (2010) ISSN: 1854-3871
27. Lin, C.Y., Hovy, E.: Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In: Proceedings of HLT-NAACL, Canada (2003)
28. Mihalcea, R.: Graph-based Ranking Algorithms for Sentence Extraction. In: Applied to Text Summarization. University of North Texas, Texas (2004)
29. Ledeneva, Y.: Automatic Language-Independent Detection of Multiword Descriptions for Text Summarization, National Polytechnic Institute, PhD. Thesis, Mexico (2009)
30. Sidorov, G.: Lemmatization in automatized system for compilation of personal style dictionaries of literary writers. “Word of Dostoyevsky”, Russian Academy of Sciences, 266–300 (1996)
31. Gelbukh, A., Sidorov, G.: Approach to Construction of Automatic Morphological Analysis Systems for Inflective Languages with Little Effort. In: Gelbukh, A. (ed.) CICLING 2003. LNCS, vol. 2588, pp. 215–220. Springer, Heidelberg (2003)
32. Sidorov, G., Barrón-Cedeño, A., Rosso, P.: English-Spanish Large Statistical Dictionary of Inflectional Forms. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010), Valletta, Malta, pp. 277–281. European Language Resources Association (ELRA) (2010)

Discourse Segmentation for Sentence Compression

Alejandro Molina¹, Juan-Manuel Torres-Moreno¹, Eric SanJuan¹, Iria da Cunha², Gerardo Sierra³, and Patricia Velázquez-Morales⁴

¹ LIA-Université d'Avignon

² IULA-Universitat Pompeu Fabra

³ GIL-Instituto de Ingeniería UNAM

⁴ VM Labs

Abstract. Earlier studies have raised the possibility of summarizing at the level of the sentence. This simplification should help in adapting textual content in a limited space. Therefore, sentence compression is an important resource for automatic summarization systems. However, there are few studies that consider sentence-level discourse segmentation for compression task; to our knowledge, none in Spanish. In this paper, we study the relationship between discourse segmentation and compression for sentences in Spanish. We use a discourse segmenter and observe to what extent the passages deleted by annotators fit in discourse structures detected by the system. The main idea is to verify whether the automatic discourse segmentation can serve as a basis in the identification of segments to be eliminated in the sentence compression task. We show that discourse segmentation could be a first solid step towards a sentence compression system.

1 Introduction

Automatic summarizers have advanced to the point that they are able to identify, with remarkable precision, the sentences that contain the most essential information for any given text. However a great deal of irrelevant information is included due to its placement in the same high-scoring sentences. This leads to an excessive waste of space in the final summary. Hence, a finer analysis is needed to prune the superfluous information while retaining that is relevant [1]. Sentence compression is intended to produce a grammatical condensed sentence that preserves important content. It represents an important resource for automatic summarization systems [2]. Indeed, some authors argue that this task could be a first step towards abstract summary generation [3]. There are expectations for sentence compression improving summarization systems. Nevertheless, there is evidence that individually compressing the sentences of the summary should produce worse results than not compressing the sentences at all [4]. We agree with [4], who hypothesizes that sentence compression systems need to take the context into account. In this paper, we research the relationship between discourse and sentence compression. First, we provide an overview of sentence compression

and discourse segmentation in Section 2. Second, we present our corpus and a discourse segmenter in Section 3. Then, we detail analysis and results using the segmenter over our corpus in Section 4. Finally, we conclude and present the future research directions in Section 5.

2 Related Work

In this section, we present related research in sentence compression and discourse segmentation. We direct more attention to studies related to discourse units.

2.1 Sentence Compression

There are some interesting early studies in sentence compression with regard to its applications. A “Telegraphic Text Reduction” method is described by [5]. Later, [1] presented a non-extractive summarization method capable of generating headlines of any size. A more consistent work was accomplished by [6], whom showed that two of the most important operations done by humans in abstract summarization are sentence reduction and sentence combination.

The Noisy Channel model (commonly used in the area of Statistical Machine Translation) was adopted for sentence compression, for the first time by [7]. The authors of this last work consolidated a well defined task set-up and included a standard corpus for sentence compression¹ and described a method of evaluation where future research could be compared. [8] discussed that the lack of examples in the standard Ziff-Davis corpus is a cause for the generation of poor compressions. [9] proposed to compress sentences by regarding of context. Its authors did not use the Ziff-Davis corpus because their discourse constraints needed context annotations. They indicated annotators to delete individual words from each sentence [10]. For the first time, a sentence compression corpus, were annotated by humans considering the context. Nonetheless, the criteria used to elicit the compressions remain quite artificial for summarization. Humans are more likely to drop long phrases in an abstract, such as is indicated by [11]. Recent studies have found good results by concentrating on clauses, instead of isolated elements, for the generation of candidates for elimination. The algorithm of [12] first divides sentences into clauses prior to any elimination and then the compression candidates are scored by Latent Semantic Analysis [13]. A critical problem, in this last study, was that sometimes the main subject of the sentence was removed. A “discourse chunking technique” were presented by [14], as an alternative to discourse parsing, thereby showing a direct application to sentence compression. They plausibly argued that, while discourse parsing at document-level stills poses a significant challenge, sentence-level discourse models have shown accuracies comparable to human performance [15].

Finally, we would like to mention some interesting studies in sentence compression in languages other than English. In [16] and [17] the authors show

¹ Using 1,067 pairs (sentence, compressed sentence) from Ziff-Davis corpus. All of the compressions were extracted artificially from 4,000 articles.

interesting results based on different methods of statistical physics applied to documents in French. In [18] and [19] the authors evaluated sentence compression in Spanish as an optimization method for several automatic summarizers. In [20] a system for summarizing subtitles in Dutch and English is described. In [21] sentence simplification phenomena are studied for Portuguese.

2.2 Discourse Segmentation

The term “segmentation”, in discourse theory, refers to divide sentence in several units maintaining discourse sense. As [22] state: “Discourse segmentation is the process of decomposing discourse into Elementary Discourse Units (EDUS), which may be simple sentences or clauses in a complex sentence, and from which discourse trees are constructed”. Thus, one sentence constitutes an EDU, but sometimes it can contain more EDUS. Example 1 shows one sentence corresponding with a single EDU, and Example 2 shows one sentence with three EDUS (marked in square brackets). This characteristic allows to consider segmentation at the sentence level, i.e., to decompose a sentence into EDUS using only local information. In this work, we use the sentence level segmentation approach.

Example 1. [The design and management of terminological databases poses theoretical and methodological problems.]

Example 2. [In today's society, there are two apparently contradictory trends:] [on the one hand, there is a growing need for harmonization at international level, due to continuous economic, political, social and cultural links and exchanges.] [but on the other hand, there is a recognition of diversity in all areas of human life.]

Discourse segmentation is the first stage for discourse parsing (the following two stages are detection of rhetorical relations and building of rhetorical trees). The fundamental theoretical framework for research on automatic discourse parsing is the Rhetorical Structure Theory (RST) by [23]. RST is an independent-language theory of textual organization that considers that one text can be divided in EDUS, being nuclei or satellites. Nuclei are segments providing the most important information with regard to the purposes of the author, while satellites are elements depending on the nuclei and giving an additional information about them. Traditional research considers that all satellites can be eliminated without losing information [24]. However, this assumption is not totally true, as it is discussed in [25]. Moreover, results are different if we consider whole-text relations or intra-sentence relations. For example, if a CONDITION satellite is eliminated (first EDU in Example 3), it is impossible to understand the meaning of the sentence.

Example 3. [If we want to develop an adequate discourse segmenter,] [it is necessary to have a syntactic parser.]

Nowadays there are automatic discourse segmentation systems for several languages: English [22], Brazilian Portuguese [26], Spanish [27] and French [28]. All of them require some syntactic analysis of the sentences.

3 Methodology

In this section, we first detail the elicitation of the corpus. Then we describe a discourse segmenter for Spanish.

3.1 Corpus Centered in Context

Due to the lack of corpora for sentence compression in Spanish, we have created a multi-genre corpus. In our corpus, each phrase was manually compressed taking into account its context as well as the important information to be retained (as opposed as to compress sentences in isolation). Four genres were selected with the intention to represent widely different registers of language: Wikipedia sections, brief news, scientific abstracts and short stories. Each genre is represented by 20 texts composed of no more than 50 phrases each one. Brief news were chosen from three news web sites in Spanish². Scientific abstracts were randomly selected from different sites of specific areas like Psychology, Natural Language Processing, Engineering, Economics and Law. The short stories are authored from Augusto Monterroso, Marco Denevi, Macedonio Fernández, Julio Torri and Rubén Darío. All of these texts are available on the Web. The corpus has 392 sentences in Wikipedia sections, 840 sentences in brief news, 270 sentences in scientific abstracts and 550 sentences in short stories.

The texts in each genre were distributed among two annotators, who were required to read them carefully and to compress them, sentence by sentence, following some few simple instructions listed in Figure 1.

For each line in the text, delete the irrelevant information in accordance with the general context of the document; try to eliminate as many elements as possible (including punctuation marks) taking into account that the full text should be readable after compression of the sentences :

- Do not rewrite.
 - Do not change the order.
 - Do not replace.
 - New versions of the sentences should be grammatical.
 - New versions must retain the original meaning of the sentence before being altered.
 - New versions must retain the original meaning of the text before compressions.
-

Fig. 1. Instructions for corpus annotation

3.2 DiSeg

After the elicitation and manual compression of the corpus, we applied over it a Spanish discourse segmenter called DiSeg [27]. This system detects discourse boundaries in sentences, offering RST discourse segments (EDUS). This

² *La Jornada* (www.jornada.unam.mx), *El Universal* (www.eluniversal.com.mx) and *Milenio* (www.milenio.com)

segmentation tool is based on a set of discourse segmentation rules using lexical and syntactic features. First, a text is preprocessed with sentence segmentation, POS tagging and shallow parsing modules using the Freeling tool-kit[29]. Then, a xml file is generated with discourse markers annotations. Finally, several rules are applied to this xml file. The rules are based on: discourse markers, as “while” (*mientras que*), “although” (*aunque*) or “that is” (*es decir*), which usually mark the relations of CONTRAST, CONCESSION and REFORMULATION, respectively; conjunctions, such as “and” (*y*) or “but” (*pero*); adverbs, as “any-way” (*de todas maneras*); verbal forms, as gerunds, finite verbs, etc.; punctuation marks, as parenthesis or dashes. The precision of DiSeg was evaluated using as gold standard a corpus including medical texts (obtaining an 80% of F-score) and terminological texts (obtaining a 91% of F-score). The gold standard, as well as the segmenter, can be downloaded on-line though the following link: <http://daniel.iut.univ-metz.fr/DiSeg/>.

4 Experiments and Results

In this section we show the quantitative and qualitative analysis of our manually compressed corpus.

4.1 Quantitative Analysis

First, we parsed our corpus with DiSeg at a sentence level. Then, we extracted all of the deleted passages that were removed from an original sentence, according to the rules described in Section 3.1. Finally, we classified them into three classes:

1. Human-deleted passages corresponding to EDUS detected by DiSeg.
2. Human-deleted passages not corresponding to EDUS detected by DiSeg.
 - (a) Passages with discourse sense.
 - (b) Passages without discourse sense.

Class 1 contains human-removed passages corresponding to DiSeg EDUS, that is, segments containing a discourse marker detected by the system. As expected, there are some passages containing a discourse marker that do not match exactly DiSeg EDUS. However, most of the passages in Class 1, do Annotators were used more to remove complete discourse passages than just markers. Table 1 shows the average of the matching proportion between passages included in Class 1 and DiSeg EDUs (defined as $(\text{passage length}) / (\text{EDU length})$), as well as the percentage of passages in Class 1 that full-matched a DiSeg EDU.

Class 2 includes human-removed passages not corresponding with DiSeg EDUS. This class is divided into two sub-classes: Class 2a which includes elements that could be considered discourse segments (because they have a discourse sense), but that are not detected by DiSeg, as they do not match with the segmentation criteria of the system. Class 2b is comprised of elements with no discourse sense, that is, short units as substantives, verbs, adverbs, adjectives, adverbial phrases,

Table 1. Matching proportion between passages in Class 1 and DiSeg EDUS

	Average matching proportion with edus	Full-match
Wikipedia	0.91	73%
News	0.92	73%
Scientific	1.00	100%
Stories	0.81	57%

Table 2. Proportions of deleted content in three classes

	Class 1 (%words)	Class 2a (%words)	Class 2b (%words)
Wikipedia	31.55	29.57	38.88
News	34.95	16.47	48.58
Scientific	30.26	17.26	52.48
Stories	20.68	09.06	70.26

Table 3. Nucleus-Satellite proportions in DiSeg Segments

	Nuclei DiSeg Segments	Satellites DiSeg Segments
Wikipedia	8 (29 %)	20 (71 %)
News	9 (13 %)	58 (87 %)
Scientific	0 (0 %)	8 (100 %)
Stories	4 (40 %)	6 (60 %)

punctuation marks, etc. In Section 4.2 we show some examples extracted from our corpus.

In Table 2 we show the distribution of deleted content, in terms of words, belonging to each of the three classes described above. We mark in bold the highest percentages for each class.

The passages with discourse sense ($\text{Class 1} \cup \text{Class 2a}$) seem important for compression, if we consider that they represent approximately half of the volume that annotators have removed from the whole corpus. We verify some differences with regard to genre. As expected, encyclopedic and journalistic texts tend to contain more removable discourse passages, reflected in explanations or extra information. By contrast, literary texts express the information in a more subtle way. The removal of isolated elements, such as adjectives and adverbs, is more appropriate for this genre. Regarding science texts, we consider that results are affected by the fact of choosing abstracts: summaries tend to contain simple phrases. Thus, we have not found as many segments as expected.

In order to have an idea of whether all satellites have been systematically deleted, we have divided Class 1 into two types: Nuclei and Satellites. Table 3 shows the proportion for each segment type. Most of eliminated EDUS were satellites, however, some nuclei were also deleted.

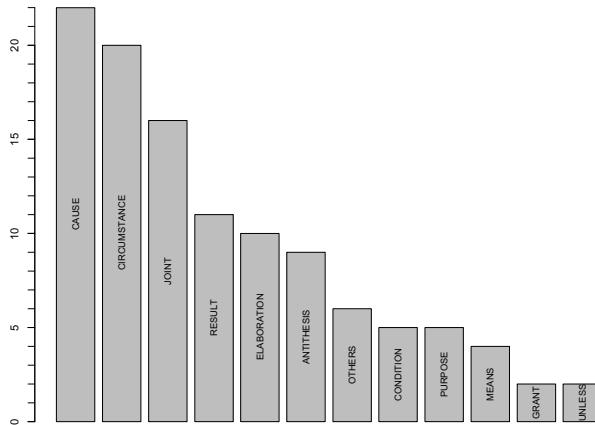


Fig. 2. Number of occurrences of DiSeg segments classified by relation

The number of occurrences of DiSeg segments (Class 1) classified by relation³ is showed in Figure 2.

It is observed that there are some relations more likely to be removed than others. Maybe this information could be used for recognizing candidate elements to be deleted in sentence compression. In the experiments carried out over our corpus, we find that the most eliminated passages correspond with: satellites of CAUSE (27.85%), satellites of CIRCUMSTANCE (25.32%), nuclei of JOINT (20.25%), satellites of RESULT (13.92%), and satellites of ELABORATION (12.66%).

4.2 Qualitative Analysis

After the quantitative analysis of the manually compressed corpus, we carried out a qualitative analysis to understand which elements tend to be removed during the compression. Along this section, we use the term “discourse marker” in a general way. We do not follow more strict classifications as the ones described by [30]. With regard to DiSeg segments (Class 1), we detect several cases. It is interesting to note that it is usually possible to assign a rhetorical relation and nuclearity (that is, if an EDU is nucleus or satellite) to a EDU, only by analyzing the discourse marker that it contains. In other words, it is not necessary to see the complete sentence. Example 4, showing a satellite of CAUSE, is an instance of this situation⁴.

³ In this work, we use the traditional rhetorical relations by [23]: RESULT, CAUSE, CONDITION, etc.

⁴ Examples in Spanish are real passages of our corpus. Translations are shown for reader understanding.

Example 4. [**ya que** se reducirían las interacciones entre fármacos, sus efectos adversos, y favorecería el cumplimiento de unos tratamientos que cada vez incluyen más pastillas.]

Example 4. [**since** the interactions among drugs would be reduced, their adverse effects, and it would help the fulfillment of some treatments that include more and more pills.]

In some cases, the discourse marker is ambiguous, and it could indicate more than one rhetorical relation. In Example 5, the discourse marker *cuando* (“when”) could indicate CIRCUMSTANCE or CONDITION. In these cases, it is necessary to read the complete sentence in order to understand the rhetorical meaning of the EDU.

Example 5. [Sin embargo, el uso de Internet a edades cada vez más tempranas representa no solamente una herramienta educativa útil,] [sino también puede constituir grandes peligros] [**cuando** su uso está relacionado con contenidos inapropiados para su adecuado desarrollo.]

Example 5. [However, the use of Internet at increasing earlier ages represents not only one useful educational tool] [but it can also constitute big dangers] [**when** its use is related to inappropriate contents for her adequate development.]

Sometimes, the marker is a gerund form, which is ambiguous as well. Examples 6, 7 and 8 show EDUS marked with a gerund, indicating different rhetorical relations: as RESULT (ex. 6), ELABORATION (ex. 7) and MEANS (ex. 8).

Example 6. [**limitándose** a reducir el factor de comportamiento sísmico que controla las resistencias de diseño.]

Example 6. [**being limited** to reducing the factor of seismic behavior that controls the resistances of design.]

Example 7. [**diseñando** mejoras para el equipo eléctrico traído del otro lado del océano gracias a las ideas de Edison.]

Example 7. [**designing** improvements for electrical equipment brought from the other side of the ocean thanks to Edison ideas.]

Example 8. [**hablando** acerca de la prevención necesaria.]

Example 8. [**talking** about necessary prevention.]

Most of the eliminated EDUS have an explicit discourse marker, such as *ya que* (“since”) in Example 4 or *cuando* (“when”) in Example 5. However, a few EDUS contain no discourse markers. In these cases, it is more difficult to assign a rhetorical relation to them. Example 9 is an instance of this situation.

Example 9. [se incluyeron además corredores entre las plantas hechos con tepujal, un material que ayuda a conservar la humedad en la tierra]

Example 9. [moreover, corridors among the plants made with tepujal, a material that helps keep the humidity in the land, were included]

In the majority of the cases, the eliminated EDU corresponds to a satellite (exs. 4-8), but sometimes it corresponds to a nucleus (ex. 9). This means that the satellite may not always be eliminated without a corresponding loss of the message. Furthermore, sometimes the nucleus is not essential for understanding the text, as traditional works on rhetorical analysis suggest [24]. With regard to passages removed by human annotators that do not correspond to EDUs detected by DiSeg, we find two cases: (a) units with discourse meaning and (b) units with no discourse meaning.

In the first case (a), we detected three regularities. The first regularity includes cases in which a removed passage starts by a participle form as shown in Example 10.

Example 10. [valorado en 40.000 dólares.]

Example 10. [valued at 40.000 dollars.]

The second regularity includes cases in which a removed passage corresponds to a relative clause as shown in Example 11.

Example 11. [que agrupaba los vídeos más vendidos.]

Example 11. [that brought together the most sold videos.]

The third regularity contains cases in which a removed passage has a discourse marker but it does not include a verb as shown in Example 12.

Example 12. [a causa de la malnutrición durante la ocupación alemana.]

Example 12. [because of the malnutrition during German occupation.]

DiSeg segmentation criteria do not detect passages in case (a) as EDUs. In spite of this, many of these segments were removed. We consider that the detection of these units would be useful for automatic compression tasks. Thus, an adaptation of DiSeg, in order to detect them, is advisable.

In case (b), we observe that human annotators eliminate some short units, such as adverbs (*después*, “after”), adjectives (*relevantes*, “relevant”), prepositional phrases (*con sus estudios*, “with their studies”), nominal phrases (*el honor*, “the honour”), etc. More exhaustive analysis of the corpus would be necessary in order to detect the regularities that allow these short units to be eliminated in compression tasks.

5 Conclusions and Future Work

We have presented an analysis of the relationship between discourse segmentation and sentence compression for sentences in Spanish. We have found that discourse segmentation could help in identifying segments to be eliminated in the sentence compression task. Furthermore, we have found that DiSeg is able to detect passages to be removed. Our future work, in the short term, is to carry out the adaptation of the discourse segmenter DiSeg in order to detect all

of the passages containing discourse sense. To do this, we will develop linguistic rules concerning relative clauses, participles forms and passages with other discourse markers. Our final goal is to create a sentence compression system based on this new adapted version of DiSeg. We propose that, in our future sentence compression approaches, we must consider two granularities. Firstly, in a coarse-grained level, irrelevant discourse segments must be removed. It is proved that some satellites can be deleted without disrupt the sentence. Secondly, in a fine-grained level, some short elements must be deleted, as long as grammar is not affected. However, deletion approaches for short elements are more prone to grammar or sense issues. In order to tackle a fine-grained level in future work it must be necessary to have information about human agreement in shorts elements deletions.

Acknowledgments. We thank Yanin Molko and Adriana Careaga for their help with the corpus annotation. This work was partially supported by a CONACyT grant 211963 and projects RICOTERM (FFI2010-21365-C03-01) and APLE (FFI2009-12188-C05-01).

References

1. Witbrock, M.J., Mittal, V.O.: Ultra-Summarization: A Statistical Approach to Generating Highly Condensed Non-Extractive Summaries. In: Proceedings of the 22nd International Conference on Research and Development in Information Retrieval, pp. 315–316. University of California, Berkeley (1999)
2. Jing, H.: Sentence reduction for automatic text summarization. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, pp. 310–315. Association for Computational Linguistics (2000)
3. Knight, K., Marcu, D.: Statistics-based summarization-step one: Sentence compression. In: Proceedings of the National Conference on Artificial Intelligence, pp. 703–710. AAAI Press, MIT Press, Menlo Park, Cambridge (2000)
4. Lin, C.: Improving summarization performance by sentence compression-a pilot study. In: Proceedings of the 6th International Workshop on Information Retrieval with Asian Languages, pp. 1–8 (2003)
5. Grefenstette, G.: Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In: Working notes of the AAAI Spring Symposium on Intelligent Text summarization, pp. 111–118 (1998)
6. Jing, H., McKeown, K.: The decomposition of human-written summary sentences. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 129–136. ACM (1999)
7. Knight, K., Marcu, D.: Summarization beyond sentence extraction: A probabilistic approach to sentence compression. Artificial Intelligence 139, 91–107 (2002)
8. Galley, M., McKeown, K.: Lexicalized markov grammars for sentence compression. In: Proceedings of the NAACL/HLT, pp. 180–187 (2007)
9. Clarke, J., Lapata, M.: Modelling compression with discourse constraints. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 1–11 (2007)

10. Clarke, J., Lapata, M.: Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research* 31, 399–429 (2008)
11. Pitler, E.: Methods for sentence compression. Technical Report MS-CIS-10-20, University of Pennsylvania (2010)
12. Steinberger, J., Jezek, K.: Sentence Compression for the LSA-based Summarizer. In: Proceedings of the 7th International Conference on Information Systems Implementation and Modelling, Citeseer, pp. 141–148 (2006)
13. Dumais, S.: Latent semantic analysis. *Annual Review of Information Science and Technology* 38, 188–230 (2004)
14. Sporleder, C., Lapata, M.: Discourse chunking and its application to sentence compression. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 257–264. Association for Computational Linguistics (2005)
15. Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, NAACL 2003, vol. 1, pp. 149–156. Association for Computational Linguistics, Stroudsburg (2003)
16. Waszak, T., Torres-Moreno, J.M.: Compression entropique de phrases contrôlée par un perceptron. In: *Journées Internationales d'Analyse Statistique des Données Textuelles* (JADT 2008), Lyon, pp. 1163–1173 (2008)
17. Fernández, S., Torres-Moreno, J.M.: Une approche exploratoire de compression automatique de phrases basée sur des critères thermodynamiques. In: *Actes de la Conférence sur le Traitement Automatique du Langage Naturel* (2009)
18. Molina, A., da Cunha, I., Torres-Moreno, J., Velazquez-Morales, P.: La compresión de frases: un recurso para la optimización de resumen automático de documentos. *Linguamática* 2, 13–27 (2011)
19. da Cunha, I., Molina, A., Torres-Moreno, J., Velazquez-Morales, P.: Optimización de resumen automático mediante compresión de frases. In: Proceedings of the XXVIII Congreso Internacional de la Asociación Española de Lingüística Aplicada, AESLA 2010 (2010)
20. Daelemans, W., Höthker, A., Sang, E.: Automatic sentence simplification for subtitleing in dutch and english. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, Citeseer, pp. 1045–1048 (2004)
21. Aluísio, S., Specia, L., Pardo, T., Maziero, E., Fortes, R.: Towards brazilian portuguese automatic text simplification systems. In: Proceeding of the Eighth ACM Symposium on Document Engineering, pp. 240–248. ACM (2008)
22. Tofiloski, M., Brooke, J., Taboada, M.: A syntactic and lexical-based discourse segmenter. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, ACLShort 2009, pp. 77–80. Association for Computational Linguistics, Stroudsburg (2009)
23. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. *Text* 8, 243–281 (1988)
24. Marcu, D.: The rhetorical parsing of unrestricted texts: a surface-based approach. *Comput. Linguist.* 26, 395–448 (2000)
25. da Cunha, I.: Hacia un modelo lingüístico de resumen automático de artículos médicos en español. Tesis; 23. Institut Universitari de Lingüística Aplicada, Barcelona, Spain (2008)

26. Mazeiro, E., Pardo, T., Nunes, M.: Identificação automática de segmentos discursivos: o uso do parser palavras. Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional. Universidade de São Paulo, São Carlos (2007)
27. da Cunha, I., SanJuan, E., Torres-Moreno, J.M., Lloberes, M., Castellón, I.: Discourse Segmentation for Spanish Based on Shallow Parsing. In: Sidorov, G., Hernández Aguirre, A., Reyes García, C.A. (eds.) MICAI 2010, Part I. LNCS, vol. 6437, pp. 13–23. Springer, Heidelberg (2010)
28. Afantinos, S.D., Denis, P., Muller, P., Danlos, L.: Learning recursive segments for discourse parsing. CoRR abs/1003.5372 (2010)
29. Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., Padró, M.: FreeLing 1.3: Syntactic and semantic services in an open-source NLP library. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), pp. 48–55 (2006)
30. Portolés, J.: Marcadores del discurso. Letras (Editorial Ariel). Ariel (2001)

Heuristic Algorithm for Extraction of Facts Using Relational Model and Syntactic Data

Grigori Sidorov¹, Juve Andrea Herrera-de-la-Cruz¹,
Sofía Galicia-Haro², Juan Pablo Posadas-Durán¹, and Liliana Chanona-Hernandez³

¹Natural Language and Text Processing Laboratory,
Center for Computing Research (CIC),
National Polytechnic Institute (IPN),
Av. Juan Dios Batiz, s/n, Zacatenco, 07738,
Mexico City, Mexico

²Faculty of sciences, Autonomous National University of Mexico (UNAM),
Mexico City, Mexico

³Engineering faculty (ESIME),
National Polytechnic Institute (IPN),
Av. Juan de Dios Bátiz, Zacatenco, 07738,
Mexico City, Mexico
www.cic.ipn.mx/~sidorov

Abstract. From semantic point of view, information is usually contained in small units, called facts that are usually smaller than sentences. Identification of these facts in a text is not a trivial task. We present a heuristic algorithm for extraction of facts from sentences using a simple representation based on a relational data model. We focus our study on texts that contain a lot of facts by their nature: structured textbooks. The algorithm is based on data obtained by a syntactic analyzer. The obtained facts can be useful for information retrieval tasks, automatic summarization, etc. Our experiments are conducted for Spanish language. We obtained better results than the similar methods.

Keywords: fact extraction, learning by reading, syntactic analysis, relational data model.

1 Introduction

Problem of extraction of facts from texts written in natural language is very important for any kind of knowledge processing tasks like, for example, information retrieval, question-answering, automatic generation of summaries, etc.

Nowadays, predominant strategies for fact extraction are related with usage of statistical methods (see, for example, [2]) that is, in fact, extraction based on patterns, when certain data is extracted that fills the predefined data structure. For example: for persons, their date of birth, place of birth, etc.; for films, their producer, date of production, principal actors, etc. [5, 7, 11, 15].

Another approach that is rather similar to what we are doing is the so called learning through reading [1, 4, 6, 14]. The main idea of this approach is to convert

syntactic structures into a logical form. There exist several systems that perform an extraction of this kind of semantic information. However, they have certain limitations: they use very specific and limited semantic representation – the first order logic that restricts them to be used by applications that also are based on the first order logic. Also, they do not apply heuristics over the obtained syntactic tree.

In this work, we consider “a fact” as a certain affirmation about some entities, thus, the fact contains a predicate (usually a verb) and its arguments. For example, from the phrase *Impulse depends on mass that the ball has*, two facts are obtained:

Impulse depends on mass

Ball has mass

We use relational database form of representation of the obtained data and develop certain rules and heuristics that allow extraction of facts on the basis of mere syntactic information. We base our research on structured textbooks that are usually focused on a certain area of study, and that are divided into short sections and paragraphs that contain lots of facts due to the didactical nature of the book. Another possibility would be to use Wikipedia. Our experiments were conducted for Spanish language. We made manual evaluation of results.

The structure of the paper is as follows. First, we describe some similar systems related to learning through reading. Then, we present the algorithm and the heuristics that are used in it. Finally, we describe the obtained results and draw some conclusions.

2 Related Work

In this section, we present works related to learning through reading.

The system proposed in [14] transforms the English phrases into first order logic representation using information about syntactic dependencies.

For each word one the following syntactic roles is obtained: principal element, subject, direct object, indirect object, complement, preposition, determinant, nominal attribute, conjunction, etc.

Then some filters are applied that eliminate the words and their corresponding structures that do not correspond to predicates, like determinants etc.

After this, logical forms are identified that correspond to dependents of the word being processed. For representation of the logical forms two types of predicates are used: one that corresponds to entities (e) and the other that corresponds to events (x). For example, let us take the phrase

Some students like to study in the mornings.

The entities are simple predicates that have only one argument. In the phrase above, the entity is: *morning:n (x)*.

The verbal predicates are constructed using the dependencies of the subject, direct object, circumstantial, indirect object, complement, and preposition. For example, the verb *like* is transformed into a predicate *like:v (e4, x3, e6)*.

The system was tested for 300 phrases in English. The precision was calculated at two levels: at the level of the argument the precision was 76.4%, and at the level of predicate, it was 84%.

Another approach proposed in [6] presents the model based on rules and machine learning for similar type of transformations as above. The well-known system FreeLing [12] is used for syntactic analysis. Corpus used for training is in Spanish. It contains syntactic, semantic and pragmatic annotations that are used for automatic learning. At the learning stage, the system finds out which formal logic argument corresponds to the given constituent. For example, *Juan vuela desde Tokio hasta Nueva York* (*John flies from Tokio to New York*):

Juan (*John*) [*P*](*x1*)
volar (*flies*) [*V*](*e1 x1*)
desde (*from*) [*P*](*e1 x2*)
Tokio [*N*](*x2*)
hasta (*to*) [*P*](*x2 x3*)
Nueva York [*N*](*x3*)

Or another example, *El baloncesto y el tenis son grandes deportes* (*Basketball and tennis are great sports*):

baloncesto (*basketball*) [*N*](*x1*)
y (*and*) [*C*](*x3 x1 x2*)
tenis (*tennis*) [*N*](*x2*)
ser (*be*) [*V*](*e1 x3 x4*)
grande (*great*) [*A*](*x4*)
deporte (*sport*) [*N*](*x4*)

The precision of the system is reported as 74%.

One more approach is described in [4] and is related to construction of the algorithm that generates what the authors call nuggets. Nuggets are independent pieces of information with meaning.

Usually, there are many nuggets in each phrase. Let us see an example, *The Danube at Cernavoda village, where the reactor is located, fell to a depth of less than three meters on Saturday, down from its usual level of almost seven meters*.

Semiautomatic extraction gives the following nuggets:

1. *Danube*.
2. *Danube is at Cernavoda village*.
3. *Danube fell*.
4. *Danube fell to a depth of less than three meters*.
5. *Danube fell on Saturday*.
6. *Danube fell down from its usual level of almost seven meters*.

Recursive procedure based on syntactic analysis that obtains more complex nuggets at each step is used. Precision of 74% is obtained as compared with human evaluation.

3 Proposed Method

We developed a heuristic algorithm that extracts facts from texts. It uses a simple relational database structure as data representation and several rules and heuristics for processing of syntactic data. For the moment, the heuristics are related to processing of coordinated and subordinated clauses, and some special syntactic phenomena.

One of the interesting problems is what facts are worth extracting, because many texts really contain very few fact related information. That was our motivation to use textbooks.

Another interesting question also remains open: if an entity (that the algorithm detects) refers to an instance of object, i.e., a specific object, or to a class of objects, like it is in the case of universal quantifier vs. existential quantifier. We think that we should mark this distinction when it will be possible to determine this.

We construct a table of predicates together with their possible syntactic relation, a table of entities, and a table of facts that contains the predicate, the entity, the identifier of the fact, and the level of the corresponding relation in the syntactic tree (the depth in a tree). Information about the syntactic level allows for generalization, because we can ignore the lower levels if necessary. See the example in Section 4 (Tables 1 to 3). In our opinion, the usage of relational database model is more intuitive than first order logic. Other idea behind using the relational database model is that we can immediately answer queries about a specific entity and can generalize easily.

As compared to nuggets, we think that the nuggets are not defined well, i.e., they are sometimes too specific and sometimes too general. For example, nugget 6 (see above) should be more general and really contains two facts: *Usual level of Danube is almost seven meters*, and *Danube fell from its usual level*. On the other hand, we will not consider nuggets 1 and 3 as facts (though nugget 3 could be deduced).

We converted the texts into their syntactic representations using the Connexor tool¹ (commercial syntactic analyzer). Still, our method does not depend significantly on the analyzer used. In fact, we are not satisfied with results that Connexor gives for Spanish (it is much better for English, though); in future, we prefer to use the well-known tool FreeLing [12] for Spanish. Another possibility may be DILUCT system [3].

We applied preprocessing that consisted in removing tables, figures and equations from the texts and splitting them into phrases. Then we processed the texts using Connexor. For example², *[The] documents written in most ancient times were found in Mesopotamia and Egypt*.

1 The | Los los det>2 @PREMOD DET MSC PL

2 documentos | documentos documento subj>6 @NH N MSC PL

3 written | escritos escrito ads>2 @POSTMOD A MSC PL

4 <in> most | más mucho ad>5 @PREMOD ADV CMP

5 ancient times | antiguos antiguo ads>3 @POSTMOD A MSC PL

¹ www.connexor.com, examples can be analyzed on-line for free.

² The original phrase is in Spanish.

6 were | fueron ser vch>7 @AUX V IND PRET PL P3
 7 found | encontrados encontrar @MAIN V PCP PERF MSC PL
 8 in | en en pm>9 @PREMARK PREP
 9 Mesopotamia | Mesopotamia mesopotamia loc>7 @NH Heur N SG Prop
 10 andl y y cc>9 @CC CC
 11 Egypt | Egipto egipo cc>9 @NH N MSC SG Prop

From this input data, we extract two types of elements, from which facts are constructed: predicates and entities.

Entities represent subject and objects/complements of verbs. The corresponding words are usually nouns (sometimes with prepositions) and adjectives. Usually, they are dependents of verbs. We save entities and predicates in separate tables, and also have a table where we save the obtained facts that consist normally of a predicate and several entities, probably at different syntactic levels.

The following should be considered while detecting entities. Sometimes, subject depends on auxiliary verb. This phenomenon should be processed separately, for example, like in the phrase *scripts were found*, the fact is *to find scripts*. In case of prepositions, we store them as part of facts. Another possibility is to store them with predicates or with entities. These representations seem to be equivalent.

3.1 Main Algorithm

First stage of our algorithm has the following steps:

1. Find the main verb in the syntactic structure. If we have an auxiliary verb that Connexor marks as the head of the phrase, then we skip it and pass to the dependent verb in infinitive, i.e., to the main verb.
2. Try to find subject of the main verb (that has syntactic mark SUBJ), and extract its main element (that will be the entity). Add the predicate (the verb) and the entity into the corresponding tables; add them as part of the fact into the table of facts.
3. Find complements that depend from the main verb and extract their entities in the way described below. After the extraction, add the predicate (the verb) and the entity into the corresponding tables; add them as other part of the fact into the table of facts. The following type of complements are processed:
 - a) Using syntactic mark COMP. For example, *The first culture in Mesopotamia was [Sumerian culture]*.
 - b) Using syntactic mark LOC, when it is a location. In Spanish, it usually has a preposition. For example, *The most ancient written documents was found [in Mesopotamia]*.
 - c) Using syntactic mark OBJ, for specification of an object: *India represents [a vast territory]*.
 - d) Using syntactic mark CNT that corresponds to a purpose or a reason: *Egyptians were characterized [by their beliefs] related to the death*.

- e) Using syntactic mark MOD that expresses a modifier: *The bibliography is an alphabetical listing [of the sources] consulted for preparation of the work.*
- f) Using syntactic mark AGT that specifies an agent for passive voice phrases, *A word can be formed [by one a various morphemes].*
- g) Using syntactic mark COM that corresponds to a comitative case (accompanied with something), *The bibliography is structured [with data] of bibliographic cards corresponding to the texts.*

In general, these steps allow extraction of facts in simple syntactic structures. As additional steps of the algorithm, we try to apply the following heuristics if possible.

3.2 Subordinate Entities

Heuristic 1 is applied for processing of subordinate entities. For example, *The book of the dead is another writing that was found in various Egyptian tombs.*

The following steps are performed:

1. Verify if subordinate phrase is present. If yes, continue.
2. Search for a verb that is the head for a relative pronoun.
3. The subject of the new fact is the previously extracted subject for the verb.
4. Search for the complements that are dependent from the verb (step 2).
5. New fact is constructed using the verb, its subject and the complement(s).

3.3 Treatment of “Purpose”

Heuristic 2 is developed for treatment of the “purpose”. This heuristic is applied for grammar construction SUBJECT - VERB - COMPLEMENT - PURPOSE (CNT). Thus, if we find mark CNT in a sentence, this rule is applied. There are two cases for this rule, depending if we have active or passive voice.

An example for active voice case: *Chemistry uses techniques for analysis of substances.*

1. Find the main verb.
2. Find its subject (SUBJ).
3. Find its complement.
4. Construct the first fact from the elements obtained at steps 1, 2, and 3.
5. Find the verb that is the head of the dependency marked as “purpose” (CNT).
6. Find its complement marked CNT.
7. Construct the second fact from the verb (step 5), subject (step 2) and complement (step 6).

A special processing is used for passive voice, for example: *Special techniques are used for analysis of substances.* In this case, the complement is already the purpose. The rest of the processing is the same.

3.4 Coordinate Entities and Clauses

Heuristic 3 is used for processing of the coordinate entities. For example, *The most ancient documents were found in [Mesopotamia and Egypt]*.

Note that we can have a set of coordinated nouns (subjects or complements), as well as a set of coordinated verbs. This heuristics depends on the manner in which the coordinated structures are represented in the syntactic tree. Heuristic 3 consists in the following steps. Note that after application of the main algorithm we have the subject, the verb and the complement.

1. Look for the elements marked as coordinated with the subject, save them in a vector₁.
2. Look for the elements marked as coordinated with the complement, save them in a vector₂.
3. Look for the elements marked as coordinated with the verb, save them in a vector₃.
4. Make all possible combinations of these vectors for construction of facts.

3.5 Treatment of “Source/Goal”

Heuristic 4 is used for processing the phrases like *Prehistory is a large period of time from appearance of the first human beings until the elaboration of the first writing materials* that contains the elements marked as source (SOU) and goal (GOA).

Heuristic 4 consists in the following:

1. Find the subject using the main algorithm.
2. Find the complement (source) marked as SOU.
3. Create the fact with the previously saved subject, the obtained complement SOU, and the predicate “STARTS”.
4. Find the complement (goal) marked as GOA.
5. Create the fact with the previously saved subject, the obtained complement GOA, and the predicate “ENDS”.

4 Evaluation of the Results

4.1 Corpus

We used in our experiments textbooks that contain many facts by its nature, unlike, say, literary work, when we can talk about descriptions, feelings, etc. Another alternative is to use Wikipedia.

We used textbooks in Spanish on history [10], language [8], and physics [12] for high school. The sentences from these books were analyzed using Connexor syntactic analyzer. We used only phrases, where Connexor constructed complete syntactic trees.

4.2 Results

Let us see an example of functioning of the system. For the phrase *La mecánica es la parte de la física que se encarga del estudio del movimiento de los cuerpos.* (*Mechanics is part of physics that deals with the study of movement of the bodies*) we obtained two facts. The obtained facts are presented in Tables 1 to 3.

Table 1. Extracted predicates for the example phrase

idPredicate	Predicate	Type
1	be	subj
2	be	comp
3	deal with	subj
4	deal with	comp

Table 2. Extracted entities for the example phrase

idEntity	Entity
1	mechanics
2	study
3	movement
4	body
5	part
6	physics

Table 3. Extracted facts for the example phrase

Predicate	Entity	Level	Preposition	Fact
1 (be, subj)	1 (mechanics)	2		1
2 (be, comp)	5 (part)	2		1
2 (be, comp)	6 (physics)	3	of	1
3 (deal with, subj)	1 (mechanics)	2		2
4 (deal with, comp)	2 (study)	5	of	2
4 (deal with, comp)	3 (movement)	6	of	2
4 (deal with, comp)	4 (body)	7	of	2

Note that the words in parenthesis are presented for illustrative purposes only. Table 3 represents two facts:

Mechanics is part of physics

Mechanics deals with study of movement of body

For evaluation, we applied our algorithm automatically to 50 sentences and compared the extracted facts with the facts extracted manually by human. A human extracted 97 facts from these sentences. The system obtained 88 facts, from which 71 were the correct ones. This gives the value of precision 80%, recall 73%, and F1-measure 76%. Our precision is slightly higher than in case of other systems (see Section 2), where precision of 74-77% is reported. Still, a question arises of how comparable the results are if other systems are developed for English language. Also, F1 measure values are unknown for other systems.

We would like to emphasize that there is no gold standard available for evaluation of systems for extraction of facts.

5 Conclusions

The aim of this work was to explore the idea of what kind of factual information can be easily extracted using simple algorithm based on syntactic information. We used intuitive data structure for representation of facts, namely, relational database.

We developed a heuristic algorithm, observing the patterns that were presented in syntactic trees obtained using Connexor tool. We made our experiments for textbooks written in Spanish. We obtained better results than similar methods.

Acknowledgements. Work done under partial support of Mexican Government (CONACYT projects 50206-H and 83270, SNI), Government of Mexico City (project PICCO10-120), European project 269180 WIQ-EI, project “Answer Validation through Textual Entailment” CONACYT-DST (India), and National Polytechnic Institute, Mexico (projects SIP 20111146, 20113295; PIFI, COFAA).

References

1. Barker, K., Agashe, B., Chaw, S.-Y., Fan, J., Friedland, N., Glass, M., Hobbs, J., Hovy, E., Israel, D., Kim, D.S., Mulkar-Mehta, R., Patwardhan, S., Porter, B., Tecuci, D., Yeh, P.: Learning by reading: a prototype system, performance baseline and lessons learned. In: AAAI 2007: Proceedings of the 22nd National Conference on Artificial Intelligence, pp. 280–286. AAAI Press (2007)
2. Calvo, H., Gelbukh, A.: Automatic Semantic Role Labeling using Selectional Preferences with Very Large Corpora. *Computación y Sistemas* 12(1), 128–150 (2008)
3. Calvo, H., Gelbukh, A.: DILUCT: An Open-Source Spanish Dependency Parser Based on Rules, Heuristics, and Selectional Preferences. In: Kop, C., Fließl, G., Mayr, H.C., Métais, E. (eds.) NLDB 2006. LNCS, vol. 3999, pp. 164–175. Springer, Heidelberg (2006)
4. Hovy, E., Kwon, N., Zhou, L.: A semi-automatic evaluation scheme: automated nuggetization for manual annotation. In: Human Language Technologies 2007: The

- Conference of the North American Chapter of the Association for Computational Linguistics, NAACL 2007, pp. 217–220 (2007)
- 5. Mann, G.: Multi Document Statistical Fact Extraction and Fusion. PhD Thesis, John Hopkins University, Maryland, 238 (2006)
 - 6. Martínez-Santiago, F., García-Cumbreras, M.: Identificación de formas lógicas en el caso del español: propuesta de un modelo basado en reglas y aprendizaje automático. In: Procesamiento del Lenguaje Natural, pp. 245–252 (2005)
 - 7. Montes-y-Gómez, M., Gelbukh, A., López-López, A.: Mining the news: trends, associations, and deviations. Computación y Sistemas 5(1), 14–24 (2001)
 - 8. Moreno, T., Moreno, G.: Lengua y Literatura 2, cuarta edn, Editorial Santillana, México (1991)
 - 9. Mulkar, R., Hobbs, J., Hovy, E., Chalupsky, H., Lin, C.: Learning by reading: Two experiments. In: Proceedings of the IJCAI Workshop on Knowledge and Reasoning for Answering Questions, KRAQ (2007)
 - 10. Nieto-López, J., Betancourt-Suárez, M., Nieto-López, R.: Historia 1, tercera edn, Sistemas Técnicos de Edición. México (1994)
 - 11. Pasca, M., Lin, D., Bigham, J., Lifchits, A., Jain, A.: Names and Similarities on the Web: Fact Extraction in the Fast Lane. In: Proc. ACL 2006 (2006)
 - 12. Padró, L., Collado, M., Reese, S., Lloberes, M., Castellón, I.: FreeLing 2.1: Five Years of Open-Source Language Processing Tools. In: Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010), ELRA, La Valletta, Malta (May 2010)
 - 13. Rincón, A., Rocha, A.: ABC de Física. Tercer curso, sexta edn, Editorial Herrero, México (1984)
 - 14. Stephen, A., Jon, P.: Dependency based logical form transformations. In: Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (2006)
 - 15. Zhao, S., Betz, J.: Corroborate and Learn Facts from the Web (2006),
<http://140.122.184.128/presentation/08-03-06/Corroborate%20and%20Learn%20Facts%20from%20the%20Web.pdf>

MFSRank: An Unsupervised Method to Extract Keyphrases Using Semantic Information

Roque Enrique López¹, Dennis Barreda², Javier Tejada²,
and Ernesto Cuadros²

¹School of System Engineering, San Agustin National University, Perú
rlopezc27@gmail.com

²School of Computer Science, San Pablo Catholic University, Perú
{dennis.barreda,jtejadac,ecuadros}@ucsp.edu.pe

Abstract. This paper presents an unsupervised graph-based method to extract keyphrases using semantic information. The proposed method has two stages. In the first one, we have extracted MFS (Maximal Frequent Sequences) and built the nodes of a graph with them. The weight of the connection between two nodes has been established according to common statistical information and semantic relatedness. In the second stage, we have ranked MFS with traditionally PageRank algorithm; but we have included ConceptNet. This external resource adds an extra weight value between two MFS. The experimental results are competitive with traditional approaches developed in this area. MFSRank overcomes the baseline for top 5 keyphrases in precision, recall and F-score measures.

Keywords: Keyphrase Extraction, Maximal frequent sequences, Semantic Graphs.

1 Introduction

Currently, keyphrases extraction task has taken big importance. This is due to exponential growth of information and the need to understand it quickly. Keyphrases of a document are the words and phrases that can precisely and compactly represent the content of the document [1]. As they represent the main topics of a document (for example, see [2]), keyphrases can be used in many Natural Language Processing (NLP) tasks, such as summarization, information retrieval, classification and clustering of documents.

Usually, keyphrases are assigned manually. Other possibilities consist in using a semiautomatic tool like described in [3], a specialized environment as presented in [4], or perform this task automatically. In scientific articles, keyphrases help readers to have a global idea of the article and in web pages they serve like metadata which describe its content. Unfortunately many others documents do not have keyphrases assigned, wasting their benefits. The main reason for the absence of keyphrases in documents is that the manual assignment is a laborious task.

As shown in [5], in recent years has reemerged interest in automatic keyphrases extraction. Different approaches have been developed to give solution to this

task, many of them have obtained very good results. However, most of them do not use semantic information. The proposed method tries to take advantage of semantic information between the words in keyphrases.

In this paper we present MFSRank, an unsupervised graph-based method for automatic keyphrases extraction using semantic information. For MFSRank performance evaluation has been used *Task # 5 of SemEval*, a collection of scientific papers, in which each document has assigned keyphrases by authors and readers. Experimental results are competitive with traditional approaches developed in this area.

The paper is organized as follows: Section 2 describes some previous work. Section 3 explains the techniques and tools used in MFSRank. Section 4 explains MFSRank method. Section 5 shows and explains the results obtained in the keyphrases extraction. Finally, Section 6 contains our conclusions and provides directions in this area.

2 Related Work

Methods for keyphrases extraction can be classified into supervised and unsupervised [6]. In this paper we have focused on unsupervised graph-based methods. Many of these methods are based on statistical information, such as term frequency (TF), inverse document frequency (IDF) and term frequency-inverse document frequency (TF-IDF). Keyphrases extraction graph-based methods have their origins in Mihalcea's work, who propose TextRank [7]. TextRank is a model that represents the text as a graph, where each vertex represents a word, and the weight assigned between two vertices represents the co-occurrence of the words in a sentence.

SingleRank [8], is a variation of TextRank. This method has three differences in relation to TextRank. First, SingleRank consider the number of co-occurrence between two words, and uses this value to calculate the weight of the edges. Second, while in TextRank only words that correspond to vertex with high ranking can be used to extract the keywords, SingleRank does not make this filter. Finally, SingleRank uses a window size of 10 instead of 2 [9].

ExpandRank [6], is another graph-based method. It uses a small set of nearest neighbors documents to provide greater knowledge, and thus, improve the keyphrases extraction from a document. Recently, Ortiz et al. presented BUAP [10], an unsupervised method, which uses two techniques: Maximal Frequent Sequences and PageRank algorithm.

3 Overview

For keyphrases extraction task, the method presented in this paper uses two techniques: Maximal Frequent Sequences and the PageRank algorithm. Moreover, Conceptnet is used as a knowledgebase.

3.1 Maximal Frequent Sequences (MFS)

In a collection of texts, the fact that some sequences of words are repeated in several sentences shows the importance of the information contained in these sequences. MFS can be useful because they could represent the most relevant parts of the texts. A maximal sequence is a sequence that is not a subsequence of any other. In other words, a maximal sequence shall not be included in any other sequence in the same order [12].

Assuming that S is a set of texts, according to [13] the formal definition of a maximal frequent sequence is:

Definition 1. *A sequence $p = a_1 \dots a_k$ is a subsequence of a sequence q if all the items a_i , $1 \leq i \leq k$, occur in q and they occur in the same order as in p . If a sequence p is a subsequence of a sequence q , we also say that p occurs in q .*

Definition 2. *A sequence p is frequent in S if p is a subsequence of at least σ documents of S , where σ is a given frequency threshold.*

Definition 3. *A sequence p is a maximal frequent sequence in S if there does not exist any sequence p' in S such that p is a subsequence of p' and p' is frequent in S .*

3.2 PageRank

PageRank algorithm [11], developed by Larry Page and Sergey Brin, is used by Google to determine the importance of a website. It is the most important factor that determines the position of the page in the search result. PageRank provides a numeric value that represents the relevance of a website on the Internet. This algorithm considers the link from one page to another as a vote or recommendation.

The PageRank algorithm builds a graph considering websites as nodes, and the input and output links as edges. The PageRank of a page V_i is calculated as:

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \quad (1)$$

Where $S(V_i)$ is the PageRank of the page V_i . d is a damping factor that can be set between 0 and 1. $S(V_j)$ are the PageRank values of each page that link to V_i . $In(V_i)$ are the pages that references to V_i . $Out(V_j)$ is the total number of page V_j output links.

3.3 Conceptnet

ConceptNet is a freely available commonsense knowledge base which supports many textual reasoning tasks, such as topic detection, word analogies, affects sensor, etc. This knowledge base has been generated automatically from 700,000

sentences of the Open Mind Common Sense Project [14]. ConceptNet represents this data as a semantic network and is composed of more than 1.6 million assertions of common knowledge which covers topics of everyday life.

One need in many textual reasoning applications is determining the context around a word. The *GetContext* function makes this task [15]. For example, to compute the top ten concepts in the contextual neighborhood of “dog” yields “bark”, “eat bone”, “guard house”, “animal”, “bite”, “catch frisbee”, “place bone beneath surface of earth”, “smell food”, “mammal” and “pet”.

4 MFSRank

The approach presented in this paper, unlike the methods mentioned in section 2, not just has considered statistical information. It has taken account semantic relationship between words that constitute a MFS. Semantic relationship between words has been obtained from Conceptnet. In essence, MFSRank has two stages: MFS Extraction and MFS Ranking. Figure 1 shows MFSRank’s architecture.

4.1 MFS Extraction

In this stage, the text has been divided in sentences. In this work we have consider the *point* as delimiter symbol of sentences. The stopwords has been omitted. Later, we have used the Porter algorithm [20] to remove suffixes from a word. Maximal frequent sequences were extracted from all stems. These sequences were formed by two or more words. Finally, a graph was constructed, where each node represents a MFS. Two MFS were linked if they appeared in the same sentence. The weight assigned to each edge has been calculated in the next stage.

4.2 MFS Ranking

The weight of the edge that links V_i and V_j nodes has been calculated as:

$$W_{i,j} = tf_{i,j} * cw_{V_i, V_j} \quad (2)$$

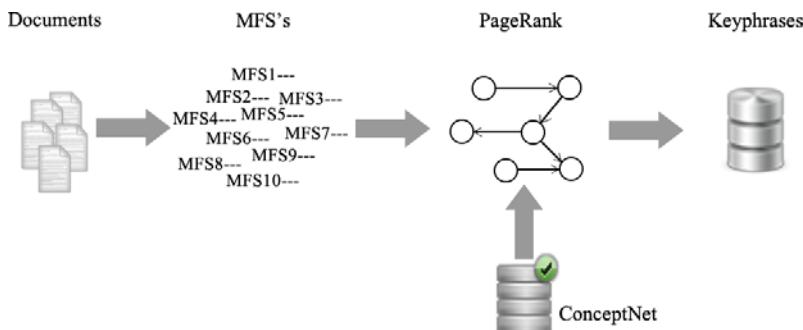


Fig. 1. Architecture of MFSRank

Where $tf_{i,j}$ is the number of times that occur V_i and V_j node in the same sentence, cw_{V_i, V_j} is the weight assigned by Conceptnet. This weight reflects the semantic similarity between two words.

As mentioned above, PageRank algorithm was used to ranking MFS. Unlike the original algorithm, MFSRank includes a weight between nodes. The importance of a MFS depends of the weight that PageRank assigned to a node according to the semantic relationship between words. The modified PageRank algorithm is shown in Equation 3.

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{W_{i,j}}{|Out(V_j)|} S(V_j) \quad (3)$$

5 Experimental Evaluation

We have used *Task # 5 of SemEval* collection to evaluate MFSRank. Specifically the *test dataset* was used, this sub-collection is formed by 100 scientific papers, which belong to the following four 1998 ACM classifications: Distributed Systems, Information Search and Retrieval, Distributed Artificial Intelligence-Multiagent Systems and Social and Behavioral Sciences-Economics. In this work we have used the baseline based on TF*IDF, Naive Bayes (NB) and Maximum Entropy (ME) methods for top 5 and top 10 candidates keyphrases given by authors and readers. Unlike MFSRank, these methods only use statistical information.

Table 1 and Table 2 show Precision, Recall and F-score obtained using MF-SRank. In both tables, the proposed method overcomes the baseline for top 5 candidates. However, for top 10 candidates, MFSRank does not overcome the baseline, because Conceptnet contains general knowledge and the collection of documents used corresponds to a specific domain (scientific articles).

Table 1. Results for keyphrases assigned by Reader

Method	top 5 candidates			top 10 candidates		
	Precision	Recall	F-score	Precision	Recall	F-score
TF*IDF	17.80%	7.39%	10.44%	13.90%	11.54%	12.61%
NB	16.80%	6.98%	9.86%	13.30%	11.05%	12.07%
ME	16.80%	6.98%	9.86%	13.30%	11.05%	12.07%
MFSRank	22.00%	9.14%	12.91%	11.90%	9.88%	10.80%

6 Conclusions and Future Work

In this paper we presented MFSRank, an unsupervised graph-based method to extract keyphrases using semantic information. The novelty of this method is the usage of the semantic relation between words. The proposed method has two stages. First, we have extracted MFS and these form the nodes of the graph.

Table 2. Results for keyphrases assigned by Author and Reader

Method	top 5 candidates			top 10 candidates		
	Precision	Recall	F-score	Precision	Recall	F-score
TF*IDF	22.00%	7.50%	11.19%	17.70%	12.07%	14.35%
NB	21.40%	7.30%	10.89%	17.30%	11.80%	14.03%
ME	21.40%	7.30%	10.89%	17.30%	11.80%	14.03%
MFSRank	26.40%	9.00%	13.42%	14.20%	9.69%	11.52%

The connection between two nodes has been established according to common statistical information and semantic relatedness. Second, MFS obtained in the first phase have ranked using the PageRank algorithm. The experimental results are competitive with traditional approaches developed in this area.

Between the main future works, we find: (1) Use a domain-specific knowledgebase, which could establish a better semantic relationship between words. (2) Select MFS using a *gap* to give more flexibility to the word sequences.

References

1. Jiang, X., Hub, Y., Lib, H.: A ranking Approach to Keyphrase Extraction. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, pp. 756–757 (2009)
2. Gelbukh, A., Sidorov, G., Guzmán-Arenas, A.: Use of a Weighted Topic Hierarchy for Document Classification. In: Matoušek, V., Mautner, P., Ocelíková, J., Sojka, P. (eds.) TSD 1999. LNCS (LNAI), vol. 1692, pp. 133–138. Springer, Heidelberg (1999)
3. Ledo Mezquita, Y., Sidorov, G., Gelbukh, A.: Tool for Computer-Aided Spanish Word Sense Disambiguation. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, pp. 277–280. Springer, Heidelberg (2003)
4. Gelbukh, A., Sidorov, G., Galicia Haro, S., Bolshakov, I.: Environment for Development of a Natural Language Syntactic Analyzer. Acta Academia 2002, 206–213 (2002)
5. Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T.: SemEval-2010 task 5: Automatic keyphrase extraction from scientific articles. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 21–26 (2010)
6. Xiaojun, W., Jianqiu, X.: Single document keyphrase extraction using neighborhood knowledge. In: Proceedings of the 23rd National Conference on Artificial Intelligence, vol. 2, pp. 855–860 (2008)
7. Rada, M., Paul, T.: TextRank: Bringing order into texts. In: Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)
8. Xiaojun, W., Jianwu, Y., Jianqiu, X.: Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 552–559 (2007)
9. Kazi, S.H., Vincent, N.: Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 365–373 (2010)

10. Roberto, O., David, P., Mireya, T., Héctor, J.: BUAP: An unsupervised approach to automatic keyphrase extraction from scientific articles. In: Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010), pp. 174–177 (2010)
11. Page, L., Brin, S., Motwani, R., Winograd, T.: The Pagerank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Libraries (1998)
12. Sandra, G., Roxana, D., Paolo, R.: Drug-Drug Interaction Detection: A New Approach Based on Maximal Frequent Sequences. Procesamiento de Lenguaje Natural 45 (2010)
13. Helena, A.M.: Discovery of Frequent Word Sequences in Text. In: Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery, pp. 180–189 (2002)
14. Liu, H., Singh, P.: ConceptNet: A Practical Commonsense Reasoning Tool-Kit. BT Technology Journal 22 (2004)
15. Liu, H., Singh, P.: Commonsense Reasoning in and Over Natural Language. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) KES 2004. LNCS (LNAI), vol. 3215, pp. 293–306. Springer, Heidelberg (2004)
16. Ledeneva, Y., Gelbukh, A., García-Hernández, R.: Keeping Maximal Frequent Sequences Facilitates Extractive Summarization. In: Sidorov, G., et al. (eds.) Advances in Computer Science and Engineering, 9th Conference on Computing (CORE 2008), Research in Computing Science, vol. 34, pp. 163–174 (2008)
17. Ian, H.W., Gordon, W.P., Eibe, F., Carl, G., Craig, G.: KEA: Practical automatic keyphrase extraction. In: Proceedings of the Fourth ACM Conference on Digital Libraries (DL 1999), pp. 254–255. ACM (1999)
18. Chong, H., Yonghong, T., Zhi, Z., Charles, X.L., Tiejun, H.: Keyphrase extraction using semantic networks structure analysis. In: Proc. of the ICDM 2006, pp. 275–284 (2006)
19. Peter, D.: Learning Algorithms for Keyphrase Extraction. Inf. Retr. 2(4), 303–336 (2006)
20. Porter, M.F.: An Algorithm for Suffix Stripping. Program 14(3), 130–137 (1980)

Content Determination through Planning for Flexible Game Tutorials

Luciana Benotti and Nicolás Bertoá

NLP Team, Universidad Nacional de Córdoba, Argentina
`benotti@famaf.unc.edu.ar, nicobertoa@gmail.com`

Abstract. The goal of this work is to design and implement an agent which generates hints for a player in a first person shooter game. The agent is a computer-controlled character which collaborates with the player to achieve the goal of the game. Such agent uses state of the art reasoning techniques from the area of artificial intelligence planning in order to come up with the content of the instructions. Moreover, it applies techniques from the area of natural language generation to generate the hints. As a result the instructions are both causally appropriate at the point in which they are uttered and relevant to the goal of the game.

1 Introduction

Nowadays, most game tutorials make the player follow a fixed script. Hence having a good or bad tutorial is still an art that depends on how good the script is at faking some freedom (for instance, by foreseeing possible reactions of the player) in order to make it more entertaining. *Natural language generation* can offer game designers techniques that, in the future, may transform tutorial design not in an art but in a science by providing efficient algorithms for managing the players knowledge and dealing with the players reactions in a dynamic way.

The goal of this work is to design and implement an agent (a computer-controlled character) which is able to generate hints that help a player advance in a game situated in a 3D virtual world. Such agent uses state-of-the-art reasoning algorithms such as *planning* in order to come up with the content of the instructions which are relevant at each point in the interaction. Furthermore, it applies techniques from situated natural language generation such as *common ground management* to generate the context-aware hints.

We believe that our contribution is interesting for the game industry as well as for the players. For players, this work is a first step towards more flexible, effective and entertaining tutorials; in order to sustain our claims we performed a human based evaluation. For the game industry, good tutorials are expensive to develop because it is costly to script all the appropriate game behaviors caused by different potential player reactions (some of which might never be triggered). Our approach can be seen as a way of automatically producing scripts that change according to the unpredictable player behavior and the changing environment, reducing the burden on the developer.

The paper proceeds as follows. In Section 2 we briefly review previous work using planning for natural language generation in virtual environments. Section 3 describes the game environment where we implemented the game tutorial, and discusses the process of planning problem generation from the game environment. Section 4 presents the process used to generate an instruction according to a plan. Section 5 describes the integration of our generated instructions into the game interaction as well as the management of common ground between the player and the agent. Section 6 presents the results of our human evaluation. Section 7 concludes the paper.

2 Natural Language Generation and Planning

Natural language generation (NLG) is a subfield of Artificial Intelligence (AI) and Computational Linguistics. It is concerned with the construction of computer systems which can produce understandable texts in English or other human languages from some underlying representation of information. NLG systems combine knowledge about language and the application domain to automatically produce reports, explanations, help messages, and other kinds of texts.

The area of NLG is a new area of research, with less than a couple of decades of experience [10]. However, it is a rapidly developing field that has put forward promising techniques in the last few years [1], in particular thanks to the shared challenges proposed for comparing current NLG technologies. One of such challenges is called GIVE (Giving Instructions in Virtual Environments). GIVE [8] is particularly relevant for our work since it applies NLG techniques to the problem of generating natural language instructions in a 3D virtual world. As a result several techniques investigated by GIVE's research community are directly applicable to this work. In the GIVE task, human players try to solve a treasure hunt in a virtual 3D world that they have not seen before. The computer has a complete symbolic representation of the virtual world. The challenge for the NLG system is to generate, in real time, natural-language instructions that can guide the player to the successful completion of their task. Only the player can effect any changes in the world, by moving around, manipulating objects, etc. Figure 1 shows a screen-shot of the user's view on the 3D world. On the top of the picture, the current instruction given by the NLG system is displayed.

The NLG tasks and techniques that are relevant for GIVE and for this work are content determination through planning, situated generation of referring expressions and common ground management; we will briefly discuss them.

The task of content determination consists in deciding which information to communicate to the player such that the information is appropriate at the current point in the interaction. Such task can be implemented using the inference technique of planning [3]. As a result, the instructions are both relevant to the goal of the game and causally appropriate at the point in which they are uttered.

The situated generation of referring expressions area [10] provides algorithms for coming up with descriptions of objects so that the player can identify such objects (e.g. "the door in front of you") taking into account both static (e.g., color) and dynamic properties (e.g., visibility) of an object.



Fig. 1. The player's view during the GIVE Challenge

Finally, common ground management is the task of generating grounding acts when appropriate according to the behavior of the player. Grounding acts are utterances that, in a strict sense, do not add new information to the discourse but instead they reinforce old information. For example, if the NLG system tells the player “take the left green kit” and the player turns slightly left to face an object, the system may generate a positive grounding act such as “yes” if that was the right object, or “no” otherwise [11].

These three techniques are useful for generating hints that convey appropriate information to the player and that consider the player reaction in order to reinforce the information. In [5] the authors propose to cast all three tasks as a planning problem. Such integration is proven to be successful for small and discrete game worlds, however it does not scale to continuous game worlds in which the player can move freely (and continuously) in 3 dimensions. As a result in this paper we propose to use planning for handling only content determination while we use more traditional NLG methods for the other two tasks.

3 Finding Plans From a Game State

The design of game characters decision making is one of the most challenging tasks when designing a game; deciding what the expert game partner is to say next is, by no means, an exception to this rule. In most games, the characters decision making is either implemented in an ad-hoc way which is scripted for that particular game, or designed as a state machine which needs to be kept small because of scalability issues [9]. A believable game partner cannot get away with a decision making process that is either scripted or small. First, the agent needs to react appropriately to infinitely many player reactions which cannot be predicted and scripted. And second, it needs to reason over complex game states in a goal directed way in order to be able to give instructions that are both appropriate in the current game context and relevant to the game goal.

Our approach to implement the decision making process of the expert game partner (also called content determination in the NLG area) is to use an

off-the-shelf automated planner. The planner that we use is FastForward (FF). FF [7] can handle big planning problems (with over a million objects) and return plans of thousands of actions in seconds. Moreover, it can handle our game environment (with a couple of hundreds of objects and plans with a maximum length of 250 actions) in a few milliseconds. Automated planners have reached a maturity level in which they can be used in real time applications even if the need of re-planning is high. In our setup there is a high need for re-planning because the behavior of the player is non-deterministic and unpredictable.

The game environment for which we have implemented our game tutorial is first person shooter game (FPS). The game scenario, called Igor¹, was developed using the Irrwizard framework² and extended in C++. As in most FPS games, the player is situated in a 3D world where he can perform several actions such as walk, jump, climb stairs, shoot and pick up different items which have different effects. The goal of the game is to kill a creature that is wandering in the 3D world. The creature cannot be killed only by shooting at it because it has a self healing mechanism that needs to be turned off first by deactivating a series of power rays in a given order. The NLG agent, which guides the player through the tutorial, knows which is the right sequence of rays as well as the position of each of these rays. It is also able to recognize other items (such as poison or health) and is aware of their effect.

The agent must extract information from the environment and represent it in the standard planning language: PDDL [6]. PDDL is intended to express the physics of a domain, that is, what facts are true in the world, what actions are possible, and what the preconditions and effects of actions are. The agent represents the number of rays that the player needs to pick up and its order. Also, the agent codifies in the planning problem the game waypoints [9] and adjacencies which are used by the planner to find the nearest path between the player and the items to pick up. Moreover, the agent constantly verifies the health level of the player and enemy and updates its condition (attacking, wandering, dead etc) in the planning problem.

Once the agent has collected all the information about the current state of the game environment, it generates a planning problem and sends it to the planner. Then the planner generates a plan that the agent will use to generate the instructions. The planner takes, on average, 8.61 milliseconds to find a plan for our game planning domain (with a standard deviation of 7.46 milliseconds). Therefore, the planner can be integrated smoothly into the real time interaction of our middle sized-game.³

4 Using Plans to Decide what to Say

This section explains how the agent uses the plan obtained from the planner. The plan is composed by a series of steps that the player must follow to meet the

¹ <https://sites.google.com/site/nicolasbertoa/igor>

² <http://irrwizard.sourceforge.net/>

³ The maximum length of a plan in our game environment is 250 actions (notice that plans and plan length varies depending of the current state of the interaction).

plan goal. When the agent has a new plan, it must decide what instruction to say next. Suppose that the planner gives us the plan steps (`MoveTo (w4,w5)`, `MoveTo (w4,w5)`, `MoveTo (w4,w5)`, `MoveTo (w4,w5)`, `MoveTo (w4,w5)`, `TakeKey(blueKey,w11)`). This plan contains two types of actions, `MoveTo` and `TakeKey`. The first type of action indicates that the player has to move from one way-point to another, and the second indicates the ray that the player has to pick as well as the way-point where it is located.

These steps form a plan to pick up the ray `blueKey` as illustrated in Figure 2. Now, the agent has the problem of deciding which plan steps to verbalize, that is, it needs to do content determination based on the plan steps. Notice that the naive approach of verbalizing all plan steps would result in a very repetitive and over restrictive game partner. Therefore we made the content determination process of our agent dependent on the area which is currently accessible to the player. We define this area as the set of waypoints that are directly visible to the player or visible by turning around on his current position, we will say that these waypoints are *visible 360*.

The left drawing in Figure 2 shows that there are only three visible waypoints of the plan at 360 degrees from the position of the player (waypoints 4, 5 and 6). The others are blocked by the environment. From these waypoints, we verbalize the one that has a referring expression which uniquely identifies it. Waypoints 4 and 6 are in the middle of rooms, so, there is no referring expression that unambiguously describes them. But in the case of the waypoint 5, there is a door, so we can use a referring expression to uniquely identify the object involved. As a result, in the situation illustrated in the left drawing the instruction “go through the door in front of you” is generated.

However, what happens in the case when there is more than one object that can be uniquely identified? The right drawing Figure 2 shows this case. There are seven visible waypoints of the plan at 360 degrees from the position of the player, and three of them have an object that can be uniquely identified: 5 (has

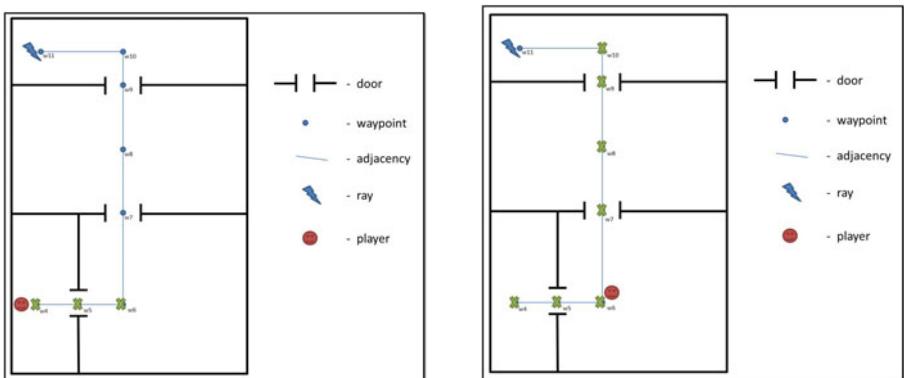


Fig. 2. Path of a sample plan with waypoints visible 360 highlighted



Fig. 3. Instructions generated using the plan and the player’s visibility

a small door), 7 (has a big door), and 9 (has some stairs). In this case, the agent generates an instruction which refers to the the furthest visible object in the path of the plan and verbalizes “see those stairs in front of you?” before saying “take them”. This strategy is used when referring to objects that are far away from the player, see Section 5 for details.

The screen-shots in Figure 3 illustrate the process we just explained. In the screen-shot we have drawn the visible waypoints and the paths between them. The arcs between waypoints illustrate the actions in the plan. In the screen-shot in the top of Figure 3 the planned actions could be verbalized as “go forward, go through the door, go forward, go forward, do you see those stairs in front of you?, take them, go forward”. As we said, we have decided to verbalize the last action in the sequence which contains a referring expression which uniquely identifies the object involved in the action. In this example, this is the case for the action “do you see those stairs in front of you?” since “those stairs” are a distinguishing referring expression (given the current position of the player) for the stairs that are further away. We say that the first actions, namely “go forward, go through the door, go forward, go forward” are *left tacit* [2] and they are expected to be inferred by the player given the causal constraints of the world (in simpler terms, in order to see “those stairs” better the player will need to approach them). The screen-shot at the bottom of Figure 3 (“Turn left”), illustrates an instruction whose goal is to make directly visible a waypoint that is visible 360.

5 Guiding an Unpredictable Player in a Changing Game

Once the agent generates the instruction, it will show that instruction to the player. Of course, the player may decide to follow the instruction or to do something else. A good game tutorial should handle both situations robustly; we discuss our strategy in Section 5.1. Furthermore, not only the player but also our environment can behave in a non deterministic way. The game partner should be able to sense and react to a changing environment in an appropriate way; Section 5.2 addresses this issue. In both of these situations grounding acts can be used to reinforce instructions that were already communicated but were not

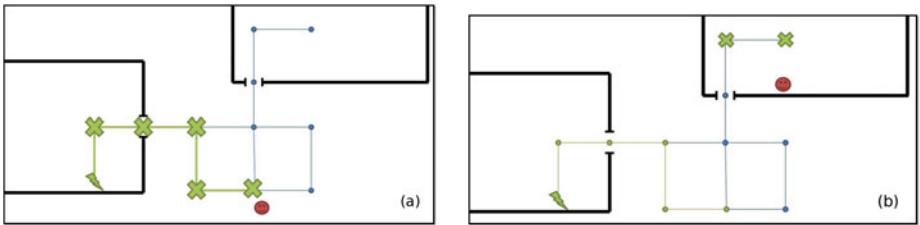


Fig. 4. Re-planning example

successfully accomplished yet; we illustrate how positive and negative grounding acts are used in Section 5.3.

5.1 Dealing with Unpredictable Behavior

Our strategy for dealing with the player’s unpredictable behavior is to find a new plan (that is, to replan) every time the player gets *too far away* from the current plan. The crucial point here is to define what it means to get too far from the current plan. Since we base our content determination procedure in the player visibility we also use the same strategy to decide when to replan. The agent will replan when all waypoints in current the plan are no longer visible 360 for the player.

Let’s illustrate our strategy by an example. Figure 4(a) shows an example scenario and the waypoints of the plan which are visible 360.

Now suppose that the player does not follow the instructions and ends up in the situation illustrated in Figure 4(b). In this new situation, re-planning is needed because none of the waypoints in the current plan (marked in green) is visible anymore from the current player position.

Summing up, the agent will re-plan when there are not visible waypoints of the plan, because it is very difficult that the agent achieves the reorientation of the player towards the goal. This strategy results effective (and not overly restrictive) while guiding the player to the game goal as shown by our evaluation results in Section 6.

5.2 Dealing with a Changing Game Environment

Suppose we have 4 ray items: blue, green, violet and red and the correct sequence to deactivate the enemy’s defense mechanism is red, blue and red. The left drawing in Figure 5 shows the plan that the agent obtained from the planner (the plan is simplified not to show move actions for presentation simplicity). What the planner cannot handle (without replanning) is the fact that, when the player picks up some object of the game, the object repositions randomly in another way-point. For example, in the right drawing in Figure 5 we can see that the obtained plan is incorrect because the red ray was repositioned at a different way-point. The planner assumes a deterministic environment, we deal with a non deterministic environment, such as our game, by means of replanning. That

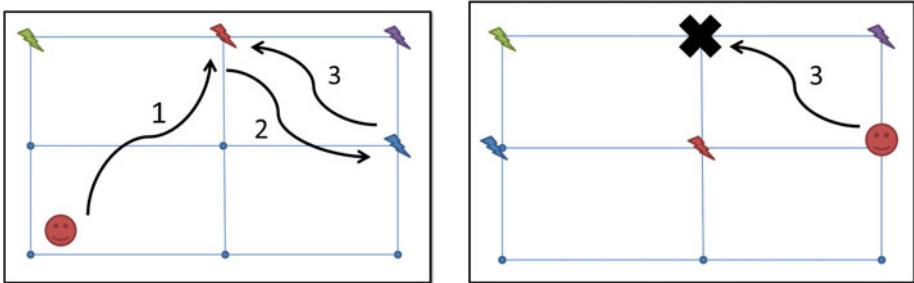


Fig. 5. Re-planning example due to non-deterministic environment



Fig. 6. Grounding acts that can be generated as a reaction to the player actions

is, when the position the environment changes in a non deterministic way (for example when a ray changes its position randomly), the agent replans.

5.3 The Importance of the Grounding Acts

Grounding acts are utterances that, in a strict sense, do not add new information to the discourse but instead they reinforce old information. Grounding acts are not required from an informational point of view of communication, however, they play an important role in order to cope with changing environments, and the unpredictable behavior of the player.

The Figure 6 shows a case of the positive grounding act “Get this ray” which follows the instruction “Turn left to see the ray” when the ray becomes visible. This utterance is a positive grounding act because, it does not add new information to the discourse. The player should be able to figure out that he should pick up this ray, otherwise, why would have the agent guided him to it? However, natural language is ambiguous and the player may not draw this conclusion so the reinforcing achieved by the grounding act is indeed useful.

The other screen-shot in the Figure 6 illustrates a case of negative grounding acts with “This is not the ray you need”, which follows the instruction “we need to find the green ray”, when the player hovers the mouse pointer on the red



Fig. 7. Multi-utterance instructions which create and use the common ground

ray. With this instruction, the agent prevents the player from re-activating the protection mechanism of the creature as a result of picking a ray in the wrong sequence. Again, in a strict sense, the negative grounding act is not necessary because the player has already been told that the next ray that is needed is green. However, the player may not remember this and may try to take the red ray which is directly in front of him.

The screen-shots in Figure 7 illustrate a typical example of common ground creation between the agent and the player. The screen-shot in the top shows the instruction “Do you see that ray in front of you?”. With this instruction the agent wants that the player to focus his attention on a particular ray. As a result it is to be expected that the player gets closer to the ray. In this new situation, the agent generates the instruction “Pick it up”. It is clear from this instruction that the agent wants the player to pick up the blue ray in front of him, but we can see that neither the ray nor its position or color are included in the instruction. The pronoun can be used because the intended ray is already in the the common ground between the agent and the player.

6 User Evaluation

In this section we describe the results of the human evaluation of our agent. For our evaluation we used objective and subjective metrics. Objective metrics were collected by logging the player behavior and the subjective metrics by asking players to complete a questionnaire. We used the same metrics that are used in the GIVE Challenge [8] to evaluate systems in terms of the effectiveness, naturalness of instructions and the engagement of the interaction. We gathered 10 volunteers for the evaluation. The demographic characteristics of the volunteers that we collected are the following: they were all male, the average age was 20 years and all of them were gamers. This subject population is the target market of the kind of game we implemented. The results that we obtained using the objective metrics are shown in Figures 8 and 9.

The Figure 8 shows the number of successful instructions, minor faults and serious faults. An instruction is considered successful if the player did exactly

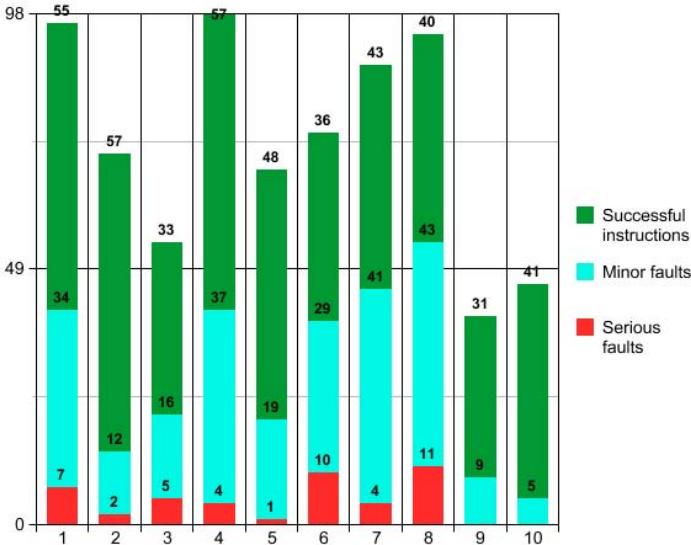


Fig. 8. Objective metrics: Successful instructions, minor and serious faults per player

what the system asked him to do, an instruction is considered a minor fault if the player deviated from the instruction without causing a replanning, and an instruction is a serious fault if the agent had to replan after uttering it. Their averages are 44.1, 28.9 and 4.4, respectively. We can see that there were few serious faults, this suggests that the agent was successful in the task of guiding the player. Also, the players deviated from 40% of the instructions and, according to the game logs, this is correlated to the presence of enemy which causes the player to not follow the plan because he is busy shooting. Finally, approximately 60% of the instructions were directly successful, the player did exactly what the agent asked him to do in more than half of the cases.

The Figure 9 shows the average of the other objective metrics we collected. The successful instructions were completed quickly, which suggests that the instructions were easy to understand and execute. Also, players were able to complete the level quickly without visiting all the map waypoints (they only visited, in average, 65.5% of the map).

The Figure 10 shows the results of the subjective metrics. From these metrics we learned that the players considered that the instructions were too repeti-

	Average	Standard deviation
Time of completeness per instruction (seconds)	28.9	17.04
Waypoints traveled	65.5	18.76
Playing time (minutes)	2.4	1.29

Fig. 9. Objective metrics: average times and distance traveled

tive (no syntactic or lexical variability). Also, we realized that the instructions where not visible enough time for the players to read them comfortably. We were pleased to observed that the statements that were related to entertainment obtained very high percentages, which suggests that the player had fun, that is the main objective of any game. Another strong point is that the instructions were considered clearly worded and were understood. Also, players perceived that most of the time the agent helped the players when they were confused.

Number	Statement	%
1	The system used words and phrases that were easy to understand	80
2	I had to re-read instructions to understand what I needed to do	70
3	The system gave me useful feedback about my progress	50
4	I was confused about what to do next	75
5	I was confused about which direction to go in	80
6	I had no difficulty with identifying the objects the system described for me	80
7	The system gave me a lot of unnecessary information	10
8	The system gave me too much information all at once	75
9	The system immediately offered help when I was in trouble	80
10	The system sent instructions too late	50
11	The system's instructions were delivered too early	10
12	The system's instruction were visible long enough for me to read them	35
13	The system's instructions were clearly worded	90
14	The system's instruction were repetitive	90
15	I really wanted to kill that creature	95
16	I lost track of time while solving the overall task	80
17	I enjoyed solving the overall task	90
18	Interacting with the system was really annoying	55
19	I would recommend this game to a friend	70
20	The system was very friendly	70
21	I felt I could trust the system's instructions	60

Fig. 10. Results of the subjective metrics

We consider that these results are encouraging, in particular because one of the main goals of our agent was to be able to participate in an engaging interaction while still providing useful instructions whenever the player needed help. We are considering different techniques to improve the agent in those characteristics in which it did not get such good results (for example, by doing corpus based generation in order to avoid being repetitive [4]).

7 Conclusions

In this paper we have presented an agent which is able to generate hints that help a player win a level in first person shooter game. Such agent uses state of the art reasoning such as planning in order to come up with the content of the

instructions and state of the art techniques from natural language generation to context-aware generate referring expressions and grounding acts.

Currently, most game tutorials make the player follow a fixed script which must contemplate as many situations as possible. When a possible reaction of the player is contemplated, the developer must add new code, increasing the development cost. Our approach is a way of automatically producing scripts that change according to the unpredictable player behavior and the changing environment, reducing the burden of the developer. Our evaluation shows that natural language techniques have a lot to contribute to the game community.

References

1. de los Ángeles Alonso-Lavernia, M., De-la-Cruz-Rivera, A.V., Sidorov, G.: Generation of Natural Language Explanations of Rules in an Expert System. In: Gelbukh, A.F. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 311–314. Springer, Heidelberg (2006)
2. Benotti, L.: Incomplete knowledge and tacit action in a dialogue game. In: Workshop on the Semantics and Pragmatics of Dialogue, Italy, pp. 17–24 (2007)
3. Benotti, L.: Implicature as an Interactive Process. Ph.D. thesis, Université Henri Poincaré, INRIA Nancy Grand Est, France, supervised by P. Blackburn. Reviewed by N. Asher and B. Geurts (2010)
4. Gandhe, S., Traum, D.: Creating spoken dialogue characters from corpora without annotations. In: Proceedings of Interspeech, Belgium (2007)
5. Garoufi, K., Koller, A.: Automated planning for situated natural language generation. In: Proceedings of the 48th ACL, Uppsala (2010)
6. Gerevini, A., Long, D.: Plan constraints and preferences in PDDL3. Tech. Rep. R.T. 2005-08-47, Università degli Studi di Brescia, Italy (2005)
7. Hoffmann, J.: Where ‘ignoring delete lists’ works: Local search topology in planning benchmarks. Journal of Artificial Intelligence Research 24, 685–758 (2005)
8. Koller, A., Striegnitz, K., Gargett, A., Byron, D., Cassell, J., Dale, R., Moore, J., Oberlander, J.: Report on the second NLG challenge on generating instructions in virtual environments (GIVE-2). In: Proceedings of the International Natural Language Generation Conference (INLG), Dublin (2010)
9. Millington, I., Funge, J.: Artificial Intelligence for Games. Morgan Kaufmann, Burlington (2009)
10. Reiter, E., Dale, R.: Building natural language generation systems. Cambridge University Press, New York (2000)
11. Traum, D.: A Computational Theory of Grounding in Natural Language Conversation. Ph.D. thesis, Computer Science Dept., U. Rochester, USA (1994); supervised by James Allen

Instance Selection in Text Classification Using the Silhouette Coefficient Measure

Debangana Dey ¹, Thamar Solorio ¹, Manuel Montes y Gomez ^{1,2},
and Hugo Jair Escalante ^{2,3}

¹ Department of Computer and Information Sciences
University of Alabama at Birmingham
1300 University Boulevard, Birmingham AL 35294.

² National Institute of Astrophysics, Optics and Electronics

³ Luis Enrique Erro No. 1, Sta. Maria Tonantzintla, Puebla, CP: 72840, Mexico.
⁴ Universidad Autonoma de Nuevo Leon
San Nicolas de los Garza, 66450, N. L. Mexico.
{deb19,solorio}@uab.edu,
{mmontesg,hugojaire}@ccn.inaoep.mx

Abstract. The paper proposes the use of the Silhouette Coefficient (SC) as a ranking measure to perform instance selection in text classification. Our selection criterion was to keep instances with mid-range SC values while removing the instances with high and low SC values. We evaluated our hypothesis across three well-known datasets and various machine learning algorithms. The results show that our method helps to achieve the best trade-off between classification accuracy and training time.

Keywords: Instance Selection, Outlier Elimination, Text Classification, Supervised Machine Learning.

1 Introduction

Text classification is the problem of assigning the most appropriate label to a new document. For the task of automated text classification based on machine learning algorithms, we need to develop a manually annotated training set in order to train the classifiers. During this developmental phase, three major problems are encountered. First, the training sets are highly vulnerable to human errors of judgment leading to mislabeling of the instances in the training corpus. Second, data ambiguity is also a reason that contributes to the confusion in the labeling of documents. For example, if a news article reports, “The President was present at the inaugural ceremony of the soccer game” it becomes difficult to categorize it as a “sports” article or as a “political” article. Such misclassified instances are noisy. Third, there are certain instances that contain redundant content and the inclusion of each one of them in the training set costs more computational time without bringing any new knowledge to the classifier. The presence of such instances causes redundancy in the training set. In automated text classification, a classifier’s accuracy is affected by the presence of the

noisy data in the training corpus. Therefore, our aim is to improve the performance of a classifier in order to achieve the most acceptable rate of accuracy by eliminating the noisy and redundant instances from the training set.

In text categorization tasks, the most common way to represent data is by building a vector space model of documents and their features. The document collection is transformed into a matrix where the documents make the rows and the features make the columns. The weights represent the relevance of the features in the documents. The set of features is generally the vocabulary of the training dataset. But when dealing with a large dataset, the processing of tens and thousands of instances and features often leads to increased computational times, difficulty in maintenance and manipulation of the data structures, hamper comprehensibility of the learned model and often leads to infeasibility in performing the task. Thus, an important step is to reduce the dimensionality of the data structures, by doing instance selection or feature selection in order to preserve the relevant information and reduce computational costs.

Through *Instance Selection*, an attempt is made to get rid of non-useful instances such that acceptable or higher accuracy is obtained at considerably reduced runtimes. The goal of instance selection can be focused on two main approaches: removal of instances that are undesirable, and selection of instances that are more informative. The first approach can be assumed to encompass the concept of *outlier elimination*, where an outlier is an object in a class that is so different from the other objects in the same class that it seems to have been generated by some other mechanism. The second approach focuses on the selection of instances that are not redundant and can contribute unique knowledge to the learning model. In instance selection, the primary intention is to keep good examples of a class that either help to represent the class firmly or distinguish it from the other classes. The concept of instance selection has taken a new shape in the form of ‘Active Learning’ in some of the recent works [1]. Active learning is a way to select useful instances to be labeled by an oracle or a human annotator from hundreds and even thousands of unlabeled instances. The goal of active learning is to reduce the cost of manually annotating the data by selecting the most informative instances. This approach is useful for machine learning problems where obtaining labeled data is very expensive.

In some of the previous works, the focus was mainly on the selection of instances in an object classification scenario [2,3,7,13]. The challenges faced in a real world text categorization problem are the numerous class labels and the huge amount of training data required to train a classifier. This in turn leads to massive data structures that demand powerful computational resources. As such, the need to find a way to reduce the size of these data structures so that they are manageable and a way to require of less number of labeled instances that are expensive to obtain, becomes inevitable. Thus, in this work, we have tried to attain the goal of Instance Selection in a text classification setting making use of the *Silhouette coefficient* measure, which is an evaluation technique for clustering algorithms [2,19]. We propose to compute the silhouette coefficient for each instance in the training set and then to eliminate the instances with high and low values. In other words, we propose eliminating the border and core instances in order to retain the instances that show a good trade-off between

descriptive and discriminative power. In order to show the performance of our proposed method, we experimented with three popular text collections in the classification domain: Reuters R8, 20 Newsgroup and the Classic4 collection. Our results show that with decreasing number of training instances, our method of retaining mid-range instances achieves a higher accuracy at an acceptable period of training time, than the accuracy obtained when high or low SC valued instances were retained.

The paper is organized as follows: in Section 2, we present some related work followed by the details of our proposed method in Section 3. We have stated some details about our datasets in Section 4. The experimental settings are outlined in Section 5 and we present our results in Section 6. We have concluded our work in the last section.

2 Related Work

In his work, Czarnowski uses the similarity coefficient to identify clusters of instances in one of his object selection approaches and uses the Silhouette Coefficient measure to evaluate the cluster quality of each of his methods [3]. A single instance was selected as a prototype of the class and used for training purposes. He uses agent-based population learning algorithms to select the desired prototypes from good quality clusters. He also uses clustering approaches for feature selection, known as *one-way clustering*. Using this method, he replaces the original feature space by clusters of features. The clusters are made based on similarity between the features. Some of the important one-way clustering methods applied in previous work are: information bottleneck [4], distributional clustering [5], and divisive clustering [6].

Lopez et al. use clustering of training instances to determine border objects in each class to build a set of prototypes to be used for classification [7]. Along with the border instances they have also included in their OSC (Object Selection by Clustering) method the centroids for each class as a core representative of the class. The intuition behind keeping border instances is to maintain the discriminative characteristics of each class, which in turn reduces the importance of core instances as they become superfluous to the class, storing redundant information [8,9]. Through their experimental evaluation of the various object selection methods like CLU method (CLUster) [10] and the DROP (Decremental Reduction Optimization Procedure) [8], they concluded that the CLU and DROP methods performed fairly well with the classifiers based on nearest neighbor rule, whereas their proposed method, OSC, performed well with the classifiers other than nearest neighbor classifiers. This was probably due to the fact that DROP and CLU are based on the nearest neighbor rule.

Another concept is that of outlier elimination, whereby the border objects are considered to be those instances that contain confusing content harmful for the classification task. The reference frame, relative to which the outliers are to be determined, can be either global or local. A local method considers only the information embedded in a document's own class, whereas a global method utilizes

information extracted from the other classes of the corpus and not only the document's own class. Shin et al. in their work approached the outlier elimination problem by using a local method [11]. They first calculated a centroid for each of the classes in the training corpus, and then imposed a threshold radius around each of those centroids. By calculating distances using the cosine similarity measure, whichever document lied at a distance greater than or equal to that threshold radius, was considered to be an outlier and was eliminated. This process was conducted for each of the classes in the corpus. Finally, this refined corpus was used to train a centroid based classifier. Their results showed that their classifier works better in the latter scenario.

In certain approaches, clusters of similar instances are formed and instances close to the clusters are considered as prototypes [12]. The algorithms chosen to compute the centroids can affect their quality and thus, have an impact on the prototype selection. Again, for large datasets, when the number of features is large, “the quality of the centroids can be poor” [3].

Lopez et al. in their work have adopted a simple, local method for prototype selection [13]. They have proposed a PSR (Prototype Selection by Relevance) method that computes the relevance weight for each instance or prototype within a class by averaging the similarity of that instance with all the other instances in its own class. A certain number of instances with the highest relevance weights have been retained. Among the retained instances, they have considered a few to be border elements. They have compared their method with other methods like DROP3, DROP5 (Different variations of DROP) and GCNN (Generalized Condensed Nearest Neighbor Rule). Their method could not outperform the other methods in case of smaller datasets but the advantage of runtime gave PSR an edge over the other methods when dealing with large datasets.

Previous works on instance selection have focused on local or global methods for the general problem of object classification. Text Classification is a relatively more challenging area of research because of the large number of classes and training sets. In our work, we propose the use of the *Silhouette Coefficient (SC)* [2] measure to rank instances for selection in the problem of Text Classification. Like Lopez et al.'s work [14] we have also tried to avoid the use of clustering techniques because clustering consumes more time to process large datasets. Moreover, our method is a global method and does not follow the nearest neighbor rule. Unlike Lopez et al.'s work, we have discarded the border and core instances to retain instances with medium relevance as depicted by their SC values. To the best of our knowledge, we are the first to evaluate this criterion of instance selection in a text classification scenario. Our intuition was that these instances represent a good trade-off between descriptive and discriminative power and, therefore, that they are the most informative for classification purposes. In addition, we have applied our method on large datasets of text documents that pose a great challenge to maintain the huge number of features generated. Our method shows enhanced accuracy and faster runtime over the dataset, reduced by our method. Taking accuracy and runtime combined as a tuple for performance evaluation we found our method to be promising.

3 Proposed Method

Our proposed method consists of ranking the instances to select those that are deemed to be more helpful in the classification task. We assume that in a corpus where instances are ranked based on a global relevance value, instances with higher relevance will be located at the *core* of any class, and those with lower relevance will lie at the *peripheral* locations of a class, since they are closer to the other classes. In such a scenario, we propose to eliminate the instances with high or low relevance values to retain what we call the *mid-ranked instances*. The motivation relies on a concept similar to term selection by transition point where mid-frequency terms are assumed to have high semantic information that can be useful in indexing the documents [25]. Based on Zipf's Law of Word Occurrences [22], refined concepts of Booth [23], and Urbizagástegui [24], transition point is a threshold frequency value computed in such a way that it can split the document vocabulary in a high or low frequency set. Pinto et. al. have used this technique to cluster documents of a corpus with narrow domain and short texts [20].

To evaluate the relevance of the training instances we propose the use of Silhouette Coefficient. In clustering tasks the *SC* width is calculated for each of the instances in the clusters in order to evaluate the cluster solution. Through this evaluation technique the following can be computed: SC value for each instance of the dataset, an average SC value for each cluster of the dataset, and an overall SC value for the whole dataset. The SC values are very helpful in denoting the cohesiveness of the instances in one cluster and the separation of instances in one cluster from those in the other clusters. Moreover, the SC measures are helpful in judging the effectiveness of a clustering algorithm.

Consider a document collection D having n classes, and let $|c_k|$ denote the number of documents from the k^{th} class of the corpus, and $dist(d_i, d_j) = 1 - \cos(d_i, d_j)$ indicate the distance between documents d_i and d_j . The SC value for the document d_i is computed by using the following formula:

$$SC_D(d_i \in c_k) = \frac{(b(d_i) - a(d_i))}{\max(b(d_i), a(d_i))} \quad (1)$$

where:

$$a(d_i \in c_k) = \frac{1}{|c_k| - 1} \sum_{\substack{\forall d_j \in c_k \\ d_j \neq d_i}} dist(d_i, d_j) \quad (2)$$

$$b(d_i \in c_k) = \min_{j \neq k} \left[\frac{1}{|c_j|} \sum_{\forall d_m \in c_j} dist(d_i, d_m) \right] \quad (3)$$

In Equation 2, we calculate the average distance of the document d_i with all the documents in its own class. This average distance becomes the *a-value* of d_i . In Equation 3, we calculate the average distance of d_i with the documents of the other classes in the corpus. We then consider the minimum of all these average values to denote the *b-value* for d_i . The SC values range from -1 to +1. Therefore, if a document has SC value near to +1, it means that for d_i , $a\text{-value} < b\text{-value}$.

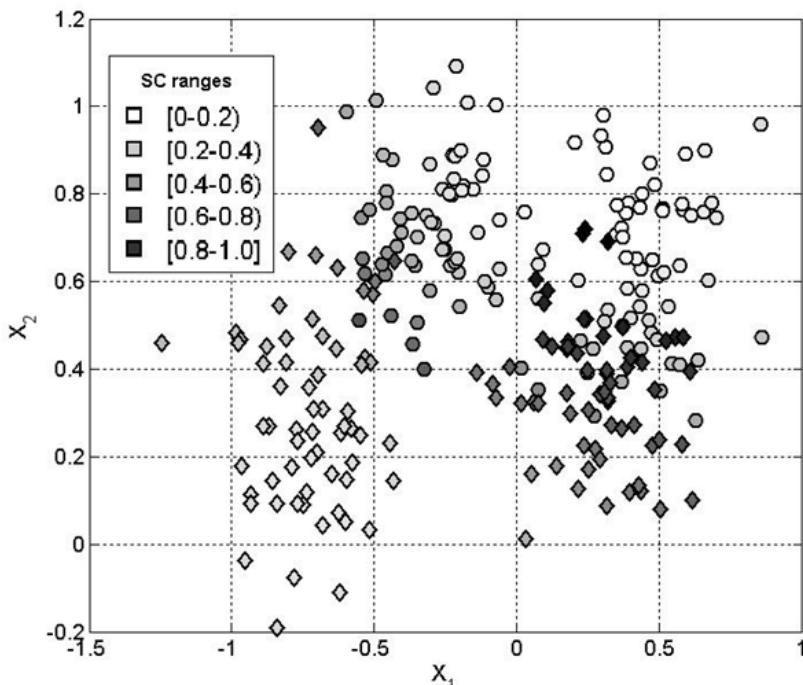


Fig. 1. Shows the SC values for a two-class syntactic data set with two attributes (X_1 and X_2), the number of classes and dimensionality were chosen for clarity

Figure 1 shows the SC value ranges for the data points in a syntactic dataset with a two-dimensional attribute space. Through the figure, we can clearly see that instances with low SC values (light gray) are those difficult to classify and can be considered noisy or potential outliers. In text categorization these instances correspond to texts that may contain information from different thematic categories and hence they can be removed. On the other hand, instances with high SC values (dark gray) are examples that can be considered easy to classify. In text categorization these instances correspond to prototypical texts of each category, although we can see that they can also be redundant and, therefore, eliminating a portion of these instances is expected to not hurt classification accuracy.

4 Datasets

The datasets used for our experiments are the following:

- Reuters-21578 R8 Collection- This dataset has 8 classes, 5,485 documents in the training set and 2,189 documents in the test set. The documents are non-uniformly distributed over the classes [17].
- Classic4 Collection- This dataset has 4 classes and 7,095 documents. We split the dataset randomly to obtain 60% documents as training set and 40% as test set [16].
- 20 Newsgroup Collection- This dataset consists of 20 classes and 18,828 documents. This dataset again was randomly split by us into 60% training and 40% testing set [18].

Each of the datasets was preprocessed for our classification task. We eliminated words from the feature set that appeared in a standard stop-list.

5 Experimental Settings

We have represented our documents using the standard vector space model where the features have formed the columns and the documents have formed the rows. The *term frequency-inverse document frequency* (tf-idf) weights have been used to fill up the term-document matrices.

For each experiment we have retained different number of text documents to determine the minimal number of training instances required for that dataset to achieve acceptable accuracies. By doing this we are also trying to dynamically determine a threshold for each dataset, such that the peripheral and the core instances can be identified. In all our experiments we start with the original training set and remove instances incrementally by steps of 10%. Thus, we end up with training sets of 90%, 80%, 70% and so on until we only have 10% on the training set. Our evaluation metric is accuracy computed as a percentage of test instances correctly classified by the algorithms.

In Table 1 we show the number of instances in each configuration for each of our data sets. The test data sets remain unchanged for all configurations; see Section 4 for a description of these sets.

Table 1. Number of instances retained for each experiment

Datasets	No. of instances									
	Original	90%	80%	70%	60%	50%	40%	30%	20%	10%
Classic4	4255	3830	3405	2978	2553	2127	1702	1277	851	426
Reuters-21578	5485	4936	4387	3839	3290	2743	2194	1646	1097	549
20 Newsgroup	11242	10118	8994	7869	6745	5621	4497	3373	2248	1124

We have tested our method using support vector machines (SVM), nearest neighbors (kNN), Naïve Bayes (NB) and random forest (RF) as the base classifiers. We have also used the Weka toolkit that implements the above classifiers [15]. As mentioned previously, our goal is to evaluate our idea of keeping instances with mid-range relevance values in terms of classification accuracy and running time. Thus, we report the difference in classification accuracy with respect to using the entire training set and running time for all three data sets.

6 Results

Table 2 shows the different experiments we performed. The first experiment (Exp 1) represents our idea of removing outliers and core instances by incrementally removing the same number of instances with the lowest and highest relevance values. For example, in a training set with only 80% of the instances, we remove 10% of the instances with the highest relevance values and 10% of the instances with the lowest values. However, for comparison purposes we also performed experiments with two different selection criteria: one where we keep instances with the lowest SC values (Exp 2), that is the border objects as in previous work, and one where we keep instances with the highest SC values (Exp 3), the core instances, again as previous work has done.

Table 2. List of the experiments performed

<i>Exp 1</i>	Keeping mid-ranged instances
<i>Exp 2</i>	Keeping instances with lowest SC values
<i>Exp 3</i>	Keeping instances with highest SC values

Table 3. The number of attributes extracted from the training set during each experiment

<i>Datasets</i>	<i>Classic4</i>			<i>Reuters-21578</i>			<i>20 Newsgroup</i>		
	<i>Methods</i>	<i>Exp 1</i>	<i>Exp 2</i>	<i>Exp 3</i>	<i>Exp 1</i>	<i>Exp 2</i>	<i>Exp 3</i>	<i>Exp 1</i>	<i>Exp 2</i>
Orig.	11867	11867	11867	11877	11877	11877	44293	44294	44293
90%	11349	11342	11323	11213	11664	10703	36209	39150	41472
80%	10776	10561	10779	10481	11352	9677	32735	34833	38709
70%	10081	9747	10149	9819	10848	8553	32735	31071	36146
60%	9395	8861	9238	9096	10268	7346	29011	27431	33028
50%	8614	7786	8177	8207	9380	5958	25465	23698	29820
40%	7723	6370	7112	7307	8410	4613	21939	20232	27194
30%	6719	5005	5945	6346	7337	3415	18394	16930	23714
20%	5375	3986	4474	5008	6052	1975	14035	13497	19738
10%	3586	2668	2664	3210	4227	726	8389	9159	13999

Tables 4 to 7 show the results for our three collections. We show the accuracy (in percentage) reached by different learning algorithms with the original training set, and

the change in accuracy after instance reduction. The tables also report the change in accuracy averaged across all the learning algorithms.

Table 4. The ‘Change in Accuracy’ observed when Exp 1 (see Table 2) was conducted

Datasets	Classic4					Reuters-21578					20 Newsgroup				
Algorithms	SMO	KNN	NB	RF	Avg.	SMO	KNN	NB	RF	Avg.	SMO	KNN	NB	RF	Avg.
Orig.	91.26	47.04	70.79	85.80	73.72	94.61	62.77	86.39	87.76	82.88	82.13	28.97	60.00	44.54	53.91
90%	0.25	-0.78	13.78	0.04	3.32	-0.55	-1.74	0.13	-0.14	-0.57	-0.67	-4.31	1.57	-1.69	-1.28
80%	0.11	-1.16	15.22	1.20	3.84	0.41	-3.15	0.64	0.18	-0.48	-3.01	-7.90	0.70	-4.37	-3.65
70%	-0.32	-1.13	22.41	1.76	5.68	0.18	-4.52	0.54	0.82	-0.74	-4.90	-11.44	0.39	-8.32	-6.07
60%	-0.56	-1.16	21.88	0.32	5.12	0.27	-4.98	1.05	0.27	-0.85	-7.09	-13.76	-1.65	-8.86	-7.84
50%	-0.81	-1.23	21.21	1.69	5.21	0.14	-6.44	1.60	-0.64	-1.34	-9.96	-16.12	-3.52	-7.93	-9.38
40%	-2.78	-1.41	20.19	-0.11	3.97	-0.82	-7.58	1.60	-1.64	-2.11	-14.35	-18.10	-5.21	-15.36	-13.25
30%	-5.32	-1.55	18.36	-4.72	1.69	-1.92	-8.41	0.77	-4.39	-3.49	-19.74	-18.98	-8.40	-19.99	-16.78
20%	-6.38	-1.73	16.63	-6.55	0.49	-6.30	-9.59	-0.37	-7.77	-6.01	-26.79	-20.94	-12.56	-21.76	-20.51
10%	-10.32	-1.76	10.64	-10.32	-2.94	-19.32	-11.42	-3.48	-13.70	-11.98	40.56	-22.55	-21.25	-30.05	-28.60

Table 5. The ‘Training Time (seconds)’ required when our method (Exp 1, see Table 2) was applied

Datasets	Classic4			Reuters-21578			20 Newsgroup		
Algorithms	SMO	NB	RF	SMO	NB	RF	SMO	NB	RF
Orig.	44.3	72.69	110.58	52.78	87.16	119.97	217.34	630.29	674.36
90%	31.99	58.97	95.08	38.13	73.52	107.92	135.13	503.8	513.48
80%	22.91	47.12	74.39	28.63	59.64	81.59	119.23	427.84	449.31
70%	18.25	42.36	65.83	21.49	48.86	66.89	96.76	306.68	305.91
60%	13.19	33.14	52	17.25	38.23	52.95	77.7	211.38	254.52
50%	9.06	23.98	41.2	11.2	28.21	38.61	46.26	154.16	161
40%	6.5	17.17	27.37	10.31	19.66	26.67	29.4	97.35	103.69
30%	3.95	11.38	16.92	5.27	12.31	16.61	14.44	59.38	61.71
20%	1.73	5.52	9.03	2.78	6.27	9.47	7.23	21.49	29.81
10%	0.5	1.72	2.95	1.02	1.88	2.84	2.59	7.88	10.06

Table 5 reports the training times recorded for each combination of machine learning algorithms and the subsets of data. The training time recorded for kNN algorithm was omitted since this algorithm does not include any training phase.

Table 6. The ‘Change in Accuracy’ observed when Exp 2 (see Table 2) was conducted

Datasets	Classic4					Reuters-21578					20 Newsgroup				
Algorithms	SMO	KNN	NB	RF	Avg.	SMO	KNN	NB	RF	Avg.	SMO	KNN	NB	RF	Avg.
Orig.	91.26	47.04	70.79	85.80	73.72	94.61	62.77	86.39	87.76	82.88	82.13	28.97	60.00	44.54	53.91
90%	0.00	-1.06	0.32	-2.54	-0.82	0.05	-1.46	-0.28	-0.96	-0.66	-0.55	-5.05	0.87	-2.60	-1.83
80%	-0.28	-1.23	-0.63	-3.10	-1.31	0.09	-2.33	-0.78	0.82	-0.55	-2.37	-8.67	-0.68	-6.61	-4.58
70%	-1.73	-1.52	-3.56	-12.33	-4.78	-0.09	-2.97	-0.87	-0.96	-1.22	-4.88	-11.76	-4.56	-13.44	-8.66
60%	-4.40	-1.73	-4.55	-18.71	-7.35	-0.91	-4.39	-1.15	-3.43	-2.47	-11.68	-14.08	-9.46	-18.75	-13.49
50%	-13.46	-1.80	-7.33	-25.97	-12.14	-7.35	-1.83	-10.60	-9.91	-7.42	-23.03	-16.16	-16.41	-23.26	-19.72
40%	-30.58	-1.90	-22.34	-35.76	-22.65	-15.44	-1.23	-24.49	-24.12	-16.32	-35.11	-19.02	-22.99	-26.80	-25.98
30%	-51.37	-1.90	-37.84	-40.03	-32.79	-29.56	-14.02	-20.83	-30.61	-23.75	-48.08	-20.27	-29.80	-30.34	-32.12
20%	-55.25	-1.90	-44.54	-40.42	-35.53	-33.90	-15.12	-23.48	-30.61	-25.78	-59.48	-21.56	-36.81	-32.15	-37.50
10%	-50.35	-1.87	-50.53	-40.59	-35.84	-29.97	-12.33	-33.90	-22.48	-24.67	-66.77	-23.04	-42.89	-35.20	-41.98

Table 7. The ‘Change in Accuracy’ observed when Exp 3 (see Table 2) was conducted

<i>Datasets</i>		<i>Classic4</i>					<i>Reuters-21578</i>					<i>20 Newsgroup</i>				
<i>Algorithms</i>		SMO	KNN	NB	RF	Avg.	SMO	KNN	NB	RF	Avg.	SMO	KNN	NB	RF	Avg.
Orig.		91.26	47.04	70.79	85.80	73.72	94.61	62.77	86.39	87.76	82.88	82.13	28.97	60.00	44.54	53.91
90%	-0.32	-0.18	14.66	0.60	3.69	1.14	-3.06	0.45	0.87	-0.15	-2.25	-3.69	-0.43	-1.18	-1.89	
80%	-0.63	-0.21	21.74	2.22	5.78	0.05	-4.39	0.36	0.05	-0.98	-4.44	-7.00	-3.77	-1.97	-4.29	
70%	-3.38	-0.46	19.56	-12.47	0.81	-1.28	-6.35	0.59	-1.74	-2.19	-6.58	-21.11	-6.75	-4.89	-9.83	
60%	-7.47	-0.81	17.58	-2.96	1.59	-6.26	-7.50	0.50	-9.91	-5.79	-9.41	-11.67	-7.30	-5.56	-8.48	
50%	-8.28	-0.95	16.95	-4.44	0.82	-31.48	-9.41	-20.06	-27.91	-22.21	-11.87	-13.94	-7.85	-6.79	-10.12	
40%	-10.50	-1.23	15.50	-6.34	-0.64	-33.44	-9.87	-24.22	-28.41	-23.99	-19.88	-13.52	-15.06	-9.55	-14.50	
30%	-14.06	-1.44	14.13	-10.15	-2.88	-33.49	-10.46	-25.27	-29.65	-24.72	-26.68	-16.56	-15.64	-13.37	-18.06	
20%	-21.35	-1.62	10.68	-16.21	-7.13	-39.06	-12.11	-25.87	-34.35	-27.85	-37.32	-19.28	-18.09	-19.30	-23.49	
10%	-35.91	-1.69	-0.28	-24.70	-15.64	-44.45	-13.11	-30.61	-37.87	-31.51	-53.72	-17.13	-26.50	-25.12	-30.62	

The results on these tables show a consistent trend, removing instances with highest and lowest relevance values yields the best trade-off between accuracy and run time. For instance, see the results for keeping only 50% of the training instances. For all three collections the smallest loss in accuracy is reached by our method, and for some learning algorithms we even gain accuracy, see results for NB in Table 4. This trend also hints at the fact that instance-based learning algorithms benefit the most from instance reduction.

Another consistent and relevant trend in our results is that if we do an extreme instance reduction, by keeping only 10% of the training instances, the loss in accuracy overall will be less when using our approach than any of the other two. By comparing the results reported in Table 4 and 7, it is clear that for a limited number of instances, such as just 10% of the total corpus, selection of mid-ranged instances for training is a better idea than instances with either high and low relevance. This is encouraging and a very practical finding since if computational resources are very expensive, and lower accuracy rates are acceptable, our approach will yield the best trade-off between efficiency and effectiveness. Our results show that at times it is also possible to achieve higher accuracy than the baseline, when the mid-ranged instances are retained, whereas, if we remove these instances, the accuracy values continue to decrease consistently with a shrinking training set, for all the learning algorithms. Therefore, it is evident that the mid-ranged instances carry important information concerning the class description and cannot be ignored.

These results also show some effects on the differences among the corpora. For instance, in the Classic4 collection, if we remove only 20% of the training instances, we obtain slightly better results by removing the ones with lowest relevance values. This leads us to believe that Classic4 has more outliers in the training set than the other two collections. The other two collections still show better results when using our method.

Lastly, it is very interesting to see that a secondary effect of our instance reduction approach is feature reduction. All tables show in the last row the resulting number of features. As can be seen in the last row of the tables, we reach a considerable reduction in the number of features. This is an effect particular to instance selection in text classification, because after removing instances, a lot of the words present in

those documents are not present in the other documents and can be removed from the feature vectors. We have not seen a related discussion about this in any previous work. It would be interesting to explore further how this method compares to other approaches on feature selection. This however, is beyond the scope of this work and we will leave it for future work.

7 Conclusions

In this paper we have proposed the use of the Silhouette Coefficient measure for the purpose of determining the most effective set of training instances in text classification. This measure is typically used to evaluate the quality of clusters generated after the application of clustering algorithms on the data. We have calculated the SC measure for each of the instances in the training set. We then rank the instances according to their SC measures; those with the lowest and highest SC values were eliminated, assuming them to be noisy or to contain redundant information that might not aid us in our classification task. We found that instances in the middle range of SC values were capable of retaining more descriptive information about their respective classes than instances with extreme SC values.

We have taken accuracy values and training time as a tuple for performance evaluation. Elimination of the instances in the training set led to a drop in accuracy from the original training set in most of the cases. But the consistent tradeoff between accuracy measures and training time was commendable when our method was applied. There were also cases when we observed a higher accuracy when instances with medium ranks were retained in the training sets. Unlike the peripheral or core instances, the medium ranked instances proved to provide more information that could help in describing their classes and also discriminate them from each other. We have evaluated our results by carrying out experiments where only peripheral instances were eliminated or only core instances were removed. Training with only the peripheral instances or only the core instances led to a higher drop in the accuracy values than when medium-ranked instances were used to train the model. The experimental results for all the three datasets showed the same trend of maintaining an acceptable accuracy coupled with considerable reduction of time.

The Silhouette Coefficient measure seemed to be an interesting parameter for evaluation and we would like to delve deeper into its analysis and try to find out what other information can be extracted with the use of this measure. We would like to test and compare our method using different datasets having narrow domain unlike the broad domain datasets used in this work. Also, it will be interesting to investigate the efficacy of our method in the scenario of short text classification using the abridged versions of the datasets used for this work. Lastly, we would like to propose other measures to determine relevant instances for training classifiers in a text classification setting.

Acknowledgments. This work was supported in part by the CONACYT-Mexico (project no. 134186) and by the European Commission as part of the WIQ-EI project (project no. 269180) within the FP7 People Programme.

References

1. Settles, B.: Active Learning Literature Survey, Computer Sciences Technical Report 1648. University of Wisconsin–Madison (2009)
2. Hubert, M., Struyf, A., Rousseeuw, P.: Clustering in an Object-Oriented Environment. *Journal of Statistical Software* 1(4) (1996)
3. Czarnowski, I.: Cluster-based instance selection for machine classification. *Knowledge and Information Systems* (2011)
4. Slonim, N., Tishby, N.: Agglomerative Information Bottleneck. In: *Advances in Neural Information Processing systems (NIPS-12)* (1999)
5. Baker, L.D., McCallum, A.K.: Distributional clustering of words for text classification. In: *Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR 1998*, pp. 96–103 (1998)
6. Dhillon, I.S., Mallela, S., Kumar, R.: A Divisive Information-Theoretic Feature Clustering Algorithm for Text Classification. *Journal of Machine Learning Research* 3, 1265–1287 (2003)
7. Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: Object selection based on clustering and border objects. In: Kurzynski, M., et al. (eds.) *Computer Recognition Systems 2*, Wroclaw, Poland. ASC, vol. 45, pp. 27–34 (2007)
8. Martinez, T.R.: Reduction Techniques for Instance-Based Learning Algorithms. *Machine Learning* 286 (2000)
9. Brighton, H.: Advances in Instance Selection for Instance-Based Learning. *Algorithms. Knowledge Creation Diffusion Utilization*, 153–172 (2002)
10. Lumini, A., Nanni, L.: A clustering method for automatic biometric template selection. *Pattern Recognition* 39, 495–497 (2006)
11. Shin, K., Abraham, A., Han, S.-Y.: Enhanced Centroid-Based Classification Technique by Filtering Outliers. In: Sojka, P., Kopeček, I., Pala, K. (eds.) *TSD 2006. LNCS (LNAI)*, vol. 4188, pp. 159–163. Springer, Heidelberg (2006)
12. Wilson, D.R., Martinez, T.R.: An Integrated Instance-Based Learning Algorithm. *Computational Intelligence* 16, 1–28 (2000)
13. Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: Prototype selection via prototype relevance. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) *CIARP 2008. LNCS*, vol. 5197, pp. 153–160. Springer, Heidelberg (2008)
14. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: *Proc. of 9th International Conference on Machine Learning*, pp. 249–256 (1992)
15. Hall, M., Frank, E., Holmes, G., Bernhard, P., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. *SIGKDD Explorations* 11(1) (2009)
16. Classic4 Dataset, <ftp://ftp.cs.cornell.edu/pub/smartz/>
17. Reuters R8 Dataset,
<http://www.daviddlewis.com/resources/testcollections/reuters21578>
18. 20 Newsgroup Dataset,
http://people.csail.mit.edu/jrennie/public_html/20Newsgroup/
19. Rousseeuw, P.J.: Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Computational and Applied Mathematics* 20, 53–65 (1987), doi:10.1016/0377-0427(87)90125-7
20. Pinto, D., Rosso, P., Jiménez-Salazar, H.: A Self-enriching Methodology for Clustering Narrow Domain Short Texts. *Computer Journal* 54(7), 1148–1165 (2011)

21. Pinto, D., Rosso, P., Jiménez-Salazar, H.: A Self-enriching Methodology for Clustering Narrow Domain Short Texts. *Computer Journal* 54(7), 1148–1165 (2011)
22. Zipf, G.K.: Human Behavior and the Principle of Last-Effort. Addison-Wesley, Cambridge (1949)
23. Booth, A.: A Law of Ocurrences for Words of Low Frequency. *Information and Control* (1967)
24. Urbizagástegui, R.: Las posibilidades de la Ley de Zipf en la indización automática, Research report of the California Riverside University (1999)
25. Pinto, D., Jiménez-Salazar, H., Rosso, P.: Clustering Abstracts of Scientific Texts Using the Transition Point Technique. In: Gelbukh, A.F. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 536–546. Springer, Heidelberg (2006)

Age-Related Temporal Phrases in Spanish and French

Sofía N. Galicia-Haro¹ and Alexander Gelbukh²

¹ Faculty of Sciences, UNAM,
Mexico City, Mexico

sng@fciencias.unam.mx

² Center for Computing Research, National Polytechnic Institute,
Mexico City, Mexico
www.Gelbukh.com

Abstract. This paper reports research on temporal expressions. The analyzed phrases include a common temporal expression for a period of years reinforced by an adverb of time. We found that some of those phrases are age-related expressions. We analyzed samples obtained from the Internet for Spanish and French to determine appropriate annotations for marking up text and possible translations. We present the results for a group of selected classes.

Keywords: temporal expressions, people's age, multilingual comparison.

1 Introduction

Some words or whole sequences of words in a text are temporal expressions: for example, *yesterday*, *Monday 12*, *two months*, *about a year and a half*, each refers to a certain period of time. Such words or sequences of words mainly share a noun or an adverb of time: *yesterday*, *month*, *year*. This causes a problem in automatically deciding whether a word or a sequence is a temporal expression. It is an important part of many natural language processing applications, such as question answering, machine translation, information retrieval, information extraction, text mining, etc., where robust handling of temporal expressions is necessary.

Automatic recognition of expressions of time was introduced in the Named Entity Recognition task of the Message Understanding Conferences¹, where temporal entities were tagged as “TIMEX.” Since then, researchers have been developing temporal annotation schemes; for example, [4] and [15] for English, [3] for French, and [16] for Spanish.

The authors in [4] produced a guideline intended to support a variety of applications in the performance of some useful tasks. As the authors pointed out, the guideline was not intended to represent all the varieties of temporal information conveyed in natural language communication. They were interested in temporal expressions that reference calendar dates, times of day, or durations. They considered

¹ <http://timexportal.wikidot.com/timexmuc6>

lexical triggers to identify temporal expressions. A lexical trigger is a word or numeric expression whose meaning conveys a temporal unit or concept. To be a trigger, the referent must be able to be oriented on a timeline, or at least oriented with relation to a time (past, present, future).

In this work, we analyzed other different Spanish temporal expressions that do not fulfill the described trigger characteristics and were not considered in those annotation guidelines. These phrases are recognized by an initial adverb: for example, *around*, *still*; they end with the noun of time, e.g. *year*, and they describe a person's age. For example: *aún a sus 50 años* "although he is 50 years old", *ahora a mis 23 años* "now I am 23 years old", *alrededor de los 55 años* "around 55 years old." These phrases are very interesting since they reinforce the meaning of time.

Automatic recognition of a person's age should be useful in machine translation tasks, among other tasks in natural language processing. Due to the constructions similarity in Spanish and French we supposed that these phrases of our interest would be similar in both languages. However we found that the French translation² "*encore à ses 38 années*" for the phrase *aún a sus 38 años* is not common in French. An interesting approach can be related to construction of a context free grammar that describes this phenomenon like it was done in [1, 2].

In this article, we present a corpus-based analysis carried out to compare such Spanish temporal expressions with age-related temporal phrases in French with the objective of determining appropriate annotations for marking up text and possible translations relating them. In section 2, we present the characteristics of the Spanish phrases and the method we applied to obtain the materials for the comparison. In section 3, we describe the French phrases obtained with the same method. In section 4, we present the comparison of such phrases. Section 5 concludes.

2 Age-Related Temporal Expressions in Spanish

Usually people's age is described by Spanish temporal expressions including the time noun *años* "years." They can be recognized in the following ways:

- with the string *de edad* (lit. "of age") after the word *años*
Example: *un vendedor de 38 años de edad* (a seller who is 38 years old)
- with the preposition: *de* (of) after an animated noun or person name and before the number of years, sometimes delimited by commas
Example: *la niña de 11 años* (the 11-year-old girl)
- with the strings: *la edad de* (lit. "the age of") before the number of years.
These phrases can be preceded by different prepositions: *a*, *con*, *desde*, *hasta*
Example: *falleció ayer a la edad de 95 años* (died yesterday at 95 years old)

There are, however, other temporal expressions that describe people's age: for example, *aún a sus 65 años*, lit "still at his 65 years", *de alrededor de 20 años*, lit. "of about 20 years." These temporal phrases denote a point in the timeline of a person; it could be a point in the timeline of the events related in the sentence or a point in a tangential timeline.

² <http://www.online-translator.com/Default.aspx/Text>

In [7] we analyze the context for Spanish temporal phrases that begin with an adverb of time (AdvT) and end with a noun of time (TimeN) expressing a person's age. We can observe the relation between the groups of words in the following examples:

1. *A sus 30 años Juan se comporta como niño*
2. *Aún a sus 30 años Juan se comporta como niño*
3. *Hoy a sus 30 años Juan se comporta como niño*

The sentences describe the same main fact: *John, who is 30 years old, behaves like a child*, but they tell us something else when we introduce a modifier (*aún* “still,” *hoy* “today”) in each one: they argue for different conclusions.

- Even at 30 years old, John behaves like a child \Rightarrow in spite of his age he behaves as if he were a child
- Today, at 30 years old, John behaves like a child \Rightarrow today he behaves like a child.

The adverbs “even” and “today” make such conclusions obligatory and reinforce the meaning of time in different forms. Both adverbs are related to time duration; one strict reading refers to 24 hours and the other to a longer period of time, but they also imply a direct judgment on the perception of the speaker, on the behavior of the subject or both.

2.1 Material Acquisition

In [7] we present a corpus-based analysis carried out to determine the context of such Spanish temporal expressions for their automatic determination. Such a method allows the manual selection of examples representing what we consider a class: a different combination of an adverb and a preposition before the number of years and then the retrieval of web examples for that class. The method consists of two steps.

MA First Step: we used one text collection compiled from a Mexican newspaper that is published daily on the Web in almost its entirety. The texts correspond to diverse sections, economy, politics, culture, sport, etc., from 1998 to 2002 [6].

We wrote a program to extract the sentences matching the following pattern:

AdvT–something–TimeN

where:

something – corresponds to a sequence of up to six words³ without punctuation marks, verbs or conjunctions

TimeN – corresponds to *año*, *años* “year, years”

AdvT – adverbs of time, a collection of 51 elements from a dictionary.⁴

From 27,054 sentences, we manually selected one arbitrary example representing a class: a different combination of an adverb and a preposition before the number of years, the five resulting classes correspond to *aún a*, *aún con*, *actualmente de*, *alrededor de*, *ahora de*.

³ A larger quantity of words does not guarantee any relation between the AdvT and the TimeN.

⁴ DRAE, Real Academia Española. (1995): *Diccionario de la Real Academia Española*, 21 edición (CD-ROM), Espasa, Calpe.

MA Second Step: since our newspaper text collection contains a subset of all possible temporal phrases expressing the age of people, we analyzed a method to obtain a more representative group of phrases and we chose to look for examples on

SEARCH(C)

For each phrase of type ADV-* -NounT or string-* -NounT in C

(1) Obtain 100 examples from the Internet

- (1.1) D = {examples excepting such instances where * includes verbs or
- (1.2) Print D

(2) Classify them according to such words retrieved by *

(3) For each group of phrases sharing words retrieved by *, assign a class D_i

- (3.1) F = class D_i
- (3.2) SEARCH(F)

UNTIL no new elements are obtained

Fig. 1. Algorithm to obtain variants of temporal expressions

the Internet. This option allowed us to find phrases generated by native speakers more quickly, including the commoner collocations. Nevertheless, we know that searching the Internet has drawbacks but we decided to do so on the basis that we do not know how the results are classified [12].

The main idea of obtaining more examples from the Internet is based on obtaining a few examples from the newspaper texts (corresponding to the five classes above mentioned), simplifying them (eliminating determinants, adjectives, etc.) and searching for variants by including Google's asterisk facility [8]. The whole procedure is shown in Figure 1. For example: for the phrase *aún con sus jóvenes 48 años* the string when simplified becomes “*aún con año*” and the search is “*aún con * años*” using the Google search engine tool limited to the Spanish language where the asterisk substitutes for the eliminated words. Google returns hits where there is a string of words initiated by “*aún con*” and then a sequence of words, ending with “*años*.”

The example for the whole procedure in Figure 1 is presented as follows:

SEARCH(“*aún con * años*”)

Step (1) 100 examples

... y el bachillerato en Lleida, *aún con dieciséis años* entró a trabajar de chico ...

aun con tantos años sigo siendo el mismo de siempre...

... porque se dio cuenta que *aún con tantos años* encima son capaces de ...

Un partido, *aún con pocos años* de actuación, inspirado en la Gran ...

... y menos *aún con Nuestros años* felices (1996) o Tarde (1998)

...

Step (2) $D = \{ \text{aún con dieciséis años}, \text{aún con pocos años}, \text{aun con tantos años}, \text{aún con Nuestros años}, \dots \}$

Step (3) For each one of the classes a new process is initiated

SEARCH(“*aún con tantos * años*”)

SEARCH(“*aún con pocos * años*”)

...

The process is repeated several times until no new repeated phrases are obtained, determining the sequences of words that appear with higher frequency. We note that in addition some phrases not corresponding to the temporal phrases we are interested in are picked up. These phrases were eliminated in the manual identification at the end of each cycle to reduce the quantity of Google searches.

After this compilation of examples, we manually selected 18 classes that appear in the first column of Table 1, where NUM treats numbers represented by digits or letters. We found that some of the 18 classes obtained from the Internet seem to preserve their meaning independently of the context and others require some form of words in context to denote the age of a person. The quantity of pages automatically obtained was limited to 50, i.e. to obtain 500 snippets.

Table 1. Selected phrases for age-related temporal phrases in Spanish

Type of phrase	# examples	% age-related
aún a tus NUM años	7	100
aún con mis pocos NUM años	1	100
aún con mis cortos NUM años	2	100
aún con sus escasos NUM años	4	100
aún con tus casi NUM años	1	100
ahora a mis NUM años	182	99
aún a sus NUM años	293	96
aún hoy a sus NUM años	38	92
aún con sus NUM años	109	86
ahora de NUM años	352	86
alrededor de los NUM años	355	84
actualmente con NUM años	270	80
ahora con casi NUM años	118	67
aún con sus casi NUM años	7	57
ahora con más de NUM años	90	46
actualmente de NUM años	28	36
ahora a los NUM años	132	36
actualmente de unos NUM años	16	19

The overall results are presented in Table 1. The second column shows the number of examples obtained, after the elimination of phrases where there is no relation between the AdvT and the TimeN. Since the examples were automatically obtained from the snippet, some of them were not considered because of the lack of text when the sentences were split and context omitted around the searched phrase. Fewer than 10% are errors because of the short snippet. Column 3 shows the results after manual syntactic and semantic analysis of the context.

Many studies focused in having a corpus that models the whole language. However, for inducing information for annotation and translation of the phrases we are interested in we collected just a particular subset of language, the one that corresponds to them. Thus, the research we report about here refers to aspects related to the collection that has been skewed by design.

3 French

The analysis described in this section was conducted on 10,234 sentences obtained from the Europarl corpus [14] that we call EuropAns from now on. The sentences were retrieved in response to a query for the noun *ans* “years,” since people’s age is described by French temporal expressions including that noun. We also use 1,035 sentences obtained from CNRTL [5].

Table 2. Selected phrases for age-related temporal phrases in French

Type of phrase	# examples	% age-related
âgé(e,s,és) d'autour de NUM ans	2	100
encore maintenant à NUM ans	2	100
aujourd'hui encore à NUM ans	1	100
maintenant âgé NUM ans	1	100
maintenant âgé de NUM ans	5	100
maintenant à l'âge NUM ans	1	100
âgé(e) de maintenant NUM ans	3	100
environ âgé de NUM ans	1	100
âgé(s,e,es) actuellement de NUM ans	37	92
actuellement âgé(e,es) de NUM ans	12	92
âgé(e,s,es) d'environ de NUM ans	37	84
âgé(e,s,es) d'environ NUM ans	24	80
maintenant à presque NUM ans	4	75
actuellement NUM ans	175	74
âgé(e,s,es) environ de NUM ans	38	74
maintenant à NUM ans	166	68
autour de mes NUM ans	3	67
autour de NUM ans	430	56
actuellement de NUM ans	236	30
environ de NUM ans	158	30
encore à NUM ans	163	23
environ NUM ans	285	19
maintenant NUM ans	523	14
encore NUM ans	412	8

The most common French phrases expressing age can be recognized in the following ways:

- with the word “*de*” before the number of years. E.g. *sa soeur de 7 ans* (his sister of seven years old)
- a number and the word “*ans*” after the person’s name, delimited by commas or parentheses. E.g. *Aretha Franklin, 31 ans, Paul Dubois (88 ans)*,
- with a string “*âgé(e) de*” (“age_{ADJECTIVE}” in English) before the number of years. E.g. *jeunes musiciens, âgés de 16 ans* (young musicians of 16 years old)
- with a string “*âge de*” (“age_{NOUN} of” in English) before the number of years. E.g. *mourut à l'âge de 27 ans* (died at the age of 27 years)

To analyze the temporal phrases expressing age similar to the Spanish phrases detailed in section2 we applied the MA First Step to the EuropAns. We obtained 257 sentences

matching the AdvT–something–TimeN pattern where Adv corresponds to 57 elements. From them we manually selected five classes to process the MA Second Step by launching the following queries: *autour * ans*, *actuellement * ans*, *encore * ans*, *environ * ans*, *maintenant * ans*.

We applied the MA Second Step to access the Google search engine tuned to the French language. The quantity of pages automatically obtained was also limited to 500 snippets. The results obtained from the Internet produced the 3,717 examples that are detailed in Table 2. The columns correspond to the same parameters as the previous table.

4 Comparison

According to [11], in many instances, statistically equivalent constructions are not semanto-syntactically equivalent. Considering quantitative data, a pair of phrases of two different languages could be non-equivalent on at least two counts: intralinguistic and interlinguistic (contrastive). The intralinguistic results can also be obtained cross-linguistically and more directly by looking at various translations of a given construction into another language. Statistically comparisons can be conducted both in texts which are attested as translations and on texts which are not translations but are comparable on account of being written on a similar topic, by similarly qualified authors using similar registers, etc. For this reasons although our materials do not come from translations they are comparable and they possess qualities in common, i.e. *tertium comparationis*.

We note that although Spanish and French exhibit a substantial degree of similar realization of age phrases, there are some divergences. Some of them correspond to prepositions and incorporation of determinants and adjectives, which are not present in French. The French expressions insert the modifier related to age (*âgée*, *âgé*, *âgées*, *âgés*) before the duration (NUM *ans*). Thus, to compare the age phrases, first we enumerate the elements to be compared: 1) adverbs, 2) surface structure of the phrase between adverb and noun *years*, and 3) adjective placement. For this comparison we divide the results in four groups corresponding to the adverbs: *ahora/maintenant* ‘now’, *alrededor/autour-environs* ‘around’, *actualmente/actuellement* ‘at present’, *aún/encore* ‘still’.

We match the above described groups in Spanish and French considering first the adverb, then the age meaning and finally the percentage of phrases with age-related meaning. These matched groups are shown in tables 3 to 6. We also consider as stated by [11] that two linguistic items across languages are statistically equivalent if in comparison with other synonymous constructions, they have maximally similar frequency of occurrence in the relevant texts. In general, we compare classifying examples as identical, different in some respects, and no equivalent.

The group associated to ‘now’ adverb (see Table 3) contains one case where the structure of the phrase between the adverb and the quantity of years is a prepositional phrase with a possessive adjective related to the person whose age is described (*mis* ‘my’). It is a productive group to describe age of persons. Structurally it is quite different of French phrases but it is quite related in meaning to the *âgé* inclusion. However the most productive group for age meaning in French is *maintenant à* NUM *ans*.

Table 3. Group of age's person phrases initiated by the adverb *ahora/maintenant*

Type of phrase	# ex/ % age	Type of phrase	# ex/ % age
ahora a mis NUM años	182/99		
ahora de NUM años	352/86	maintenant âgé NUM ans	1/100
		maintenant âgé de NUM ans	5/100
		âgé(e) de maintenant NUM ans	3/100
ahora con casi NUM años	118/67	maintenant à presque NUM ans	4/75
ahora con más de NUM años	90/46		
ahora a los NUM años	132/36	maintenant à NUM ans	166/68
		maintenant à l'âge NUM ans	1/100
		maintenant NUM ans	523/14

There is one case that we can compare as identical in structure and meaning but using a different preposition, it is showed in the third row of table 3 (*ahora con casi NUM años*). The cases we classify as different in some respects correspond to second and fifth rows. The case *ahora de NUM años* have a *de* prepositional phrase in its counterpart's French phrase but governed by the adjective and with less percentage of examples in French phrases. The fifth case differs in the article for the years in Spanish and for the noun *âge* in French. The last row shows a case with no equivalent in Spanish.

Table 4. Group of age's person phrases initiated by the adverb *alrededor/autour-environ*

Type of phrase	# ex/ % age	Type of phrase	# ex/ % age
alrededor de los NUM años	355/84	environ âgé de NUM ans	1/100
		âgé(e,s,és) d'autour de NUM ans	2/100
		âgé(e,s,es) d'environ de NUM ans	37/84
		âgé(e,s,es) d'environ NUM ans	24/80
		âgé(e,s,es) environ de NUM ans	38/74
		autour de NUM ans	430/56
		environ de NUM ans	158/30
		autour de mes NUM ans	3/67
		environ NUM ans	285/19

The group of the phrases initiated by the ‘around’ adverb (see Table 4) is the only French group including one case where the structure of the phrase between the adverb and the quantity of years is a prepositional phrase with a possessive adjective related to the person whose age is described (*autour de mes NUM ans*). This structure is common in Spanish and we describe it in the last group. This French case has no equivalent in this group. The *environ NUM ans* case has no Spanish equivalent in this group in addition of being the least productive.

The last case is classified as without equivalent and corresponds to the least productive of age's person phrases in this group. The first case is classified as different in some respects. These phrases have a *de* prepositional phrase in both languages governed by the adjective in four French counterparts. The other three phrases have the prepositional phrase governed by the adverb, being more similar to the Spanish phrase but less productive of age's person phrases.

Table 5. Group of age's person phrases initiated by the adverb *actualmente/actuellement*

Type of phrase	# ex/ % age	Type of phrase	# ex/ % age
actualmente con NUM años	270/80	âgé(s,e,es) actuellement de NUM ans	37/92
		actuellement âgé(e,es) de NUM ans	12/92
		actuellement NUM ans	175/74
actualmente de NUM años	28/36	actuellement de NUM ans	236/30
actualmente de unos NUM años	16/19		

The third group corresponds to ‘at present’ adverb (see Table 5). It contains one identical case: *actualmente de NUM años*, one no equivalent for Spanish: *actualmente de unos NUM años* having an adjective for the period of years, and one no equivalent for French: *actuellement NUM ans* without a prepositional phrase.

Table 6. Group of age's person phrases initiated by the adverb *aún/encore*

Type of phrase	# ex/ % age	Type of phrase	# ex/ % age
aún a tus NUM años	7/100		
aún con mis pocos NUM años	1/100		
aún con mis cortos NUM años	2/100		
aún con sus escasos NUM años	4/100		
aún con tus casi NUM años	1/100		
aún a sus NUM años	293/96		
aún hoy a sus NUM años	38/92		
aún con sus NUM años	109/86		
aún con sus casi NUM años	7/57		
		encore maintenant à NUM ans	2/100
		encore à NUM ans	163/23
		encore NUM ans	412/8
		aujourd'hui encore à NUM ans	1/100

The first row shows the case we classify as different in some respects. The case *actualmente con NUM años* has a prepositional phrase, the preposition *con* in Spanish and *de* in French. But we know that the selection of a functional preposition is determined by the governor, and the preposition is typically not carrying much semantic information. Here, the prepositional phrase is governed by the adjective in the first case and governed by the adverb in the second French case.

The last group associated to ‘still’ adverb (see Table 6) is the one that no contain identical items in structure. All the Spanish cases present a similar structure of the phrase between the adverb and the quantity of years being a prepositional phrase with a possessive adjective related to the person whose age is described and half of the cases additionally include a modifier of the number of years (*aún con mis pocos NUM años*, *aún con mis cortos NUM años*, *aún con sus escasos NUM años*, *aún con tus casi NUM años*, *aún con sus casi NUM años*.) These last phrases belong to the colloquial language. The French phrases of this group have the minor percentage of persons' age meaning and only a case includes a prepositional phrase.

4.1 Annotation

Annotation of temporal expressions in Spanish developed in [16] considers those phrases expressing day times, dates and duration (TIMEX3). Annotation for temporal expressions in French developed in [1] includes in addition the annotation of events and tags for more information. Generally, temporal expressions are annotated as entire constituents, typically noun phrases (e.g. 38 years). For example:

3 heures de vol (3 hours flight), it is annotated in [1] as follows:

<TIMEX3 tid="t1" type="DURATION" value="P3H">3 heures</TIMEX3>
de<EVENT>vol</EVENT>, where. ‘t#’ is the unique identifier for a marked temporal expression and P10S means ‘period’ 10 ‘seconds’.

For more complex temporal expressions the annotation is divided, for example:

Ahora hace casi veinte años de la caída del muro de Berlín. ‘Now almost twenty years ago of the fall of the Berlin Wall’ [Published: September 24, 2008], it is annotated in [12] as follows:

- a. extent: *ahora* type:DATE value:PRESENT REF mod:--
- b. extent: *hace casi veinte años* type:DURATION value:P20Y mod:LESS THAN
- c. extent: [empty tag] type:DATE value:1988 mod:AFTER

Where: extent c is representing the date of the day the sentence was written.

Owing to the insertion of phrasal prenominal modifiers in the temporal phrases discussed above, temporal expressions denoting people’s age in Spanish and French should be annotated following the directions in [3]. We propose to annotate the phrases we are analyzing here with the TLINK tag. TLINK is a temporal link that could represent the relation between two temporal elements: TIMEX3-TIMEX3. We propose to include the type AGE in addition of DATE, TIME, DURATION and SET. The phrases we consider for this annotation are those more productive with appropriate contexts extracted from the examples. We include the prepositional phrase in the extent of the temporal phrase since its value could be a mod attribute value. For example, we annotate the phrases *ahora a mis NUM años* and *actuellement âgé de NUM ans* in such proposed form:

ahora a mis NUM años

```
<TIMEX3 tid="t1" type="DATE" value="PRESENT_REF" >ahora</TIMEX3> <TIMEX3 tid="t2"
type="AGE" value="PnumY" temporalFunction="TRUE" anchorTimeID="t1"> a mis NUM años
</TIMEX3>
```

<TLINK timeID="t1" relatedToTime="t2" relType="SIMULTANEOUS"/>

actuellement âgé de NUM ans

```
<TIMEX3 tid="t1" type="DATE" value="PRESENT_REF" >maintenant</TIMEX3> <TIMEX3
tid="t2" type="AGE" value="PnumY" temporalFunction="TRUE" anchorTimeID="t1"> âgé de NUM ans
</TIMEX3>
```

<TLINK timeID="t1" relatedToTime="t2" relType="SIMULTANEOUS"/>

4.2 Translation

The results classified as ‘different in some respects’ are the cases that we analyze to be translated. Such Spanish temporal expressions form a contiguous sequence given

an appropriate context, and should be translated into an entire sequence as a multi-word unit. We propose their translation by specific templates as a preprocessing step applied before any other type of processing. Preprocessing is considered in different works, for example in [12] the application of a rule-based language-independent chunker that contains distituency rules, in [13] specifying time and date phrases.

As considered in [9], the meaning of a lexical unit must be preserved in the target language, even though it may take a different syntactic form in the source and target languages. When translating a functional preposition, the identity of the source language preposition is thereby of less importance. In our cases, the prepositions are different in the source and target languages and the binding of the prepositional phrases have a variant in French, attached either to the adverb or the adjective *âgé*.

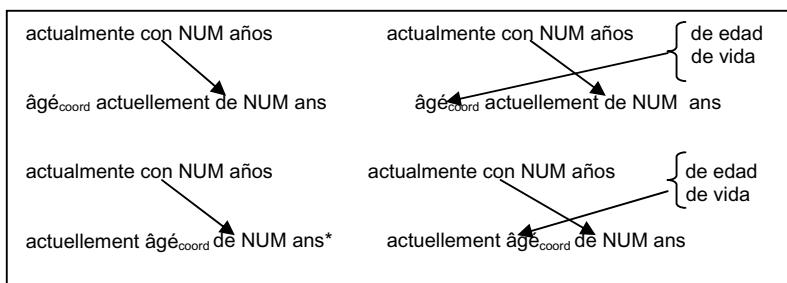


Fig. 2. Translation templates for ‘at present’ adverb

5 Conclusions

The variety in the structure of temporal expressions requires analysis of different combinations of classes of words. We analyzed temporal expressions including the noun of time *year* that are modified by an adverb of time and the whole phrase expressing a person’s age. These phrases are interesting since adverbs make some inferences obligatory and reinforce the meaning of time in different forms. Besides the time duration involved, they imply a direct judgment on the perception of the speaker, on the subject or both.

We first presented the method to enrich the classes of temporal phrases when only a few examples are compiled. To obtain a more representative sample we compiled examples from the Internet for each class where several dialectal variations of languages are considered. The same method was applied to a very small collection of texts for French.

Our study provides insights into the cross-lingual behavior of the temporal structure of age expressions particularly in the type AdvT–something–TimeN as realized in Spanish and French.

We made an empirical verification of a substantial degree of parallelism between the realization of such age expressions in Spanish and French but showed the differences in their frequency, structure and variety.

Acknowledgements. Work partially supported by Mexican Government (CONACYT, SNI), Government of Mexico City (ICYT project PICCO10-120), European project 269180 WIQ-EI, project “Answer Validation through Textual Entailment” CONACYT-DST (India), and National Polytechnic Institute, Mexico (projects SIP 20111146, 20113295; COFAA).

References

1. Barcelo, G., Cendejas, E., Bolshakov, I., Sidorov, G.: Ambigüedad en nombres hispanos. *Revista Signos* 42(70), 153–169 (2009)
2. Barceló, G., Cendejas, E., Sidorov, G., Bolshakov, I.A.: Formal Grammar for Hispanic Named Entities Analysis. In: Gelbukh, A. (ed.) *CICLing 2009. LNCS*, vol. 5449, pp. 183–194. Springer, Heidelberg (2009)
3. Bittar, A.: ISO-TimeML Annotation Guidelines for French. Version 1.0 (2010), <http://www.linguist.univ-paris-diderot.fr/~abittar/docs/FR-ISO-TimeML-Guidelines.pdf>
4. Ferro, L., Gerber, L., Mani, I., Sundheim, B., Wilson, G.: TIDES 2005 Standard for the Annotation of Temporal Expressions. MITRE Corporation (2005)
5. Gaiffe, B., Nehbi, K.: Les éditions régionales et les informations générales de l’Est Républicain version 0.3 (1999, 2002, 2003), <http://www.cnrtl.fr/corpus/estrepublicain/>
6. Galicia-Haro, S.N.: Using Electronic Texts for an Annotated Corpus Building. In: 4th Mexican International Conference on Computer Science, ENC 2003, Mexico, pp. 26–33 (2003)
7. Galicia-Haro, S.N., Gelbukh, A.F.: Assessing Context for Age-Related Spanish Temporal Phrases. In: Coyle, L., Freyne, J. (eds.) *AICS 2009. LNCS*, vol. 6206, pp. 92–102. Springer, Heidelberg (2010)
8. Gelbukh, A.F., Bolshakov, I.A.: Internet, a true friend of translator: the Google wildcard operator. *International Journal of Translation* 18(1-2), 41–48 (2006)
9. Gustavii, E.: Target Language Preposition Selection—an Experiment with Transformation-Based Learning and Aligned Bilingual Data. In: *Proceedings of EAMT 2005* (2005), http://stp.lingfil.uu.se/~ebbag/Target_Language_Preposition_Selection.pdf
10. Kilgarriff, A.: Googleology is Bad Science. *Computational Linguistics* 33, 147–151 (2007)
11. Krzeszowski, T.: *Contrasting Languages: The Scope of Contrastive Linguistics*. Mouton De Gruyter, Berlin (1990)
12. Lefever, E., Macken, L., Hoste, V.: Language-independent bilingual terminology extraction from a multilingual parallel corpus. In: *Proceedings of the EACL 2009*, pp. 496–504 (2009)
13. Och, F.J., Ney, H.: The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics* 30, 417–449 (2004)
14. Salmon-Alt, S., Bick, E., Romary, L., Pierrel, J.M.: La FREEBANK: Vers une base libre de corpus annotés. In: *Proceedings of TALN 2004*. Fes, Morocco (2009), Online resource: http://corp.hum.sdu.dk/tgrepeye_fr.html
15. Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., Pustejovsky, J.: TimeML Annotation Guidelines Version 1.2.1 (2006), http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf
16. Saurí, R., Saquete, E., Pustejovsky, J.: Annotating Time Expressions in Spanish. TimeML Annotation Guidelines. Version TempEval-2010

Sentiment Analysis of Urdu Language: Handling Phrase-Level Negation

Afraz Zahra Syed¹, Muhammad Aslam², and Ana Maria Martinez-Enriquez³

^{1,2} Department of CS & E, U.E.T., Lahore, Pakistan

afrazsyed@yahoo.com, maslam@uet.edu.pk

³ Department of CS, CINVESTAV-IPN, D.F. Mexico

ammartin@cinvestav.mx

Abstract. The paper investigates and proposes the treatment of the effect of the phrase-level negation on the sentiment analysis of the Urdu text based reviews. The negation acts as the valence shifter and flips or switches the inherent sentiments of the subjective terms in the opinionated sentences. The presented approach focuses on the subjective phrases called the SentiUnits, which are made by the subjective terms (adjectives), their modifiers, conjunctions, and the negation. The final effect of these phrases is computed according to the given model. The analyzer takes one sentence from the given review, extracts the constituent SentiUnits, computes their overall effect (polarity) and then calculates the final sentence polarity. Using this approach, the effect of negation is handled within these subjective phrases. The main contribution of the research is to deal with a morphologically rich, and resource poor language, and despite of being a pioneering effort in handling negation for the sentiment analysis of the Urdu text, the results of experimentation are quite encouraging.

Keywords: Natural language processing, computational linguistics, sentiment analysis, opinion mining, shallow parsing, Urdu text processing.

1 Introduction

The web 2.0 has emerged as a far-reaching and extensive resource of the user generated text in the form of forums, blogs, reviews, social and political discussions, etc. This information overloading has given rise to various new and unexplored aspects of the natural language processing (NLP). It is an apparent reality that such a huge body of knowledge, good and bad both, generated by millions of minds, around the globe, cannot be left free and unbridled [1]. This need for automatic and fast analysis of such text has given rise to the field of sentiment analysis (SA) [2].

Although for English language, this area is under consideration from the last decade [3][4][5][6]. But, unfortunately, morphologically rich languages (MRLs), i.e., Arabic, Turkish, Urdu, Finnish, etc, are relatively overlooked by the research community, because in these languages, the word level complexity is very high due to the frequent morphological operations [7]. In addition to complex morphology, Urdu language has some other distinctive features, which make it a challenging domain,

e.g., influences from various languages, morphological complexity, lexicon intricacy, context sensitivity of the script, and free word order due to independent case marking. For the same reasons, sentiment analysis algorithms and approaches, developed for other well-studied languages, are not effective for Urdu text.

Therefore, this paper presents a pioneering effort for the handling of the phrase-level negation in automatic sentiment analysis of the Urdu text, because for Urdu language no such effort is available so far. We propose a grammatically motivated model, which can handle the language specific issues of this morphologically rich and resource poor language [8]. We use the sentiment annotated lexicon based approach. The current research related to this approach, mainly fall into two categories, i.e., *word-level classification and phrase-level classification*. The *word-level classification* focuses on the polarity orientation and intensity of the subjective words and marks them with the word polarities. In early works on sentiment analysis, word-level classification was considered sufficient [9][10]. For example [10] consider adjectives only to identify the inherent sentiments. But, these approaches were unable to handle the effect of negation, i.e., not, never, no, etc, which are frequently used in opinions and comments and flip the polarities of the subjective words [2]. Due to this deficiency, the concept of *phrase-level classification appeared* [3][4][5].

We apply phrase-level classification method, with a distinction that, we emphasize on the subjective phrases called the *SentiUnits* [6]. The *SentiUnits* contain adjectives as the core terms and include the negation particles as their logical constituents. Hence, the total effect of the negation is dealt along with the effect of the subjective words. This approach is much appropriate to handle the free word order property of the Urdu language. Also, it handles the variant grammatical structures of the Urdu sentences, very successfully, as indicated by the experimentation results.

The rest of the paper is organized in five sections. Section 2 gives a brief overview of the Urdu language and discusses the challenges, which are put forward due to its distinctive features. Section 3 focuses on the state of the art research related to our approach. Section 4 presents our approach along with the system model and illustrate it with examples. The experimentation and the results are given in Section 5. Finally, Section 6 concludes our effort with some future directions.

2 Distinctive Feature of Urdu Language

Urdu is a major language spoken and understood around the globe. Here, we describe some distinctive features of the language, which make it a challenging domain for NLP community [11].

Influencing languages: The main dialects of Urdu are Modern Vernacular Urdu, Dakhini, Rekhta, Pinjari, and Hindi. Its vocabulary and writing style is more inclined towards Arabic and Persian, i.e., from right to left and cursive, but grammar is influenced by Sanskrit. It means that, the sentence structures, the morphological operations, and the word order is Sanskrit based [12].

Morphological complexity: Urdu lies in the category of morphologically rich languages (MRLs) like Arabic, Persian, Chinese, Turkish, Finnish, and Korean. The MRLs require considerable challenges for natural language processing, machine translation and speech processing [7]. These languages are distinctive due to highly

productive and frequent morphological processes at the word level, e.g., compounding, reduplication, inflection, agglutination and derivation, etc. Due to these morphological operations the same root words can generate multiple word forms. This makes the stemming process quite challenging. For example, the word “علم” (*ilm*, knowledge) is the root of the words “عالم” (*aalim*, knowledgeable), “عالیہ” (*aalimah*, female knowledgeable), “معلم” (*moalim*, educator), “معلمه” (*moalimah*, female educator), “معلوم” (*maalooom*, know), and “معلومات” (*maaloomaat*, information).

Lexicon Intricacy: The intricacy of the lexicon of a particular language depends upon the inherent complexity of the language which makes its vocabulary. The Urdu language is a blend of many languages and has linguistic influences from Arabic, Persian, Turkish, Sanskrit and English. For example, the words, “قمر” (*qamar*, moon), “پوجا” (*pooja*, worship), “بدرین” (*badtareen*, worst), and “تلیگرام” (*telegram*, telegram) are respectively, Arabic, Sanskrit, Persian, and English loan words. This variability in vocabulary results into the variability in the grammar rules. Not only the morphological operations are inconsistent but also the sentence structures changes, because most of the loan words follow the grammar rules of their parent language. For example, the Sanskrit language based adjectives show inflection to agree with the noun they qualify, like, “جیسا” (*jaisa*, such as), becomes “جیسی” (*jaisee*, such as) and “جیسے” (*jaisay*, such as) for gender and number, respectively. But, Persian loan adjectives show no inflection, like, “تازہ” (*tazah*, fresh) [13].

Context sensitivity of the script: Urdu script is cursive and context sensitive. The Urdu alphabets are categorized as joiners and non joiners. In a word, the joiner alphabets join with each other in different shapes according to their position in the word. If the ending alphabet of a word is a joiner then it tends to connect with the first letter of the next word, resulting into a misidentification of the word boundaries. Hence, this context sensitivity result into word segmentation issues [14], as the spaces are not always exact indicators of the word boundaries, as in case of English.

Independent case markers: The case markers mark the grammatical roles or functions to the words, with which they are attached. Usually, they are lexically attached through inflection or derivation. But, in Urdu language, the case markers are independent lexical units and are treated as independent parts of speech [15]. They influence the sentence structure and can cause grammatical ambiguities. For example, the free word order property of the Urdu text is due to lexically independent case markers, e.g., both the phrases; “رنگوں کے نام” (*rangoon kay naam*, colors' names) and “نام رنگوں کے” (*naam rangoon kay*, colors' names) are accurate with the same meanings, but have different word order because of the case marker “کے” (*kay*).

We deduce that Urdu has certain distinctive features, like; complex morphology, variable vocabulary and grammatical rules, independent case marking and context sensitive script. Hence, it requires different approaches and algorithms for sentiment analysis, which can incorporate these issues, efficiently.

3 State of the Art Research

Considering the linguistic features of the Urdu language, we employ the lexicon based approach of sentiment analysis. The system is composed of an analyzer and a

sentiment annotated lexicon of the Urdu text. Along with the use of the lexicon, we focus on the phrase level classification of the given text, in which, the effect of negation is handled within the phrase. Here, we describe phrase-level classification approaches along with some contributions which handled negation independently.

Phrase-level classification: The research works, which emphasize on the extraction of the phrases, define these phrases in different ways. For example the work presented in [3] stresses on the identification of the subjective nouns, modified by the adjectives and extracts the phrases containing these nouns. The work in [4] focuses on proverb and adjective based phrases, and converts the sentences into pre-structured patterns with adjectives and adverbs as the core terms. Whereas [5] extracts the subjective expressions in the given texts. The approaches presented in [16][17][6] concentrate on the identification of the appraisal expressions.

Approaches for handling negation: Negation handling in sentiment analysis as an independent task is not yet a well solved issue, even for English text [18][19]. This is because of the context sensitive use of the negation particles. The first computational model for the treatment of the negation is presented in [20]. It models negation via contextual valence shifting. The polarity of a subjective expression is reversed due to the use of negation mark. The work in [21] also proposes an approach for contextual valence shifting and in addition to dealing with the simple negation particles; this work decides a simple scope for negation, i.e., if the negation particle immediately precedes a subjective expression then its polarity is flipped. As an extension of this work, a parser is added for scope computation in [22].

The work in [23] uses supervised machine learning method. It selects the features, like, negation features, shifter features, and polarity modification features, for an advanced negation modeling. A technique to compute the polarity of complex noun phrases and headlines using compositional semantics is presented in [24]. The research in [18] investigates the effect of different scope models of negation. It achieves the scope detection through, the heuristic rules focused on polar expressions and static and dynamic delimiters.

All of the above contributions treat the negation as independent lexical units, which can affect the entire words, phrases or sentences. But, there are many cases in which, the negation comes within the word structure, e.g., “بے فایدہ” (bay fayeeda, useless). There are a few works addressing this type of negation [24].

Sentiment Analysis in Urdu: Urdu language processing is not a very well explored field. This is due to the distinguishing linguistic features of the language as discussed in the Section 2. There are only a few contributions for sentiment analysis of the Urdu text, [6] and [25]. The both contributions have emphasized on the extraction of the appraisal expressions, from the given text and use shallow parsing based chunking with adjectives as the head words. Apart from sentiment analysis there are also some worth mentioning contributions in Urdu language processing. For example, the EMILLE project is related to a multi-lingual corpus for the South Asian Languages. Their independent POS tagged corpus for Urdu has about 1,640,000 words [26]. [11], gives another POS tagged corpus. The automatic extraction of the Urdu lexicon using

corpus is presented in [12] and [27]. The corpus in [27] is based on cleaned text from Urdu news websites, having nearly 18 million words. [8], develops semantic-role-labeled corpus using cross lingual projections.

The discussion in Section 3 shows that, for English language the field of automatic sentiment analysis is well established, even there are many contributions which have attempted the handling of negation. But, Urdu language processing still needs much consideration and there are still several aspects to explore.

4 Treatment of Negation

Negation is one of the most frequent linguistic structures that change the word, phrase, or sentence polarity. Negation is not only limited to the negation markers or particles, like, *not*, *never*, or *no*, but there are various concepts, which serve to negate the inherent sentiments of a comment. Moreover, the presence of the negation influences the contextual polarity of the words but it does not mean that all of the words conveying sentiments will be inverted [19].

There are different forms of negation discussed in the literature. Here, we give three main forms [23]. Negation can be morphological, i.e., attached as prefix or suffix making a single lexical unit, e.g., the prefix “بے” (bay) as in “بے پرواء” (*bayparwah*, careless) is used to negate the word “پرواء” (*parwah*, care). Or, it can be implicit, like, “بے گھری تھمارے معیار سے کم ہے” (*yeh gharee tumharay mayaar say kam hay*, this watch is below your standard/level). This comment even with the absence of a negation particle conveys a negative opinion. Lastly, the negation can be explicit with the use of negation particles, e.g., “بے گھری تھمارے معیار کے مطابق نہیں” (*yeh gharee tumharay mayaar kay mutabiq naheen*, this watch is not according to your standard/level). Consider another example “بے بھیں” (*yah film buri naheen*, this film is not bad). In these sentences, a negative and then a positive effect is conveyed by the negation particle “بھیں” (*naheen*, not). This type of negation can be determined automatically. Most of the efforts for an automatic treatment of the negation for sentiment analysis give attention to the last type, in which the negation appears explicitly [20][21][22].

4.1 Negation in Urdu Language

In our work, we focus on the negation which appears explicitly in the given text through negation particles. In Urdu, both sentential and constituent negation exists [13]. Some prominent negation particles are “مت” (*mat*, don’t), “ن” (*na*, no), “نہیں” (*naheen*, not), “بنا” (*bina*, without), and “بھیر” (*baghair*, without).

Sentential Negation: The negative particles “نہیں” (*naheen*, not), “مت” (*mat*, don’t) and “ن” (*na*, no) are used to express sentential negation. The particle “نہیں” (*naheen*, not) appears before the main verb, which may or may not be followed by an auxiliary verb. In imperative constructions, the particles “مت” (*mat*, don’t) and “ن” (*na*, no) are used in the preverbal position. Table 1 gives the use of these negation particles before the main verbs; “جاتا” (*jata*, goes) and “پڑھو” (*parho*, read).

Table 1. Examples of sentential negation from Urdu text

Examples
”وہ سکول نہیں جاتا ہے“ (who school naheen jata hay, He doesn't go to the school.)
”کتاب مت پڑھو“ (kitaab mat parho, Don't read the book.)
”کتاب نا پڑھو“ (kitaab na parho, Don't read the book.)

Constituent Negation: This negation is used to negate some particular constituent/constituents of a sentence. The negative particle comes after the negated constituent. Some common constituent negation particles are; ”نہیں“ (naheen, not), ”مت“ (mat, don't), ”نہ“ (na, no), ”علاوه“ (ilaawa, except), ”سو“ (siva, except) and ”بنا“ (bina, without). In Table 2, the negation particles, ”نہیں“ (naheen, not), ”مت“ (mat, don't), ”نہ“ (na, no) and ”سو“ (siva, except) are used after the negated constituent.

Table 2. Examples of constituent negation from Urdu text

Examples
کبیرہ کالا نہیں نیلا ہے (camera kala naheen neela hay, The camera is blue, not black)
انگور مت/نا خریدوانار خریدو (angoor mat/na khareedo anar khareedo, Don't buy grapes, buy pomegranate)
موبائل کے سوا سب اچھا ہے (mobile kay rang kay siwa sab acha hay, All is fine with the mobile except its color)

Use of multiple negation particles: Sometimes the double negation marks are used to put emphasize on something. For example, in the sentence, ”وہ سکول نہیں نا گیا“ (woh school naheen na gya, he did not go to school. The two negation particles ”نہیں“ (naheen, not) and ”نہ“ (na, no) are used to give stress or emphasize.

Negation in coordinate structures: In the coordinate structures the negation particle does not move to the coordinate point, unless the identical element is deleted from the second negative conjunct. But, in the situation like ‘neither ... nor’, it appears in the beginning position. For example, ”نہ گھر نیا ہے نا بودار“ (na ghar nya hay, na hawa daar, The house is neither new and nor ventilated).

4.2 Phrase-Level Negation: SentiUnit Extraction Model

For our approach of automatic sentiment classification, based on subjective phrases, we give emphasis, on the accurate identification of sentiment expressions, called the *SentiUnits*. The SentiUnits are made by single or multiple words and are exclusively responsible for the entire sentimental orientation of a single sentence, from the given review. The explicit negation particles, e.g., ”نہیں“ (naheen, not), ”مت“ (mat, no), ”نہ“ (na, no) etc, are handled within these expressions. The model is grammatically motivated and works on the grammatical structure level of the sentences. It uses a sentiment-annotated lexicon based approach for the identification of such expressions from the corpuses of Urdu text based reviews (see Figure 1).

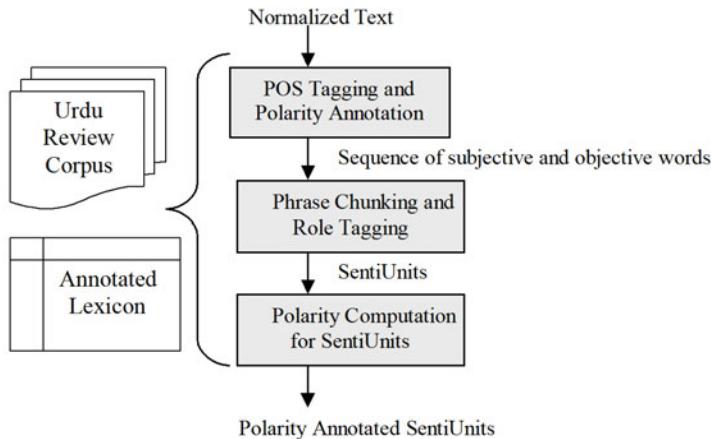


Fig. 1. SentiUnit extraction and polarity computation

For a given Urdu language based review, the SentiUnit extraction and polarity computation takes place in three phases;

- Firstly, the normalized text is passed to the parts-of-speech (POS) tagger, which assigns POS tags to all the terms. Along with this tagging the word polarities are also annotated to the subjective words. This polarity annotation takes place with the help of the sentiment annotated lexicon of the Urdu text.
- These annotated subjective terms (adjectives) are considered as the headwords for the next phase in which shallow parsing is applied for phrase chunking and the adjectival phrases are chunked out. Now, these chunks are converted into SentiUnits by attaching the negation, modifiers, conjunctions, etc.
- In the last phase, the identified SentiUnit are analyzed for polarity computation. The polarity of the subjective terms is treated with the combined effect of the negation, if it exists in the SentiUnit. Hence, the overall sentiment or impact of the SentiUnit is a combination of its constituents.

4.3 Overall Review Polarity

Figure 2 shows the overall process of the review polarity calculation. When the system is given a review for classification it sets the review polarity R_p and sentence count S_{Count} to zero. Then, it takes each sentence one by one. The analysis begins with the text normalization resulting into word segmentation. These words are passed to the SentiUnit extraction and polarity computation module, which gives polarity annotated SentiUnits as discussed in Section 4.2. Now, the sentence polarity S_p is computed using the polarities of its constituent SentiUnits. The total R_p is the sum of all known sentence polarities S_p . Then, R_p is compared with the threshold value. If R_p is greater than the threshold, then, the review is positive and vice versa.

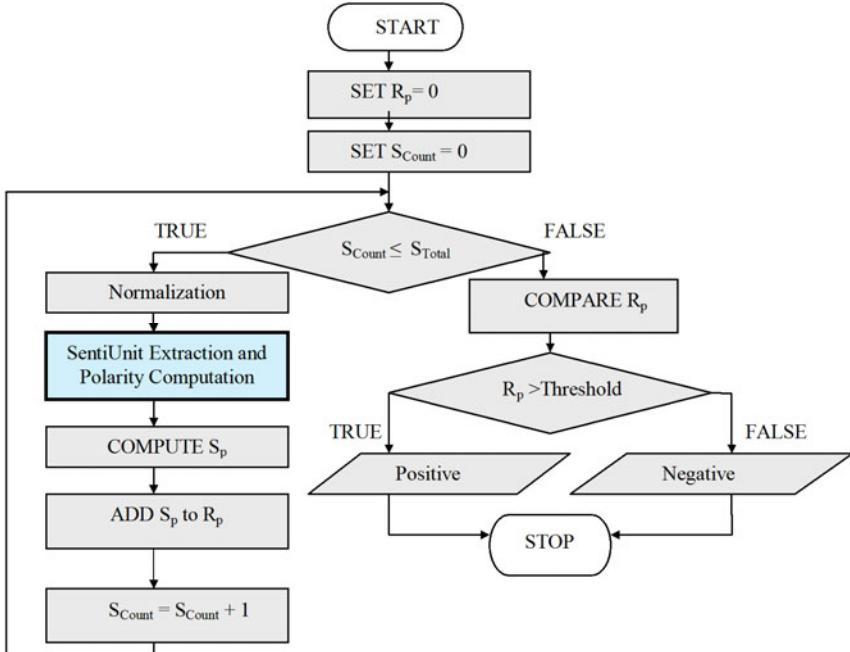


Fig. 2. Computation of the overall polarity of the Urdu text based review

Example: As an example we take a short review from Urdu language based movie review corpus and apply our approach of SentiUnits extraction on it.

”شکل اچھی نہیں۔ نا ہی بدایت کاری قابل ستائش ہے۔ فلم کی کہانی بورنگ ہے۔ بیرو کی اداکاری اور“

(film ki khani boring hay, hero ki adakari aur shakal achi naheen, na hi hdaayetkari qabil-e-staayesh hay, The story of the film is boring. Hero's acting and looks are not good. Nor is the direction appreciable.)

The review is based on three comments. In the first comment no negation particle is used, the SentiUnit is based on an adjective “بورنگ” (boring, boring) with negative polarity resulting into an overall negative impact. In the second comment, the SentiUnits is made by a positive adjective “اچھی” (achi, good) and a negation mark “نہیں” (naheen, not), which reverse the effect of the adjective to make an overall negative impact. The SentiUnit in the third comment is based again on a positive word “قابل ستائش” (qabil-e-staayesh, appreciable) and a negation mark “نا” (na, nor), which appear in the beginning of the comment and hence, it conveys a negative opinion. So, the overall polarity of the review is negative. The POS tagging and phrase chunking of the review is given in Table 3. Column 1 gives the POS tags of each comment, where, N, PM, ADJ, NEG, and CJC represent respectively noun, possession marker, adjective, negation, and conjunction. In column 2, the noun phrases NP along with SentiUnits SU are given.

Table 3. POS tagging and phrase chunking of the given review

Phrases & SU	POS tags	Comments
[NP] [SU]	[N + PM + N] [ADJ + NEG]	فلم کی کہانی بورنگ ہے
[NP] [CJC]	[N + PM + N] [CJC]	شکل اچھی نہیں بپرو کی اداکاری اور
[NP] [SU]	[N] [ADJ + NEG]	نا بی بیان کاری قابل ستایش ہے
[NP] [SU]	[NEG] [N] [ADJ]	

5 Experimentation

For performance evaluation of the sentiment analysis systems, the experiments are performed on real corpuses of user reviews. For this purpose, reviews corpuses are collected and sentiment annotated lexicons are developed. From the earliest works, the SA research community focused on construction of such lexicons, e.g., General Inquirer [28] was the first manually collected lexicon of English text and [9] presented the first automatically generated lexicon of adjectives. Afterwards, this became popular area for researchers and they applied different approaches. For example, the work in [29] employed the statistical approaches, while [9] and [30] utilized the corpus-based and lexical relationships based approaches, respectively. Some other contributions used machine learning methods [3] [4] [5] [10].

Most of these English text based lexicons are developed by using already prepared linguistic recourses. But, for Urdu language, we find no such readily available resources which can be used for lexicon construction [6][8]. Therefore, for experimentation, we manually developed and annotated a lexicon with maximum coverage of the vocabulary, particularly the polar terms like adjectives and their modifiers. Also, we have included all types of negation particles which are used explicitly. As the performance of a sentiment analyzer is highly domain specific, therefore we have carefully selected the adjectives and other subjective words from the domain of movies for polarity annotation. Moreover, we have collected a corpus of Urdu text based movie reviews. This corpus contains 450 reviews, among which 226 are positive and 224 are negative. The collected reviews are mainly for action, comedy, and horror movies. The average length of the reviews is 4 to 6 sentences.

5.1 Results

We carry out experimentation to analyze the behavior of the negation. For this reason, we divide the dataset into three different sets of data. We clean out the neutral comments from all the three sets, during the test-bed normalization process.

Set 1 includes the sentences, in which, both implicit and explicit negation is absent. The polarity of these sentences depends only on the subjective terms and other polarity shifters. *Set 2* contains those sentences, in which only explicit negation particles are used and implicit negation is absent. To compile *Set 3*, we add implicit negation sentences in *Set 2*. In this set both implicit and explicit negation is present in addition to polar terms.

Table 4. Experimentation results in terms of precision, recall, and f-measure

Test Data Sets	Precision	Recall	F-Measure
<i>Set 1</i>	0.864	0.837	0.850
<i>Set 2</i>	0.590	0.779	0.677
<i>Set 3</i>	0.510	0.615	0.558

Table 4 gives the results from the three sets of data, in terms of precision, recall, and f-measure. The *Set 1* in which only polar terms are present, gives the best results of the classification. Whereas, the results from Set 2 are lower than the previous one, as it contains only the sentences with the negation particles. From this result, it is inferred that the negation particles can cause relatively high rate of missclassification. But, the average results from *Set 1* and *Set 2* are quite satisfactory. The results from *Set 3* show that the implicit negation still needs an improved treatment. Apart from the results, we have following worth mentioning observations:

- On the average two to three negation particles appear in a single review and the use of negation is author dependent; some authors tend to use more negative particles than others. “*تہیں*” (*naheen*, not) is the most used particle. In comparative, sentences, the negation particle “*لے*” (*na*, no) is used with multiple targets of the appraisal.
- The sentential negation is rarely misclassified as compare to the constituent negation.
- Morphological negation is automatically handled, because most of the words inflected by the lexical negation marks, e.g., “*بے*” (*bay*), “*بے*” (*ba*), etc, are already present in the lexicon and are annotated with respective polarities, e.g., “*فایدہ بے*” (*bayfayeeda*, useless) is a lexical entry with a negative polarity.

6 Conclusions

In this paper, we have applied the lexicon based sentiment analysis approach on the Urdu language, using phrase level classification to treat negation. We consider two major contributions of the research. Firstly, it handles a challenging language, with complex morphology and least explored by language processing. Urdu language has distinctive linguistic characteristics and therefore, requires different approaches. Secondly, it focuses on the extraction of the subjective expressions, called SentiUnits, within which the effect of negation is automatically considered.

For the experimentation, we have collected a corpus based on movie reviews. The results of the experiments performed on three different sets of data show that our approach works efficiently for handling the sentences with polar terms and explicit negation, which are main concern of this contribution. But, implicit negation still needs much consideration; we leave this aspect as our future endeavor. Moreover, we plan to extend this model by considering noun phrases as the targets of the extracted SentiUnits. Domain adaptation is also another goal, which can be achieved by extending our annotated lexicon with the words from multiple domains.

References

1. Glaser, J., Dixit, J., Green, D.P.: Studying hate crime with the Internet: What makes racists advocate racial violence? *Journal of Social* 58(1), 177–193 (2002)
2. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Foundation and Trends in Information Retrieval* 2(1-2), 1–135 (2008)
3. Riloff, E., Wiebe, J., Wilson, T.: Learning subjective nouns using extraction pattern bootstrapping. In: *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pp. 25–32 (2003)
4. Turney, P.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: *Proc. the Association for Computational Linguistics (ACL)*, pp. 417–424 (2002)
5. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In: *Proc. the Conference on Empirical Methods in Natural Language Processing* (2003)
6. Syed, A.Z., Aslam, M., Martinez-Enriquez, A.M.: Lexicon Based Sentiment Analysis of Urdu Text Using SentiUnits. In: Sidorov, G., Hernández Aguirre, A., Reyes García, C.A. (eds.) *MICAI 2010. LNCS (LNAI)*, vol. 6437, pp. 32–43. Springer, Heidelberg (2010)
7. Abdul-Mageed, M., Korayem, M.: Automatic Identification of Subjectivity in Morphologically Rich Languages: The case of Arabic. In: *1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, WASSA* (2010)
8. Mukund, S., Ghosh, D.: Using Cross-Lingual Projections to Generate semantic Role Labeled Corpus for Urdu- A Resource Poor Language. In: *23rd International Conference on Computational Linguistics COLING* (2010)
9. Hatzivassiloglou, V., McKeown, K.: Predicting the semantic orientation of adjectives. In: *Proc. the Joint ACL/EACL Conference*, pp. 174–181 (1997)
10. Hatzivassiloglou, V., Wiebe, J.: Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In: *18th International Conference on Computational Linguistics, New Brunswick, NJ* (2000)
11. Muaz, A., Ali, A., Hussain, S.: Analysis and Development of Urdu POS Tagged Corpora. In: *Proc. the 7th Workshop on Asian Language Resources, IJCNLP* (2009)
12. Humayoun, M., Hammarström, H., Ranta, A.: Urdu morphology, orthography and lexicon extraction. In: Farghaly, A., Megerdoomian, K. (eds.) *Proceedings of the 2nd Workshop on Computational Approaches to Arabic Scriptbased Languages, Stanford LSA*, pp. 59–66 (2007)
13. Schmidt, R.: *Urdu: An Essential Grammar*. Routledge Publishing, New York (1999)
14. Lehal, G.S.: A Word Segmentation System for Handling Space Omission Problem in Urdu Script. In: *Proc. the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP)*, the 23rd International Conference on Computational Linguistics, COLING, pp. 43–50 (2010)
15. Rizvi, S.M.J., Hussain, M.: Modeling case marking systems of Urdu-Hindi languages by using semantic information. In: *NLP-KE* (2005)
16. Bloom, K., Argamon, S.: Unsupervised Extraction of Appraisal Expressions. In: Farzindar, A., Kešelj, V. (eds.) *Canadian AI 2010. LNCS (LNAI)*, vol. 6085, pp. 290–294. Springer, Heidelberg (2010)
17. Whitelaw, C., Garg, N., Argamon, S.: Using appraisal groups for sentiment analysis. In: *Proc. the ACM SIGIR Conference on Information and Knowledge Management (CIKM)*, pp. 625–631. ACM (2005)
18. Jia, L., Yu, C., Meng, W.: The effect of negation on sentiment analysis and retrieval effectiveness. ACM (2009)

19. Wiegand, M., et al.: A survey on the role of negation in sentiment analysis. In: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing 2010. Association for Computational Linguistics (2010)
20. Polanyi, L., Zaenen, A.: Context Valence Shifters. In: Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text (2004)
21. Kennedy, Inkpen, D.: Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. In: Proceedings of FINEXIN (2005)
22. Kennedy, Inkpen, D.: Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. Computational Intelligence 22 (2006)
23. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. In: Proc. HLT/EMNLP (2005)
24. Moilanen, K., Pulman, S.: The Good, the Bad, and the Unknown. In: Proceedings of ACL/HLT (2008)
25. Syed, Z.A., Aslam, M., Martinez-Enriquez, A.M.: Adjectival Phrases as the Sentiment Carriers in the Urdu Text. Journal of American Science 7(3), 644–652 (2011)
26. Hardie, A.: Developing a tag-set for automated part-of-speech tagging in Urdu. In: Archer, D., Rayson, P., Wilson, A., McEnery, T. (eds.) Proc. the Corpus Linguistics Conference. UCREL Technical Papers, vol. 16, Department of Linguistics, Lancaster University, UK (2003)
27. Ijaz, M., Hussain, S.: Corpus based Urdu Lexicon Development. In: Conference on Language Technology (CLT 2007). University of Peshawar, Pakistan (2007)
28. Stone, P.J., Dunphy, D.C., Smith, M.S., Ogilvie, D.M.: The General Inquirer: A Computer Approach to Content Analysis. MIT Press, Cambridge (1966)
29. Turney, P., Littman, M.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems 21(4), 315–346 (2003)
30. Kamps, J., Marx, M., Mokken, R.J., De Rijke, M.: Using WordNet to measure semantic orientation of adjectives. In: Proceedings of LREC-04, 4th International Conference on Language Resources and Evaluation, Lisbon, PT, vol. IV, pp. 1115–1118 (2004)

Unsupervised Identification of Persian Compound Verbs

Mohammad Sadegh Rasooli¹, Heshaam Faili², and Behrouz Minaei-Bidgoli¹

¹ Department of Computer Engineering, Iran University of Science and Technology

{rasooli@comp., b_minaei@iust.ac.ir}

² School of Electrical & Computer Engineering, Tehran University

hfaili@ut.ac.ir

Abstract. One of the main tasks related to multiword expressions (MWEs) is compound verb identification. There have been so many works on unsupervised identification of multiword verbs in many languages, but there has not been any conspicuous work on Persian language yet. Persian multiword verbs (known as compound verbs), are a kind of light verb construction (LVC) that have syntactic flexibility such as unrestricted word distance between the light verb and the nonverbal element. Furthermore, the nonverbal element can be inflected. These characteristics have made the task in Persian very difficult. In this paper, two different unsupervised methods have been proposed to automatically detect compound verbs in Persian. In the first method, extending the concept of pointwise mutual information (PMI) measure, a bootstrapping method has been applied. In the second approach, K-means clustering algorithm is used. Our experiments show that the proposed approaches have gained results superior to the baseline which uses PMI measure as its association metric.

Keywords: multiword expression, light verb constructions, unsupervised identification, bootstrapping, K-means, Persian.

1 Introduction¹

A collocation is "an arbitrary and recurrent word combination" [1] or frequently appeared sequence of adjacent words [2]. In [3], collocations are classified into two main categories: 1) theoretical, and 2) empirical. Empirical collocations are the ones occurring in corpora while theoretical collocations are the ones known in linguistics. Collocations are ranged from lexically restricted expressions (e.g. strong tea), phrasal verbs (e.g. look after), technical terms (e.g. prime minister), and proper names (e.g. Los Angeles) to idioms (e.g. spilt the beans) [4]. In [4], five types of n-grams were considered as true collocations: 1) stock phrases (e.g. major problem), 2) Named entities (e.g. Prague castle), 3) support verb constructions (e.g. make decision), 4) technical terms (e.g. prime minister), and 5) idiomatic expression (e.g. kick the bucket). A multiword expression (MWE) is known as a type of collocation that refers to a single concept [5] which the whole meaning is often not a function of its constituent meaning parts [6] and differs in the meaning level [7]. Because of the idiosyncrasy in meaning, MWEs are considered

¹ This Paper is funded by Computer Research Center of Islamic Sciences (C.R.C.I.S).

different from multi word phrases [8]. One of the main tasks in NLP is the detection of MWEs. In [9], it is stated that MWE is one of the two main challenges in NLP. In addition, MWEs are very frequent in real language data occurrences [9]. Hence, the problem of their identification should be solved in order to have a sound learning of the language.

The majority of MWEs are verbal expressions such as verb constructions (LVC), verb noun constructions (VNC), and light verb particle constructions (VPC) [10]. VNCs are either idiomatic or literal [7]. Idioms are defined as sequences of words that are semantically idiosyncratic and non-compositional [11]. VPCs consist of a head verb and one or more obligatory particles such as prepositions (e.g. make up), adjectives (e.g. cut short) or verbs (e.g. let go) [12]. Light verbs are classes of verbs that independently lack semantic force to function as predicates and need some predicative nouns or adjectives to complete their meanings [13]. An LVC is made up of a light verb and a preverb (in most cases, a noun) which its meaning is non-compositional. Furthermore, the preverb (nonverbal part) has a verbal or predicative meaning and the verbal part lacks its literal semantics [14]. The most challenging nature of LVC is its lexical semantic variation, i.e. polysemy; in which the verbal part of the construction tends to have different meanings according to the context [6]. The syntactic patterns of LVC occurrences in language corpora tends to be similar to each other [14], however, the argument structure of the LVC is different from the light verb itself. This kind of evidence is very frequent in languages like Persian, Korean and Kurdish [13]. In Persian, while the nonverbal part of the LVC is not the object of the verbal part; the verbal part is free of literal semantics and only works as a verbal agent for the predicative preverb to express an action in the sentence [13, 15]. There is a major difference between Persian compound verbs (the other name of Persian LVC) and the ones in other languages like English. Although MWEs are not productive in other languages [9], in Persian any predicative noun can combine with its related verbal element and make a new compound verb. For example, the verb "kardan" in Persian (synonym to "to do" in English) makes most of the compound verbs in Persian and many new predicative nouns can combine with it and make a new LVC in Persian. For instance the new Persian word "email" can combine with "kardan" and make a new LVC ("email kardan (email - to do)" means "to compose email") and the argument structure or valency (in the notion of dependency grammar) changes.

In this paper, by considering special aspects of Persian compound verbs, we proposed two unsupervised learning methods on Persian corpus in order to improve the task of Persian compound verb identification. In section 2, some related works on MWEs (especially multiword verbs) are described. In section 3, the main challenges in Persian compound verbs are stated and the shortcomings of data resources in Persian are also reminded. Experimental results and conclusion are presented later in the paper.

2 Related Works

Several statistical methods on collocations and MWEs extraction have been proposed that mainly focused on idiomatic expression, LVCs and multiword verbs [16] where most of these methods were based on lexical association measures, such mutual information [4]. One the most famous works in recognizing non-compositional phrases

is done in [17] where pointwise mutual information (PMI) measure was used as the measure of non-compositionality. The main idea of the proposed work was based on the hypothesis that a phrase is non-compositional when "its mutual information differs significantly from the mutual information of phrases obtained by substituting one of the word in the phrase with a similar word". In addition, Dependency relations between verbs and their objects were counted as the co-occurrence evidences. This hypothesis has some deficiencies especially in mutual information overestimation when the counts of data are very small [18]. However, this measure became state of the art in many comparisons. For instance, in [19], 84 different bigram association measures were compared for Czech multiword expressions and PMI gained the best results. In [20], 31 measures were reexamined and the normalized PMI gained the best results. In [21], five classic measures were compared for identifying German pp-verbs and t-score gained the best precision. In [22], with a measure based on thermodynamic models, an identification task was applied in search engines that outperformed the PMI measure with more informative counts from the internet. It is worth to remind that the performance of an association measure really depends on the nature of the data and it can not be said strictly which method retrieved the best results in all cases [4].

Most of the works on identification of multi word verbs were done via unsupervised methods [5]. In [23], a statistical measure based on log-linear models of n-way interaction of words, based on multinomial distribution, was proposed in order to identify English multiword verbs. In [24], three methods based on different linguistic features were used in identifying VPCs. In the first method, a simple part of speech (POS) tagger was used to enumerate potential candidates. In the second one, a simple chunker was used and in the last method some grammatical information was added which not only improved method accuracy but also enumerated both positive and negative evidences in corpus sentences. In [25], a word sense disambiguation (WSD) method was used as a clustering task for discriminating literal and non-literal MWEs. In this work, KE algorithm was used which is based on recognizing similarity between a target sentence and other sentences and this similarity is known as attraction. In [26], with using latent semantic analysis (LSA), a vector based semantic similarity model was applied to word co-occurrence counts to find multi word verbs. In this method, cosine similarity measure was used to compare vectors and the method was evaluated on German texts. In [27], it was assumed that idiomatic expressions appear in a small number of canonical forms and in the other hand, literal multi words appear in several forms and have no restriction in the syntactic form appearance. Based on this assumption, two models of co-occurrence vectors were proposed which were evaluated on BNC corpus to find idioms in VNCs. One of the models was the model proposed in [26] and the other was based on the assumption that canonical forms are more likely to be idiomatic than non-canonical form. In [28], some linguistic features on multi word verbs such as pluralization, passivization, change in determiner type were considered as relevant patterns of syntactic variation. In this work, maximum likelihood estimation was used to estimate the probability of being idiomatic and the probability distribution of syntactic variations was compared via KL-Divergence. In [29], the syntactic fixedness was estimated by summing up pointwise mutual information between all syntactic variations. In [8], statistical measures for estimating

idiiosyncrasy in MWEs were proposed that those measures were based on PMI measure. The extension of this work is written in [11, 14]. In [11], the syntactic fixedness of an idiom was considered as a positive evidence; i.e. the syntax and order in the idiom constituents in data occurrences does not differ from the main form of the idiom. In this work, it is stated that most idioms are known to be lexically fixed and each different syntactic variation of an idiom, different to the main lexical order, is considered as negative evidence [14] and with this assumption, the model proposed by [17] was improved. At last, KL-divergence was used to measure syntactic fixedness of an expression; i.e. the degree of syntactic fixedness of the verb-noun pair was estimated as the divergence of syntactic behavior of the target verb-noun pair from the typical syntactic behavior (prior distribution). In that work, it is shown that methods proposed in [17] and [1] are not significantly better than random selection. In [7], with inspiring models in [8, 26], a vector based similarity measure was used, but the notion of context was changed. Context included all nouns, verbs, adjectives and adverbs occurred in the same paragraph as the target word. With this assumption, five different parameter settings were experimented and the ratio of a word pair in each context was inspired from tf-idf measure in information retrieval.

There are fewer works done in supervised multiword verb identification in comparison to unsupervised methods. In [5], linguistic features such as word lemma, chunk, named entity and part of speech were used as features of classification and then support vector machines (SVM) classifier was used for classification. In this work, the VNC-Token dataset [30] was used as the train data which VNC token expressions are manually tagged as either idiomatic or non-idiomatic. In [31], the context of a paragraph of a target word was mapped into a co-occurrence word vector, then Dice Coefficient and Jaccard index measures were used to estimate the similarity between vectors. In [32], 55 different association measures were mapped into a vector and three different learning methods (linear logistic regression, linear discriminant analysis (LDA), and neural networks) were used to combine these measures. Comparing the mentioned methods, LDA gained the best result. In [4], 84 association measures were compared and a classifier was built by choosing an appropriate threshold for each measure. Furthermore, based on Pearson's correlation coefficient, hierarchical clustering and dimension reduction were done on these association measures in order to handle sparsity in feature combination. Finally, four different learning algorithms (linear logistic regression, linear discriminant analysis, SVM, and neural networks) were used to combine association measures and it was shown that in all cases, the combination of measures outperformed using measures alone. In this work, neural network gained the best result on Prague dependency Treebank (PDT).

3 Challenges in Persian Compound Verb Identification

As mentioned in the previous section, most recent methods in multiword verb identification (both supervised and unsupervised) used some linguistic features which applying them needs a tagged corpus of syntactic features such as dependency relations. One of the major difficulties in this field of research is the lack of reliable datasets such as [30, 33] for supervised learning and datasets such as [34] for unsupervised learning. The only reliable corpus in Persian is Bijankhan corpus [35] that just has

been annotated by part of speech tags and some morphosyntactic features. Some features such as dependency relations in [11, 16, 17, 36], chunks and named entity relations in [5], and dictionary of collocations in [11] are not currently accessible in Persian language. The other major challenge in this problem is the productiveness of Persian compound verbs [37, 38] in which handcrafted verb lists from current dictionaries are not appropriate and reliable. One of the important cues for identifying Persian compound verbs is the change in argument structure (or subcategorization frames) compared to the light verb itself [13, 39] and this cue is not tagged in the current corpus of Persian language. In works such as [11, 27-29, 36], syntactic fixedness was used to identify idiomatic multiword predicates, but this kind of cue is not applicable to Persian. Not only, the nonverbal element in Persian LVC can be inflected but also this inflection does not ever mean that the nonverbal part can not combine with the light verb as a compound verb. Even an inflected noun can be the nonverbal predicate in the compound verb [40].

The other problem in Persian compound verb identification is the sparseness in the data, therefore, methods like the one in [17] do not lead to satisfactory results and for so many verbs the scores like PMI are very near to each other. Hence, this method does not work properly in this case. This sparsity is due to separability [41] and preverb inflection ability [40] in Persian LVCs. Consider the example in (1):

(1) *Man (I) bâ (with) to (you) sohbat-hâ-ye (speak-plural-Ezafe) besyâr (very) ziyyâdi (high) dar (in) mored-e (about-Ezafe) jang-e (war-Ezafe) irân (Iran) va (and) erâq (Iraq) kardam (do-simple past, first person, singular of "kardan")*.

— **Meaning:** I spoke a great deal with you about the war between Iran and Iraq.

As shown in (1), there is a distance of 9 words between the light verb ("kardan") and non-verbal element ("Sohbat"). Also the nonverbal element is pluralized and an Ezafe (an indicator of noun modification [42]) is attached to it². In sentence (1), 4 out of 8 words between the light verb and nonverbal element are nouns ("sohbat", "jang", "irân", and "erâq"), so it can be stated that there are four candidates for being a predicative noun of the light verb "kardan". If we generalize this phenomenon to all sentences in the corpus, the event space for multiword verb is more than a simple adjacency of nouns and verbs. The number of potential candidates increases with the sentence length. In this work, we had a case study of the light verb construction "kardan" in Persian and recognized that 98 percent of nonverbal elements appear adjacent to the light verb at least once and 91 percent of sentences with LVCs are the ones in which light verbs and nonverbal elements are adjacent in Bijankhan corpus. In other words, we seem to face a tradeoff between precision and recall. In the one hand, if we only consider adjacent nouns to the verbs, some valuable information will be lost and in the other hand if we consider all possible candidates, identification precision will decrease.

4 Persian Compound Verb Identification

In this paper, two kinds of unsupervised methods for detecting Persian compound verbs have been experimented. The first task is based on bootstrapping and the second

² In scarce cases, the Ezafe construction appear in typography.

is the k-means clustering algorithm. Furthermore, a modified version of PMI measure is proposed in order to handle data sparseness in Persian compound verbs. This measure is shown in equation (1), in which $fr(x)$ is the frequency of x . We named this measure Probabilistic PMI (PPMI). In the following sections, after introducing PPMI, details of bootstrapping and k-means are described.

$$PMI(x, y) = \log \frac{p(x,y)}{p(x).p(y)} = \log \frac{n.fr(x,y)}{fr(x).fr(y)} \quad (1)$$

4.1 PPMI

If all possible alternatives of a compound verb are considered in PMI, data sparseness will decrease precision. In the other hand, if only adjacent candidates are considered, some valuable information will be lost. In our experiment, we designed a rule based compound verb tagger based on a list of real compound verbs. Linguistic rules for Persian compound verb were used in the tagger to find the verbs. The rules were extracted from Persian grammar books such as [43]. The Bijankhan corpus [35] was tagged with this tagger. The distribution of the distance between the light verb and the nonverbal element is shown in Table 1. This observation made us sure about the deficiency in the classic PMI measure. If we consider all possible candidates, there will not be a good precision in the results. On the other hand, if only adjacent words are considered, some valuable information will be lost. In order to solve this problem, we introduce a modified version of the PMI measure. In this version, each co-occurrence is not counted as one (as in PMI measure). In other words, the co-occurrence count is a number between zero and one based on the distance between the nonverbal element candidate and the light verb.

Table 1. Experimental distribution of the distance between the light verb and preverb in Persian compound verbs

Distance	Probability
1	0.91
2	0.045
3-5	0.025
6-10	0.015
≥ 10	0.005

In PPMI, besides counting every candidate as one, each co-occurrence is a value between zero and one according to the word distance between the light verb and the preverb. The values are gained via estimating an experimental distribution in Bijankhan corpus [35] as shown in Table 1. The reason why we used experimental distribution is that famous distributions such as polynomial distribution were tested via K square test and did not pass the test successfully. By this, all counts in PMI change. Furthermore, to increase precision, we only consider nonverbal candidates that were adjacent to the light verb. For example, consider the candidates in sentence (1). The co-occurrence counts in the classic PMI, the PMI measure used in this paper and PPMI is shown in Table 2.

Table 2. An example of co-occurrence count in PMI and PPMI measures

<i>Preverb</i>	<i>Light Verb</i>	<i>Distance</i>	<i>Co-occurrence count (in classic PMI)</i>	<i>Co-occurrence count (in PMI used in this paper)</i>	<i>Co-occurrence count (in PPMI)</i>
<i>Candidate</i>					
sohbat	kardan	9	1	0	0.005
jang	kardan	4	1	0	0.025
irân	kardan	3	1	0	0.045
erâq	kardan	1	1	1	0.91

4.2 Bootstrapping

Besides deciding for all compound verb candidates at once, bootstrapping was done incrementally up to a threshold. By this, in each iteration, some compound verbs with high PMI scores³ are assumed to be real compound verbs. In the first phase, the compound verb list is empty. Based on the association measure (in here, PMI), some candidate with high score are inserted to the list. After choosing the candidates as compound verbs, the corpus is reprocessed with the assumption that the chosen candidates were truly compound verbs. In the next iteration, the corpus is processed based on the assumption that the compound verb list is real. In each iteration, the compound verb list becomes bigger. After some iterations (based on a threshold decided manually), the method uses PPMI in order to find verbs that are not adjacent. The algorithm is described in Fig. 1. For example, consider sentence (1) again. Assume that the program is in the first iteration, the data format is as in Table 3.

Table 3. Data format for bootstrapping in sentence 1 in the first iteration

<i>Preverb Candidate</i>	<i>Light Verb</i>	<i>Distance</i>
sohbat	kardan	9
jang	kardan	4
irân	kardan	3
erâq	kardan	1

Now assume that in the first iteration ("sohbat kardan") is recognized as a compound verb. Then the corpus reshapes as in Table 4 in which the other candidates lose chance as being a compound verb candidate in sentence 1.

Table 4. Data format for bootstrapping in sentence 1 after recognizing "sohbat kardan" as a compound verb

<i>Preverb Candidate</i>	<i>Light Verb</i>	<i>Distance</i>
Jang	sohbatkardan	4
Irân	sohbatkardan	3
Erâq	sohbatkardan	1

³ The PMI is measured only for candidates that the preverb is adjacent to the light verb.

For the first iterations only PMI in which adjacent words are regarded as concurrence count is bootstrapped. After a threshold, PPMI is considered in order to catch compound verbs that are not frequently adjacent.

Bootstrapping Algorithm for verb identification

- (1) Compound-verb-list=empty
 - (2) Construct training data from corpus
 - (3) Calculate PMIs
 - (4) While (highest-PMI \geq threshold)
 - (5) Select k candidates with highest PMIs and add them to the compound-verb-list
 - (6) Reconstruct the training data based on the compound-verb-list
 - (7) Recalculate PMIs
 - (8) End-while
-

Fig. 1. A summary of bootstapping algorithm for compound verb identification

4.3 Kmeans

In order to apply kmeans, four different features are combined. In order to find the best feature set, multiple experiments were done on several combinations of the features. These features are: 1) PMI (only for adjacent compound verbs), 2) PPMI, 3) Average distance between light verb and preverb, and 4) Average number of nouns between the light verb and the preverb. Euclidian distance [44] is used as the distance measure in this algorithm. All of the features were normalized as a number between zero and one. The number of iterations in the Kmeans algorithm is determined manually. Each cluster was labeled after the algorithm was finished. The label of the cluster was assigned manually by counting whether the compound verbs are in majority or non-compound verbs. For example, if there is a cluster of 250 candidates and 140 candidates are compound verbs, we manually tagged the cluster as the compound verb cluster. We tested several numbers of Ks in order to find the best number of clusters, but the surprising fact was the equal results in all cases of cluster numbers (at last, we chose k as 2).

5 Experiments and Results

We used a finite State automata (FSA) based method to parse Persian light verbs. The multiword capability of a light verb in Persian has some state transition rules that these state transition rules in the automata are extracted from Persian grammar books. The entire Bijankhan corpus is analyzed and verb inflections are simplified to their lemmas. In addition, unrelated candidates are filtered via simple formal rules in Persian compound verbs, e.g. before the postposition (known as "râ" in Persian) there can not occur a preverb or no premodifier is accepted for a preverb [43]. Furthermore, we used morphosyntactic features in Bijankhan corpus [35] to find noun lemmas. To test our approach, we tried to find compound verbs in which their light verb is an inflection of "kardan". The compound verbs with this light verb mostly occur with nouns as

preverbs and do not occur with prepositional preverbs (e.g. "Preposition + Noun" as preverb). As a result, we considered only nouns for the sake of simplicity. We used only candidates that have the co-occurrence counts more than or equal to five as in [4]. We examined both the method used in [17] and the PMI with the best threshold as the baselines. The thresholds for PMI and PPMI in bootstrapping are selected manually. In Kmeans method, 8 different combinations are used in order to gain the best feature set in identification task. The results show that bootstrapping has gained the best results. The feature set used in each kmeans method is shown in Table 5.

Table 5. The features used in kmeans (+: features used, -: features not used)

Method abbreviation	Average distance between LVC	Average Number of nouns between LVC	PPMI	PMI
Kmeans (1)	+	+	+	+
Kmeans (2)	×	+	+	+
Kmeans (3)	+	+	×	+
Kmeans (4)	+	+	+	×
Kmeans (5)	+	×	+	+
Kmeans (6)	×	×	+	+
Kmeans (7)	+	×	×	+
Kmeans (8)	×	+	×	+

As stated before, the method in [17] and the method based on PMI by choosing the best threshold are used as the baseline. The best threshold is chosen based on F-core. The results are shown in Table 6.

Table 6. Results on Persian compound verb identification

Method	Prec.	Rec.	F-Score
<i>Lin</i>	45.15	29.06	35.36
<i>PMI with the best threshold</i>	47.36	32.09	38.26
<i>Bootstrapping on PMI</i>	90.17	68.90	78.11
<i>Bootstrapping on PMI and PPMI sequentially</i>	87.77	71.33	78.70
<i>Kmeans (1)</i>	77.79	59.96	67.72
<i>Kmeans (2)</i>	79.82	61.36	69.38
<i>Kmeans (3)</i>	81.60	62.16	70.57
<i>Kmeans (4)</i>	71.80	65.77	68.65
<i>Kmeans (5)</i>	74.72	60.36	66.78
<i>Kmeans (6)</i>	75.64	59.36	66.52
<i>Kmeans (7)</i>	82.51	50.05	62.31
<i>Kmeans (8)</i>	79.52	66.07	72.17

6 Discussion and Analysis

As seen in Table 6, two features are more informative than the others in our evaluation. The first is the PMI measure that is not appropriate alone. The second one is the

average number of nouns between the light verb and its preverb. On the other hand, using the combination of PMI and average word distance between LVC parts, although the recall is significantly lower than other combinations, the algorithm has gained the best precision. The experiments show that with bootstrapping only on the PMI of adjacent LVCs the results outperform others. Even though via bootstrapping on PPMI, we obtained better F score, the difference is not statistically significant. The key in the success in the results of bootstrapping is the filtering of irrelevant candidates iteratively (as shown in Table 3 and Table 4). The trends in bootstrapping iterations are shown in Fig. 2.

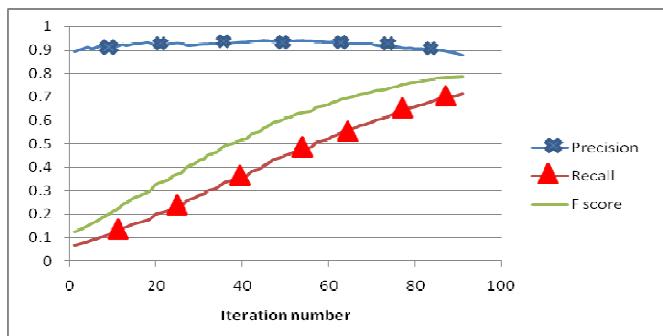


Fig. 2. Trends in bootstrapping

7 Conclusion

In this paper, it is shown that successful association measures such as PMI are not sufficient for Persian compound verb identification. This phenomenon is due to the data sparsity that is caused by word distance flexibility in Persian LVC parts. As stated in previous sections, about 9 percent of LVCs are occurred in the Bijankhan corpus nonadjacent and 2 percent of LVCs do not even occur once adjacent to each other. Even the introduced modification to PMI, i.e. PPMI, does not obtain better results. We gained significantly better results via bootstrapping, i.e. by filtering irrelevant candidates iteratively. By using Kmeans, it is observed that the average number of nouns between the candidate preverb and light verb effects on the identification performance.

8 Future Works

One other important aspect in Persian language processing is the identification Persian syntactic verb valencies (or subcategorization frames) and Persian semantic verb valencies (or argument structure) or semantically clustering verbs regarding polysemy in verbs. This work was done for producing a potential list of Persian verbs to manually select true verbs and assign appropriate verb valencies. After checking obtained verbs by hand, we are going to do other unsupervised (and perhaps semisupervised) tasks to identify syntactic and semantic valencies of Persian verbs.

Acknowledgment. I would like to thank Ali Hadian, Manouchehr Kouhestani, Maryam Aminian, Amir Saeed Moloodi and Majid Laali and four anonymous reviewers for their usefull comments on this work. This Paper is funded by Computer Research Center of Islamic Sciences (C.R.C.I.S).

References

1. Smadja, F.: Retrieving collocations from text: Xtract. *Computational Linguistics* 19(1), 143–177 (1993)
2. Choueka, Y., Klein, T., Neuwitz, E.: Automatic retrieval of frequent idiomatic and collocational expressions in a large corpus. *Journal for Literary and Linguistic Computing* 4(1), 34–38 (1983)
3. Evert, S.: Corpora and collocations. In: *Corpus Linguistics. An International Handbook*, pp. 1212–1248 (2009)
4. Pecina, P.: Lexical association measures and collocation extraction. *Language Resources and Evaluation* 44(1), 137–158 (2010)
5. Diab, M.T., Bhutada, P.: Verb noun construction MWE token supervised classification. In: *Workshop on Multiword Expressions (ACL-IJCNLP 2009)*, pp. 17–22. Association for Computational Linguistics, Suntec (2009)
6. Bannard, C., Baldwin, T., Lascarides, A.: A statistical approach to the semantics of verb-particles. In: *ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pp. 65–72. Association for Computational Linguistics (2003)
7. Diab, M.T., Krishna, M.: Unsupervised Classification of Verb Noun Multi-word Expression Tokens. In: Gelbukh, A. (ed.) *CICLING 2009. LNCS*, vol. 5449, pp. 98–110. Springer, Heidelberg (2009)
8. Fazly, A., Stevenson, S.: Distinguishing subtypes of multiword expressions using linguistically-motivated statistical measures. In: *Workshop on A Broader Perspective on Multiword Expressions*. Association for Computational Linguistics, Prague (2007)
9. Sag, I., et al.: Multiword expressions: A pain in the neck for NLP. In: *6th Conference on Natural Language Learning (COLING 2002)*, pp. 1–15 (2002)
10. Villavicencio, A., Copestake, A.: On the nature of idioms. In: *LinGO Working* (2002)
11. Fazly, A., Cook, P., Stevenson, S.: Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics* 35(1), 61–103 (2009)
12. Villavicencio, A., Copestake, A.: Verb-particle constructions in a computational grammar of English. Citeseer (2002)
13. Karimi-Doostan, G.: Light verbs and structural case. *Lingua* 115(12), 1737–1756 (2005)
14. Fazly, A., Stevenson, S., North, R.: Automatically learning semantic knowledge about multiword predicates. *Language Resources and Evaluation* 41(1), 61–89 (2007)
15. Karimi-Doostan, G.: Event structure of verbal nouns and light verbs. In: *Aspects of Iranian Linguistics: Papers in Honor of Mohammad Reza Bateni*, pp. 209–226 (2008)
16. Fazly, A., Nematzadeh, A., Stevenson, S.: Acquiring Multiword Verbs: The Role of Statistical Evidence. In: *31st Annual Conference of the Cognitive Science Society*, Amsterdam, The Netherlands, pp. 1222–1227 (2009)
17. Lin, D.: Automatic identification of non-compositional phrases. In: *37th Annual Meeting of Association for Computational Linguistics*, pp. 317–324. Association for Computational Linguistics, College Park (1999)
18. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19(1), 61–74 (1993)

19. Pecina, P.: An extensive empirical study of collocation extraction methods. In: ACL Student Research Workshop. Association for Computational Linguistics (2005)
20. Hoang, H.H., Kim, S.N., Kan, M.-Y.: A re-examination of lexical association measures. In: Workshop on Multiword Expressions (ACL-IJCNLP 2009), pp. 31–39. Association for Computational Linguistics, Suntec (2009)
21. Krenn, B., Evert, S.: Can we do better than frequency? A case study on extracting PP-verb collocations. In: ACL Workshop on Collocations. Citeseer (2001)
22. Bu, F., Zhu, X., Li, M.: Measuring the non-compositionality of multiword expressions. In: 23rd International Conference on Computational Linguistics (Coling 2010). Association for Computational Linguistics, Beijing (2010)
23. Blaheta, D., Johnson, M.: Unsupervised learning of multi-word verbs. In: 39th Annual Meeting and 10th Conference of the European Chapter of the Association for Computational Linguistics (ACL 39), Toulouse, France (2001)
24. Baldwin, T., Villavicencio, A.: Extracting the unextractable: A case study on verb-particles. In: 6th Conference on Natural Language Learning (COLING 2002). Association for Computational Linguistics, Stroudsburg (2002)
25. Birke, J., Sarkar, A.: A clustering approach for the nearly unsupervised recognition of non-literal language. In: EACL 2006, pp. 329–336 (2006)
26. Katz, G., Giesbrecht, E.: Automatic identification of non-compositional multi-word expressions using latent semantic analysis. In: Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties. Association for Computational Linguistics, Sydney (2006)
27. Cook, P., Fazly, A., Stevenson, S.: Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In: Workshop on A Broader Perspective on Multiword Expressions. Association for Computational Linguistics, Prague (2007)
28. Fazly, A., Stevenson, S.: Automatically constructing a lexicon of verb phrase idiomatic combinations. In: EACL 2006 (2006)
29. Bannard, C.: A measure of syntactic flexibility for automatically identifying multiword expressions in corpora. In: Workshop on A Broader Perspective on Multiword Expressions. Association for Computational Linguistics, Prague (2007)
30. Cook, P., Fazly, A., Stevenson, S.: The VNC-Tokens Dataset. In: LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008), pp. 19–22 (2008)
31. Diab, M.T., Krishna, M.: Handling sparsity for verb noun MWE token classification. In: Workshop on Geometrical Models of Natural Language Semantics. Association for Computational Linguistics, Athens (2009)
32. Pecina, P.: A machine learning approach to multiword expression extraction. In: Shared Task for Multiword Expressions (MWE 2008), pp. 54–57 (2008)
33. Kaalep, H.-J., Muischnek, K.: Multi-word verbs of Estonian: a database and a corpus. In: LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008), pp. 23–26 (2008)
34. BöMOVÁ, A., et al.: The Prague Dependency Treebank: A three-level annotation scenario. Treebanks: Building and Using Parsed Corpora, 103–127 (2003)
35. Bijankhan, M.: The role of the corpus in writing a grammar: An introduction to a software. Iranian Journal of Linguistics 19(2) (2004)
36. Fazly, A.: Automatic acquisition of lexical knowledge about multiword predicates. Citeseer (2007)
37. Dabir-Moghaddam, M.: Compound verbs in Persian. Studies in the Linguistic Sciences 27(2), 25–59 (1997)

38. Family, N.: Explorations of Semantic Space: The Case of Light Verb Constructions in Persian. In: Ecole des Hautes Etudes en Sciences Sociales, Paris, France (2006)
39. Pantcheva, M.: First Phase Syntax of Persian Complex Predicates: Argument Structure and Telicity. *Journal of South Asian Linguistics* 2(1) (2010)
40. Müller, S.: Persian complex predicates and the limits of inheritance-based analyses. *Journal of Linguistics* 46(03), 601–655 (2010)
41. Karimi Doostan, G.: Separability of light verb constructions in Persian. *Studia Linguistica* 65(1), 70–95 (2011)
42. Ghomeshi, J.: Non-projecting nouns and the ezafe: construction in Persian. *Natural Language & Linguistic Theory* 15(4), 729–788 (1997)
43. Anvari, H., Ahmadi-Givi, H.: Persian grammar 2, 2nd edn. Fatemi, Tehran (2006)
44. Deza, E., Deza, M.M.: Encyclopedia of Distances. Springer, Heidelberg (2009)

Testing a Theory of Perceptual Mapping Using Robots

Md. Zulfikar Hossain, Wai Yeap, and Olaf Diegel

Auckland University of Technology, Auckland, New Zealand
mhossain@aut.ac.nz

Abstract. This paper describes the implementation of a new approach to mapping using a mobile robot that is based upon a theory of human perceptual mapping. Its key feature is that it does not need to continuously track the robot's position as the robot moves through the environment and it does not perform error correction due to robot sensors. It still manages to produce an approximate map of the environment that is adequate for orienting oneself and for knowing where things are located. Arguably, it is the kind of map that humans seem to have.

Keywords: Perceptual map, mobile robots, spatial layout of the environment.

1 Introduction

A perceptual map is a representation of one's immediate environment that one creates as one moves about in the environment. With it, one could immediately point to things out of sight, or know one's current position in the environment. In short, it is a description of the spatial layout of things in one's surrounding, using a global co-ordinate system centered either on the self or an arbitrary part of the environment. As the perceptual map is intended to describe the spatial layout of one's *immediate* surroundings, it is not intended as a representation for one's total experience.

Nonetheless, robotics researchers have developed algorithms for computing a perceptual map for the entire environment explored by the robot and these environments can still be quite large. Their algorithms are probabilistic in nature and their aim is to allow the robots to simultaneously localize and map the environment (for example, see [1] and for an introduction, see [2]). Hence, they refer to their problem as a SLAM problem. Their goal is to create an accurate perceptual map.

The "maps" of humans (and animals) have also been much studied in the psychological sciences [3-6]. Humans' maps are known to be much more complex since we also build a conceptual view of our world. Different physical spaces are grouped together into a hierarchy of places; each place serving a purpose. Some objects are singled out and remembered as landmarks while some are forgotten. Humans cannot remember precisely where things are but many still have a good sense of direction, even after a brief exposure to the environment [7]. Psychologists have argued that humans first compute a perceptual map prior to enriching it with other details to become a cognitive map (e.g. [8-10]. Computational models of cognitive maps (e.g. [11-13]) often suggest the presence of a perceptual map.

However, how such a map is computed is little understood. Many have assumed that it would be computed using an approach similar to that studied by robotics researchers. That is, successive views are integrated to form a single unified map. As Yeap [14] pointed out, such an approach fails to account for much of what is known about human cognitive maps. For example, and as robotics research has shown, to use such a method, the map produced must be as accurate as possible. If not, the map will quickly be rendered useless. It is unlikely that humans compute a precise metric map in their head. The idea that we integrate successive views to form a unified map is also not supported by studies on saccade eye movements. For the latter problem, researchers were investigating how we perceive a stable world given that our eyes make rapid movements to focus on different parts of the retina. It was discovered that humans fail to notice significant changes made between saccades, a phenomenon which is now recognized as change blindness [15-17].

Yeap [14] proposed a new algorithm for computing human-kind of maps. Unlike SLAM, it does not emphasize on computing a precise map and it does not need to integrate successive views. The key idea of his approach is summarized in section 2. In section 3, we describe an implementation of the theory on a mobile robot, and in section 4, we show the results obtained. The algorithm was tested on two different environments and the maps produced successfully capture the layout of the environments tested. Section 5 concludes the paper with a summary of the results obtained and a discussion for future directions.

2 A Theory of Perceptual Mapping

In developing his theory of perceptual mapping, Yeap [14] made two observations. First, he observed that a view affords us more than a description of the surfaces in front of us. It tells us what and where things are, where we can move to next, what events are unfolding, where there might be dangers, and others [18]. In short, a view is in fact a significant representation of a local environment and it should be made explicit in the map as a description of a local environment rather than as some spatially organized surfaces. Second, he observed that the world we live in is relatively stable. That is, it does not change much when we blink our eyes or take a few steps forward. As such, there is no immediate need to update the view in our perceptual map as we move. For example, consider your first view of a corridor when entering it and assume an exit can be seen at the other end. If you walk down this corridor to the exit, then the description of the corridor space afforded in the first view adequately describes the local environment you were going through. Updating this description to include, for example, a view of a room besides the corridor as you walk past it will enrich the description, but is unnecessary if the room is not entered.

Combining both observations, Yeap [14] suggested that one's perceptual map is computed by integrating views only when one is about to move out of the local environment afforded in an earlier view. This immediately poses two problems: if we do not update our map as we move, how do we know where we are in our map and more importantly, how do we update the map with the next view when the need arises? Yeap observed that humans recognize where they are by recognizing familiar objects in their environment. Again consider the corridor example above and

assuming you are half way down the corridor. How do you then know where you are in the map? If you recognize the exit at the end of the corridor to be the same exit in the initial view, then, using triangulation, you can locate your approximate position in the map. Thus, one possible solution is to keep track of objects seen in the remembered view in the current view. These objects are referred to as reference targets. If some could be found, one could triangulate one's position in the map and thus localize oneself. However, at some points, one will not be able to do so and this is when one needs to expand the map to include a new view (albeit, a new local environment). These points are known as limiting points. If the new view to be added is selected at a point just before reaching a limiting point, it could be added to the map using the same method of triangulation.

In summary, the theory specifies that what is made explicit in a perceptual map is an integrated global representation of views selected during a journey. This is because each of these views provides an adequate description of the spatial layout of the local environment experienced. The basic algorithm for implementing the theory involves recognising objects in the current view that were remembered in the perceptual map, and using them to triangulate position of unknown objects (including the self) in the map.

3 Implementations and Results

We implemented the theory using a mobile robot (a Pioneer 3DX from MobileRobots Inc. and equipped with a SICK laser rangefinder) mapping two different indoor environments as shown in Figure 1. In Figure 1a, the route taken was about 100m long and in Figure 1b, the route was about 170m long. The theory leaves open three implementation issues, namely how and what reference targets are selected, when a new view should be added to the perceptual map, and how much information in each view is combined with what is in the map.

Implementing the theory on a laser-ranging mobile robot means that our choice of a reference target is limited to perceived surfaces in view, each being represented as a 2D line. Since these lines are recovered from laser points, their shape varies between views depending on the robot vantage point. Consequently, we choose reference targets with a minimum length of 40cm, which has at least an occluding edge. The latter ensures a reference point exists on the surface for relative positioning of new surfaces into the map. A minimum of two such points is needed.

Since the robot cannot recognize these reference targets directly from one view to the next, their recognition has to be "simulated" or done indirectly. For our implementation, we use the transformation method. Transforming between two successive views, the robot can predict where these reference targets will be. They are then "recognized" via the use of some heuristics such as the proximity of two surfaces, the sudden appearance of a new surface in front of another, and others. It is emphasized that the transformation method is used here because of the lack of any recognition ability of our robot. It is not part of the theory.

According to the theory, a new view is added to the perceptual map when the robot reaches a limiting point in the current space. A limiting point is defined as a view that contains less than two reference targets in view. However, note that if the robot

reaches a limiting point as defined here, it would not be able to update its map due to insufficient points to triangulate its position in the map. As such, when the robot reaches a limiting point, it will update its map using its *previous* view.

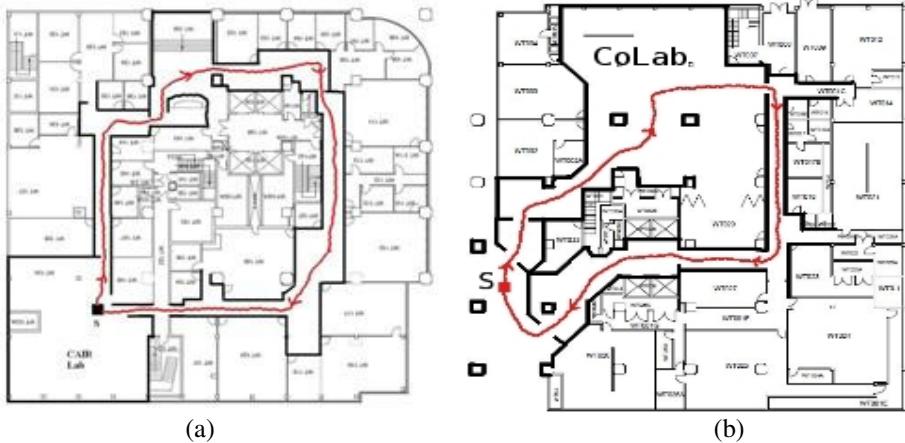


Fig. 1. Two test environments

The algorithm implemented can now be described formally as follows. Let PM be the perceptual map, V_0 be one's initial view, and R be some reference targets identified in V_0 . Initialize PM with V_0 . For each move through the environment, do:

1. Execute move instruction and get a new view, V_n .
2. If it is a turn instruction, use V_n to expand PM and create a new R. Go to step 1.
3. Search for the reference targets in V_n by transforming V_{n-1} to V_n using the mathematical transformation approach. [recognition]
4. If two or more targets are found, remove unseen targets from R. Go to step 1. [tracking]
5. Add V_{n-1} to PM. This is achieved by first triangulating the robot's position and orientation in PM using the available reference points. What is seen in V_{n-1} is used to replace what is in front of the robot. [expanding]
6. Create a new set of R from V_n . Go to step 1.

Note that when expanding the map, any surfaces in the map that were entered in the last update and are also found in V_{n-1} are removed. There is no need to “update” the description of such surfaces for two reasons. Firstly, the exact details of individual surfaces are unimportant. Secondly, it is always assumed that the current view provides a better description of what is remembered. An exception, however, is that remembered surfaces which are partially in view are not removed. For such surfaces, their descriptions need to be updated.

4 Results

Figure 2b shows that the perceptual map is not always updated as the robot explores the environment using the new algorithm. Figure 2a shows three consecutive steps of the robot. During step 2, the map is not updated.

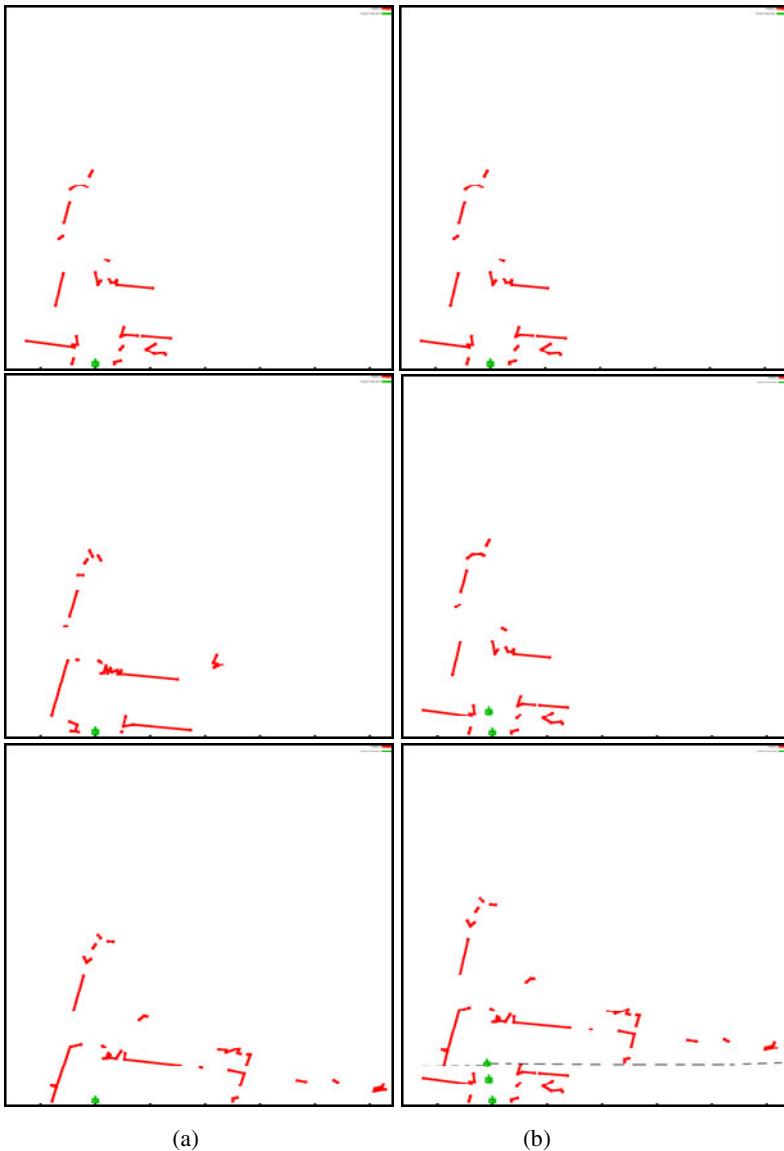


Fig. 2. Updating of the perceptual map as the robot explores the environment

Figure 3 shows two perceptual maps produced as the robot traversed the path through the environment as shown in Figure 1a. We have tested the robot with several different start locations and all the maps produced show a good layout of the environment.

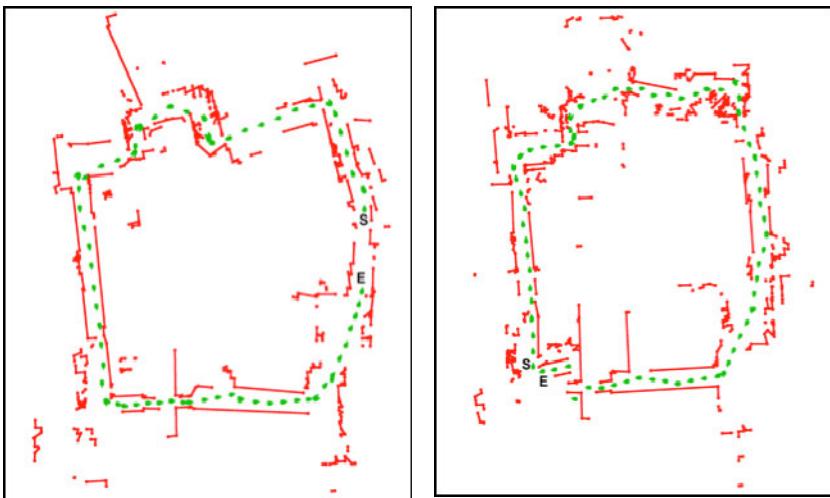


Fig. 3. Perceptual maps produced when traveling in an anti-clockwise (left) and a clockwise (right) direction

Figure 4 shows two perceptual maps produced as the robot traversed the path through the environment shown in Figure 1b. This turns out to be an interesting environment. The robot did not detect any features as it moved into the straight path on the right of Figure 1b. This is a long empty corridor. Consequently, Figure 4a shows the robot fails to compute a map for that part of the environment and is deemed to be lost.

It is interesting to note that humans often are disoriented in featureless environments (such as walking down a spiral corridor) and in such situations, one has to use additional higher reasoning to cope with the situations. The algorithm is thus modified so that it extends the surfaces on both sides of the view to the distance moved if no reference targets are perceived and until new reference targets are found again. Figure 4b now shows the robot successfully mapped the environment.

It is also interesting to note that when developing the algorithm in general, we also observed how the use of a single view to describe one's local environment provides stable descriptions of local environments. This is in contrast to the map produced using the traditional transformation approach. The latter keeps changing due to error correction. For example, even though one sees a straight corridor from its entrance, one could end up with a curved corridor at the end because of errors. With our approach, one sees and remembers a straight corridor as the local environment that one is entering into and that memory did not change as one move in it. Psychologists refer to this phenomenon as our perception of a stable world.

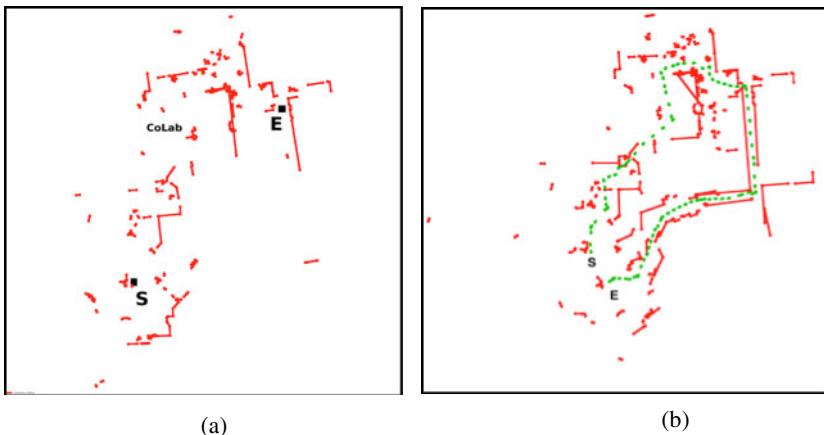


Fig. 4. Perceptual maps produced when exploring the environment as shown in Figure 1b

5 Conclusion

This paper shows the implementation of a new theory of perceptual mapping using a laser-ranging mobile robot. The theory was developed to explain how humans and animals compute their perceptual map. The results showed that despite no error correction, a useful layout of the environment is easily computed if good reference targets are made available. However, in one of the environments, the lack of good reference targets caused a failure to map the environment. This demonstrates that the algorithm, like humans, is not robust and one could get lost. In situation like this, one needs to exploit other forms of knowledge to solve the problem.

Future work will focus on developing more robust algorithms based upon this new theory of perceptual mapping for robot mapping, and designing new experiments on human and animal cognitive mapping processes to test the validity of the theory as a theory of human/animal perceptual mapping.

References

1. Thrun, S., Burgard, W., Fox, D.: *Probabilistic Robotics*. MIT Press, Cambridge (2005)
2. Thrun, S.: Simultaneous localization and mapping. In: Jefferies, M.E., Yeap, W.K. (eds.) *Robotics and Cognitive Approaches to Spatial Mapping*, pp. 13–42. Springer, Heidelberg (2008)
3. Downs, R.M., Stea, D.: *Image and Environment: Cognitive Mapping and Spatial Behavior*. Aldine, Chicago (1973)
4. O’Keefe, J., Nadel, L.: *The Hippocampus as a Cognitive Map*. Oxford University Press, Oxford (1978)
5. Tolman, E.C.: Cognitive maps in rats and men. *Psych. Rev.* 55(4), 189–208 (1948)
6. Golledge, R.G.: *Wayfinding Behavior: Cognitive Mapping and other Spatial Processes*. The Johns Hopkins University Press, Baltimore (1999)

7. Ishikawa, T., Montello, D.R.: Spatial knowledge acquisition from direct experience in the environment: Individual differences in the development of metric knowledge and the integration of separately learned places. *Cog. Psych.* 52(2), 93–129 (2006)
8. Burgess, N.: Spatial memory: how egocentric and allocentric combine. *Trends in Cog. Sci.* 10(12), 551–557 (2006)
9. Mou, W., McNamara, T.P., Valiquette, C.M., Rump, B.: Allocentric and Egocentric Updating of Spatial Memories. *J. Expt. Psych.: Learning, Memory, and Cognition* 30(1), 142–157 (2004)
10. Glennerster, A., Hansard, M.E., Fitzgibbon, A.W.: View-Based Approaches to Spatial Representation in Human Vision. In: Cremers, D., Rosenhahn, B., Yuille, A.L., Schmidt, F.R. (eds.) *Statistical and Geometrical Approaches to Visual Motion Analysis*. LNCS, vol. 5604, pp. 193–208. Springer, Heidelberg (2009)
11. Kuipers, B.J.: The Spatial Semantic Hierarchy. *Artificial Intelligence* 19, 191–233 (2000)
12. Yeap, W.K.: Towards a Computational Theory of Cognitive Maps. *Artificial Intelligence* 34(3), 297–360 (1988)
13. Yeap, W.K., Jefferies, M.E.: Computing a Representation of the Local Environment. *Artificial Intelligence* 107(2), 265–301 (1999)
14. Yeap, W.K.: A computational theory of humans perceptual mapping. In: 33rd Annual Conference of the Cognitive Science Society, pp. 429–434 (2011)
15. Snowden, R.J.: Visual perception: Here's mud in your mind's eye. *Current Biology* 9(9), R336–R337 (1999)
16. Irwin, D.E., Zelinsky, G.J.: Eye movements and scene perception: memory for things observed. *Perception & Psychophysics* 64(6), 882–895 (2002)
17. Simons, D.J., Rensink, R.A.: Change blindness: past, present, and future. *Trends in Cognitive Sciences* 9(1), 16–20 (2005)
18. Gibson, J.J.: *The perception of the visual world*. Houghton Mifflin, Boston (1950)

A POMDP Model for Guiding Taxi Cruising in a Congested Urban City

Lucas Agussurja¹ and Hoong Chuin Lau²

¹ The Logistics Institute Asia Pacific, National University of Singapore
21 Heng Mui Keng Terrace #04-01, Singapore 119613
tlila@nus.edu.sg

² School of Information Systems, Singapore Management University
80 Stamford Road, Singapore 178902
hclau@smu.edu.sg

Abstract. We consider a partially observable Markov decision process (POMDP) model for improving a taxi agent cruising decision in a congested urban city. Using real-world data provided by a large taxi company in Singapore as a guide, we derive the state transition function of the POMDP. Specifically, we model the cruising behavior of the drivers as continuous-time Markov chains. We then apply dynamic programming algorithm for finding the optimal policy of the driver agent. Using a simulation, we show that this policy is significantly better than a greedy policy in congested road network.

Keywords: agent application, intelligent transportation, POMDP, taxi service.

1 Introduction

Taxis are a major mode of transport in every urban city in the world. There were about 24,000 taxis in Singapore, and 87,000 licensed taxi drivers, and taxis provide about 850,000 trips daily, as of April 2009. In Singapore, taxis are operated by a small number of taxi companies. The largest taxi company, ComfortDelGro, operates over 15,000 taxis, and captures about 63% of the market share. Like many congested cities in the world, people in Singapore often view taxis as a more efficient mode of transportation compared to private cars. Statistics released by the Singapore Land Transport Authority shows that a taxi travels some 120,000 km a year, and more than a third of such travel is empty cruising, which represents a wastage of resources.

Although research related to taxi services can be found in the economics, transportation, and operation research literature, studies from the AI point of view, however, has been lacking to our knowledge. This is an opportunity for applied AI research, since the system resulting from the collective behavior of the drivers can be readily modeled as a multi-agent system, with drivers acting as rational agents towards maximizing their individual utilities. We believe that the public transport arena offers a rich domain for application of AI concepts and methodologies.

In this paper, we consider the specific problem of modeling and guiding taxi driver cruising behavior in a congested city consisting of a large number of taxis and passengers. This work is made possible by a large dataset obtained from a large taxi operator. We model, for simplicity, the network as a graph where nodes represent zones while arcs represent adjacency. Passengers “arrive” in each zone as a Poisson process, while taxis are either in the state of **occupied** or **cruising** (when no passenger is onboard and the driver is actively seeking passengers, whether moving about or resting at a location). We are concerned with improving the overall utilization rate by offering intelligent guidance on its cruising behavior (i.e. deciding when and where to move from zone to zone considering the passenger arrival processes as well as in response to the behavior of the other drivers).

We propose a POMDP to model decision process of a taxi driver during cruising. Our purpose in this paper is not to invent a new algorithm for solving POMDP. Rather, motivated by the vast recent literature in AI on solving large-scale POMDPs (see Related Works below), we choose POMDP as the modeling framework for our problem, and present in this paper how such model is built, and experimental results obtained. The proposed model can be embedded into a vehicular device that guides the taxi driver to move intelligently within a highly dynamic environment (such as in a congested city) that will maximize occupancy and hence revenue.

2 Preliminaries

Our system consists of:

1. A directed graph $G = (V, E)$, representing the road network on which the taxis operate. Each node in the graph is associated with a zone. A taxi may move from one zone to the other following the edge linking the zones. We assume that the graph is connected.
2. A set of n drivers indexed by $i \in \{1, \dots, n\}$. The cruising behavior of a driver governs its preference in choosing which zone to cruise in. In this paper, we assume that each driver cruising behavior is independent of the others.
3. The agent driver under consideration.

The objective in this paper is to compute the best cruising policy (response) of the agent, one that maximizes the time that the agent is occupied, given the cruising behavior of the other drivers and a finite period (the planning horizon). We make the assumption that all the drivers and the agent are always in operation during the planning horizon. The drivers and agent constantly transit from the cruising (the term is used interchangeably with “unoccupied”) state to occupied state and back to cruising state.

A partially observable Markov decision process for a single agent can be described as a tuple $\langle S, A, T, R, \Omega, O \rangle$, where:

- S is a finite set of states of the system,
- A is a finite set of actions of the agent,

- $T : S \times A \rightarrow \Delta(S)$ is the *state transition function*, where we write $T(s, a, s')$ for the probability of ending in state s' , given that the agent is in state s and takes action a ,
- $R : S \times A \rightarrow \mathbb{R}$ is the *reward function*, where $R(s, a)$ is the expected reward for taking action a while in state s ,
- Ω is a finite set of possible observations the agent can receive,
- $O : S \times A \rightarrow \Delta(\Omega)$ is the *observation function*, where we write $O(s, a, o)$ for the probability of making observation o after the agent took action a in the current state s .

The agent maintains an internal *belief state* $b \in \Delta(S)$ that described its belief of the current state of the system based on its previous observations, actions and the initial belief state b^0 . A t -step policy tree is a perfect $|\Omega|$ -ary tree of depth t describing completely the agent's possible actions for the next t steps. Each node in the tree specifies an action with the root as the starting action, and each branch specifies a possible observation and the corresponding child node specifies the next action to be taken.

3 Related Works

As an integral part of public transportation system of an urban city, taxi services have been extensively studied from the economics (see [11,14,12]), transportation and operation research (see [13,10]) point of views. In the economics literature, research has been conducted to gain insights into the nature of the demand and supply of taxi services, their interaction and the resulting market equilibrium. The economic consequences of regulatory restraints, such as entry restriction and price control, have been examined as well. In operation research literature, quantitative models have been built to capture driver's movement behavior, passenger's searching behavior, and the competitive nature of the drivers.

POMDP is a modeling framework used for agent planning in partially observable stochastic domains [7]. It has been used in AI robotics and planning. The majority of these studies concentrate on finding a scalable algorithm to solve the model (e.g. [3,4,6,8,9]), while some focus on looking at special cases of POMDP for which efficient algorithm might be found. Research into its multi-agent counterpart have become active in recent years, both in competitive setting [5] and non-competitive setting ([1,2]).

4 POMDP Model for Single Cruising Agent

In this section, we present our model for a taxi cruising decision process. First and foremost, it should be noted that two models are proposed for cruising behavior. For the agent driver under consideration, we use a POMDP to model its decision process from which we derive its optimal cruising behavior. The behavior of each of the other drivers in the system is modeled by the continuous time Markov chain (see Section 4.2). In this sense, the behavior of these drivers

are "static" in that they do not change in reaction to the behavior of the agent or the other drivers. A more dynamic model would be the partially observable stochastic game (POSG), where each driver is modeled as a POMDP. Given the current state of research [5], POSG is much more complicated and computationally inhibiting in our context where we are interested to model a large number of drivers. Hence in this work, we focus on a single POMDP. Henceforth, to avoid confusion, we will use the term "driver" to refer specifically to a driver who is not the agent under consideration.

4.1 System States

At any point of time, there are n taxis operating (excluding the agent considered). A state of the system consists of the state of the agent and each of these taxis. Specifically, a state of the system is described by a tuple $s = \langle \omega, L, D, (\delta_i), (l_i), (h_i), (d_i) \rangle$, with $1 \leq i \leq n$, where:

- $\omega = 1$ if the agent's taxi is occupied, and $\omega = 0$ otherwise (i.e. cruising),
- $L \in V$ is the current location zone of the agent,
- $D \in V \cup \{0\}$ is the destination zone of the passenger if the agent's taxi is occupied, $D = 0$ otherwise,

and for each driver i :

- $\delta_i = 1$ if the taxi is occupied, and $\delta_i = 0$ if it is currently cruising,
- $l_i \in V$ is the current location zone of the driver,
- h_i contains the path history of the driver if it is occupied, that is, the sequence of zones starting with the zone from which the passenger is found to the current location zone of the driver. If the taxi is unoccupied, h_i is empty.
- $d_i \in V \cup \{0\}$ if the destination zone of the passenger if the taxi is occupied, $d_i = 0$ otherwise.

To simplify notations, we will omit the subscripts of a variable when we consider the vector value of the corresponding variable. For example, $l = (l_1, \dots, l_n)$, and $d = (d_1, \dots, d_n)$.

4.2 A Model for Driver Cruising Behavior

We model the driver cruising behavior as a continuous time Markov chain, which consists of a set of states where the transition time from one state to another is exponentially distributed. The legitimacy of exponential distribution has been verified from a real trace of taxi data of a large Singapore taxi operator for one month on the Singapore road network (see Section 6.1).

The set of all possible states of a driver is given by

$\{C_1, \dots, C_{|V|}, O_1, \dots, O_{|V|}\}$, where state C_j ($j \in V$) corresponds to the driver being in zone j and cruising for passengers; while state O_j ($j \in V$) corresponds to the driver being in zone j and currently delivering a passenger to its destination. The following are the sets of parameters which govern the transitions between the states:

- $\{\lambda_{jk}\}_{j,k \in V}$, where λ_{jk} is the rate of transition from state C_j to state C_k . This set of parameters describes the cruising behavior of a driver. In our model, we assume that the drivers are uniform, i.e., have the same cruising behavior.
- $\{\pi_j\}_{j \in V}$, where π_j is the rate of transition from state C_j to state O_j . This set of parameters describes the rate of finding a passenger in a zone, and is dependant upon the number of drivers cruising in the zone (see equation 1).
- $\{\pi'_j\}_{j \in V}$, where π'_j is the rate of transition from state O_j to state C_j . This set of parameters describes the rate of finding the dropoff point of a passenger in its destination zone.
- $\{\rho_{jk}\}_{j,k \in V}$, where ρ_{jk} is the rate of transition from state O_j to state O_k . This set of parameters describes the congestion rate of the road network.

Together, these sets of parameters constitute the Q -matrix of the Markov chain, which describes the chain completely. As mentioned previously, we assume that the drivers operate independently of each other. Each zone $j \in V$ has a constant passenger arrival rate μ_j . When a passenger arrives in a zone, it will be randomly picked up by one of the drivers cruising in the zone. In other words, for every $j \in V$, we have:

$$\pi_j = \frac{\mu_j}{(1 - \omega) \cdot \alpha(L, j) + \sum_{i=1}^n [(1 - \delta_i) \cdot \alpha(l_i, j)]}, \quad (1)$$

as the rate of finding a passenger in zone j , where $\alpha(j, k) = 1$ if $j = k$, and $\alpha(j, k) = 0$ otherwise.

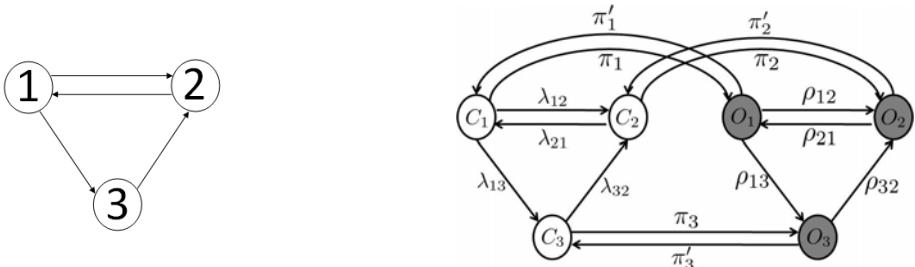


Fig. 1. Example: The figure on the left shows a network of 3 nodes, while the figure on the right shows the corresponding Markov chain describing the behavior of a driver

An example of the Markov chain on a simple 3-node network is given in Figure 1. Two useful quantities for our modeling that can be derived from the Markov chain are: (1) The probability of leaving a state after t time has passed which is given by $1 - \exp\{-(\sum \text{outgoing rate}) \cdot t\}$, and (2) The probability of moving from one state to the other with transition rate λ which is given by: $\lambda / \sum \text{outgoing rate}$. In the example of Figure 1, the probability of leaving the state C_1 after one unit of time has passed is $1 - \exp\{-(\lambda_{12} + \lambda_{13} + \pi_1)\}$, with the probability of going to state O_1 given by $\pi_1 / (\lambda_{12} + \lambda_{13} + \pi_1)$.

4.3 Actions and State Transition Function

Next, we define the actions of the agent and derive the state transition function T for the POMDP. We do this by discretizing the continuous Markov chain into periods of one minutes, where each period corresponds to one step of the POMDP. The state transition function T can be separated into two independent components as follows:

$$\begin{aligned} T(s, a, s') & \quad (2) \\ & = \Pr[\langle \omega', L', D', \delta', l', h', d' \rangle | \langle \omega, L, D, \delta, l, h, d \rangle, a] \\ & = \Pr[\omega', L', D' | \omega, L, D, \delta, l, a] \cdot \Pr[\delta', l', d' | \omega, L, \delta, l, d]. \end{aligned}$$

Consider the first term of the right hand side of Equation 2. The actions available to the agent depend on the state of the agent.

Case 1: $\omega = 0$. In the unoccupied state, the agent may take one of the following actions in a step: (a_1) continue cruising in the current zone, or (a_2) make an attempt to move to an adjacent zone. In the unoccupied state, the agent may have the chance of getting a passenger as well. Given that $\omega = 0$ and the agent takes action a_1 , the following may happen in one step: (1) The agent doesn't find any passenger, or (2) The agent manages to find a passenger with k as the destination zone. In both cases, the location of the agent doesn't change. The following gives the probability of each of these cases respectively:

$$\Pr[0, j, 0 | 0, j, 0, \delta, l, a_1] = e^{-\pi_j},$$

$$\Pr[1, j, k | 0, j, 0, \delta, l, a_1] = (1 - e^{-\pi_j}) \cdot \gamma_{jk},$$

where γ_{jk} is the probability of a passenger's destination zone being in k given that its starting zone is j . On the other hand, if the agent chooses a_2 and make an attempt to move to an adjacent zone z , one of the following may occur: (1) A passenger is found, and the agent stays in the current location, (2) No passenger is found, and the agent manages to move to zone z , or (3) No passenger is found and the agent stays in the current location. The following gives the probability of each of these cases respectively:

$$\Pr[1, j, k | 0, j, 0, \delta, l, a_2 \rightarrow z] = \frac{\pi_j \cdot \gamma_{jk}}{\pi_j + \rho_{jz}} \left(1 - e^{-(\pi_j + \rho_{jz})} \right),$$

$$\Pr[0, z, 0 | 0, j, 0, \delta, l, a_2 \rightarrow z] = \frac{\rho_{jz}}{\pi_j + \rho_{jz}} \left(1 - e^{-(\pi_j + \rho_{jz})} \right),$$

$$\Pr[0, j, 0 | 0, j, 0, \delta, l, a_2 \rightarrow z] = e^{-(\pi_j + \rho_{jz})}.$$

Case 2: $\omega = 1$. In the occupied state, the action of the agent is determined (denoted by a_3). When it is not in the destination zone of the passenger, it will make an attempt to move to an adjacent zone along the shortest path from the current zone to the destination zone. And when it is in the destination zone of

the passenger, it will try to find the dropoff point within the zone. For the former case, the following are its success and failure probabilities respectively:

$$\Pr[1, z, k | 1, j, k, \delta, l, a_3 \rightarrow z] = 1 - e^{-\rho_{jz}},$$

where z is the next zone in the shortest path from j to k , consequently:

$$\Pr[1, j, k | 1, j, k, \delta, l, a_3 \rightarrow z] = e^{-\rho_{jz}}.$$

And for the later case, its success and failure probabilities are given by:

$$\Pr[0, k, 0 | 1, k, k, \delta, l, a_3 \rightarrow k] = 1 - e^{-\pi'_k},$$

and:

$$\Pr[1, k, k | 1, k, k, \delta, l, a_3 \rightarrow k] = e^{-\pi'_k}$$

respectively.

Consider the second term of the right hand side of Equation 2. Assuming independence among the drivers, we have:

$$\begin{aligned} & \Pr[\delta', l', d' | \omega, L, \delta, l, d] \\ &= \prod_{i=1}^n \Pr[\delta'_i, l'_i, d'_i | \delta_i, l_i, d_i, \omega, L, \delta_{-i}, l_{-i}, d_{-i}], \end{aligned}$$

where the $-i$ subscript denotes a vector without the i -th element. Each driver behavior is modeled as an independent Markov chain as described in Section 4.2. In the unoccupied state, in one step of the POMDP, the following may occur: (1) No passenger is found, and the driver continue to cruise in the same zone, (2) No passenger is found, and the driver moves to an adjacent zone, or (3) A passenger is found in the current zone. The following are the probability of each of these events respectively:

$$\Pr[0, j, 0 | 0, j, 0, \dots] = \exp \left\{ - \left(\sum_{z \neq j} \lambda_{jz} + \pi_j \right) \right\},$$

$$\Pr[0, k, 0 | 0, j, 0, \dots] = \frac{\lambda_{jk}}{\sum_{z \neq j} \lambda_{jz} + \pi_j} \left[1 - \exp \left\{ - \left(\sum_{z \neq j} \lambda_{jz} + \pi_j \right) \right\} \right],$$

$$\Pr[1, j, k | 0, j, 0, \dots] = \frac{\pi_j \cdot \gamma_{jk}}{\sum_{z \neq j} \lambda_{jz} + \pi_j} \left[1 - \exp \left\{ - \left(\sum_{z \neq j} \lambda_{jz} + \pi_j \right) \right\} \right].$$

In the occupied state, the behavior of a driver is similar to that of the agent. The driver will attempt to move the the next adjacent zone in the shortest path to the destination zone. Once in the destination zone, the driver will try to find

the exact dropoff point of the passenger. The following probabilities are derived similarly to those of the agent:

$$\Pr[1, j, k | 1, j, k, \dots] = e^{-\rho_{jz}},$$

where z is the next zone in the shortest path from j to k , and:

$$\Pr[1, z, k | 1, j, k, \dots] = 1 - e^{-\rho_{jz}}.$$

Similarly:

$$\Pr[1, k, k | 1, k, k, \dots] = e^{-\pi'_k},$$

$$\Pr[0, k, 0 | 1, k, k, \dots] = 1 - e^{-\pi'_k}.$$

This completes the description for the state transition function of the POMDP. It should be noted that all other transitions that are not defined in this section have zero probability of occurring.

4.4 The Complete Agent POMDP Model

We define the rest of the components of the POMDP here. The agent's reward function R is simply a binary function that return 1 if the agent is in occupied state and 0 in the unoccupied state, i.e., $R(s, a) = 1$ if $\omega = 1$ and $R(s, a) = 0$ otherwise. This models the objective of the agent to be in the occupied state as long as possible. Following the data that we have for the taxi movement in Singapore, the agent may observe the state s of the system in each step of the POMDP, except for the destination zone of the passenger of the occupied drivers, that is, the agent may observe the variables $\omega, L, D, \delta, l, h$ completely, but not the variable d . The observation function, therefore, is given by:

$$O(\langle \omega, L, D, \delta, l, h, d \rangle, a, o) = \sum_{d'} T(\langle \omega, L, D, \delta, l, h, d \rangle, a, \langle o, d' \rangle).$$

This completes the definition of the POMDP model.

5 Solving the Finite Horizon POMDP Model

Next, we present our solution method to compute the optimal T -horizon policy for the POMDP model. We apply the standard dynamic programming algorithm [7] for solving general POMDP. The main difference is the representation of the belief state. Given the specific nature of the agent's observation function, we are able to represent its belief state as a set of trees (see below).

5.1 Agent Belief State

A belief state b is a probability distribution over S . In our case, since the agent observes the state of the system completely except for the destination zone of occupied drivers, its belief state is reduced to a probability distribution over

$\{V \cup \{0\}\}^n$, which is all the possible value for vector d . We let $b(d)$ denote the probability assigned to the destination vector d by belief state b . The agent starts with the initial belief state b^0 , and in each step of the POMDP, it computes a new belief state b' , based on the old belief state b , an action a , and an observation o . The new degree of belief in a destination vector d' , can be computed as follows:

$$\begin{aligned} b'(d') &= \Pr[d'|b, a, o] \\ &= \prod_{i=1}^n \Pr[d'_i|b, a, \langle \omega', L', D', \delta', l', h' \rangle] \\ &= \prod_{i=1}^n \Pr[d'_i|b, \delta'_i, h'_i]. \end{aligned}$$

It is obvious that if $\delta'_i = 0$, then $d'_i = 0$ with probability 1. For the case when $\delta'_1 = 1$, the value of $\Pr[d'_i|b, \delta'_i, h'_i]$ for each i is computed as follows. When there is only one element in the path history h'_i , i.e. $h'_i = (j)$, then for all $k \in V$, $\Pr[k|b, 1, j] = \gamma_{jk}$. When there is more than one element in h'_i , where $h'_i = (j, \dots, z)$, we first construct a shortest path tree with the zone j as its root. This tree would contain all the nodes in V (assuming that the network is connected). Then, consider the subtree with z as its root, and let $V_{j,z}$ be the set of all the zones in the subtree. For every zone $k \in V_{j,z}$, we have:

$$\Pr[k|b, 1, (j, \dots, z)] = \frac{b(k)}{\sum_{k' \in V_{j,z}} b(k')} ,$$

while the probability is zero for the rest of the zones outside the subtree. This computation process is illustrated in Figure 2.

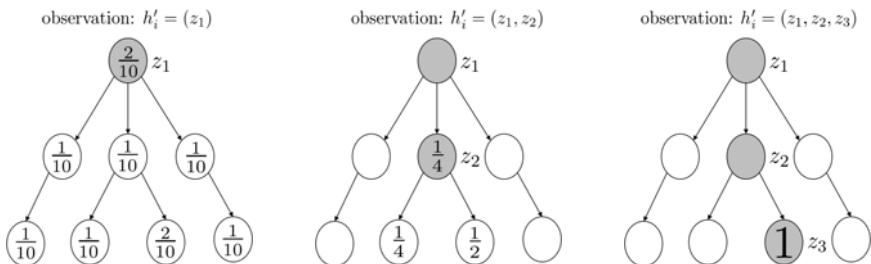


Fig. 2. Example of the agent's belief regarding the destination zone of an occupied driver, which is represented by a shortest path tree. The number in the node represents the probability of having that node be the destination zone, with empty node denoting zero probability. The tree is updated as new observation is received.

5.2 Dynamic Programming for POMDP

The first step in solving a POMDP by dynamic programming is to convert it into a completely observable Markov decision process (MDP) with the set of states $\mathbb{S} = \Delta(S)$, i.e. the set of all possible beliefs about the state. In this paper, we focus on computing the optimal policy for T -horizon planning given an initial belief state of the system b^0 . When $T = 1$, the agent can take only a single action. When $T = 2$, it can take an action, make an observation, then take another action based on the observation made. In general, the agent's t -step policy can be represented by a tree. The root of the tree specifies the first action. The resulting observation determines the edge to be followed from the root, which determines the next action to be taken. A t -step policy tree, therefore, is a perfect $|\Omega|$ -ary tree of depth t . The next step, is to find the value function, that gives the value of a policy, given an initial belief state. We can then use the value function to directly determine the optimal policy. Let $\mathcal{V}_{p^t}(b)$ be the value function denoting the value of executing the t -step policy p^t , given the belief state b . Since:

$$\mathcal{V}_{p^t}(b) = \sum_{s \in S} b(s) V_{p^t}(s), \quad (3)$$

where $V_{p^t}(s)$ is the value of executing the policy p^t in the state s , the next step is to find the value of a policy given a state of the system. This is given by:

$$\begin{aligned} V_{p^t}(s) &= R(s, a(p^t)) + \\ &\sum_o O(s, a(p^t), o) \sum_{s'} T(s, a(p^t), s') V_{o(p^t)}(s'), \end{aligned} \quad (4)$$

where $a(p^t)$ denotes the action associated with the root of the policy tree p^t , and $o(p^t)$ denotes the subtree (a $(t - 1)$ -step policy tree) of p^t , whose root can be obtained by following the edge associated with observation o , from the root of p^t . Given an initial belief state b^0 , the optimal T -step policy is then given by:

$$\Pi^*(b^0) = \arg \max_{p^T} \mathcal{V}_{p^T}(b^0).$$

Notice from equation 3 that, associated with each t -step policy p^t is a vector of $|S|$ -dimension. We call this the *value vector* of the policy and denote it by V_{p^t} . A single value vector V_{p^t} is enough to compute $\mathcal{V}_{p^t}(b)$ for different values of b . The general dynamic programming algorithm for POMDP proceeds in T iterations, where each iteration consists of two steps.

In the first step of iteration t , the algorithm is given a set P^{t-1} of $(t - 1)$ -step policy trees computed from the previous iteration, and the corresponding set of value vectors V^{t-1} . It then computes the new sets P^t and the corresponding V^t as follows. First, a set of t -step policy trees, P_+^t is created by generating every possible t -step policy tree that makes a transition, after the initial action and observation, to the root of some t -step policy tree in P^t . Note that

$|P^t| = |A||P^{t-1}|^{|\Omega|}$. The value vector for each of the new t -step policy tree can then be computed from equation 4, and stored in the set V_+^t .

In the second step, the set P_+^t is pruned by removing policy trees that need not be considered because they are dominated by some other policy trees in the set. Specifically, a policy tree p^t with the corresponding value vector V_{p^t} is dominated if for all $b \in \mathbb{S}$ there exists another policy $q^t \in P^t \setminus p^t$ such that $\mathcal{V}_{q^t}(b) \geq \mathcal{V}_{p^t}(b)$. This test of dominance is implemented using linear programming. If p^t is removed from P_+^t , then the corresponding value vector V_{p^t} is removed from V_+^t as well. The resulting set, P^t and V^t , are then sent to the next iteration of the algorithm.

6 Experimental Results

6.1 Extracting Driver Cruising Behavior from Historical Data

In order to derive the state transition function, we first analyze a real trace of historical data of a large Singapore taxi operator for one week on the Singapore road network. For this purpose, we obtained a dataset from a local taxi operator, which consists of a detailed information of each taxi in each second interval - its speed, its status (passenger on board, free, offline, etc), current position (given in latitude-longitude coordinate).

We first divide the Singapore's road network into 88 logical zones (see for instance <http://www.onemap.sg/index.html>) and derive a mapping function from lat-long coordinates to the respective zone number. The information we need to extract from the dataset is the distribution patterns of the cruising. From the information extracted, we observe that for a randomly chosen 500 drivers and over all the zones, the following random variables are close to being exponentially distributed: (1) cruising time of an unoccupied taxi in a zone before moving to adjacent zone without finding any passenger, (2) time taken by the taxi moving from zone to zone while occupied, and (3) time taken to find the exact place in the destination zone to drop the passenger off. The frequency histograms for these random variables can be seen in Figure 3. This provides the motivation for us to model the driver cruising behavior as a continuous time Markov chain.

6.2 Simulation

To evaluate the quality of the POMDP policy, we compared it with two other policies, namely random and greedy. The random policy is to choose a random zone within its neighborhood to move to. The greedy policy, when unoccupied, is to move to a zone (among its current and adjacent zones) that has the least number of cruising taxis. Note that the random policy provides the lower bound benchmark. Any method that seeks to improve the agent cruising policy should at least perform better than a random policy. The question is whether the POMDP is more intelligent compared to the myopic greedy policy.

We simulated both passengers "arrivals" and drivers movement on the network. The passenger arrival in each zone is implemented as a Poisson process. As

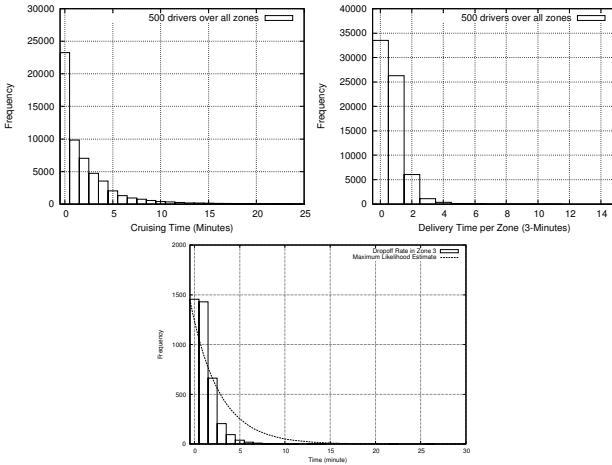


Fig. 3. Observations from Real Dataset

a passenger appears in a zone, it joins a FIFO queue associated with the zone. If there are some taxis cruising in the zone, the passenger at the head of the queue is removed and assigned to a randomly chosen taxi. The average time spent in the queue models the average time spent by passengers waiting for a cruising taxi. We are interested in the utilization rate of the single agent driver in the network. The movement of all other drivers follows the continuous-time Markov chain model. When entering a zone in the cruising state, a passenger might be assigned to the driver, and it enters the occupied state. Otherwise, a set of exponentially distributed random variables is instantiated, one for each adjacent zone according to the corresponding rate in the Markov chain. The driver will wait for the smallest value of the instantiated random variables before moving to a zone following the edge associated with the random variable. While waiting, a passenger may still be assigned to the driver, in which case, it enters the occupied state instead. For initialization, we allow the system to run without the agent for a small time interval to reach a stable state. After a stable state has been reached, we freeze the system and save its configuration. Then, the agent driver to be tested is inserted to the system in a randomly chosen zone before resuming the simulation. The same agent is inserted to the same configuration several times before the average result is collected.

We performed our experiments on two sets of different network topologies and settings. First, on small grid networks. The parameters of the simulation are as follows. The average time between two passenger appearances in each zone is set at 0.5 minutes. The destination zone of a passenger are uniformly distributed over all the zones. During simulation, the agent accesses the state of the system every minute and decides on an action to take. The metric for comparison between the random, greedy and POMDP agents is the percentage of the 20-minute period when it was in the occupied state. Table 1 shows the results on different sizes of network, while the number of drivers was kept constant at 20. As shown in

the table, the improvements are more significant on a congested network (larger taxi-to-zone ratio) than on a sparse network. This is intuitive, as competition among the taxi drivers is fiercer on a congested network, and the agent with better cruising policy will generally perform better.

Table 1. Results of small grid networks

	2 × 2	3 × 3	4 × 4	5 × 5
POMDP agent	41%	72%	81%	85%
greedy agent	22%	61%	75%	80%
random agent	17%	51%	66%	71%

The second set of experiments is performed on a real 15-node network that models the congested central business district of Singapore and its surrounding zones (Figure 4). The arrival rate of passengers in each zone is derived from the real dataset discussed above. We consider only passengers with origins and destinations within these 15 zones, and restrict the cruising area of the drivers to be within these zones as well. Similar to the first set, the agent, in the unoccupied state, accesses the state of the system every minute and decides on the action to take. The number of drivers operating in the area is set to be 250. We simulate 2 hours of real time operations. As shown from the result, the POMDP agent performs significantly better than the greedy agent.

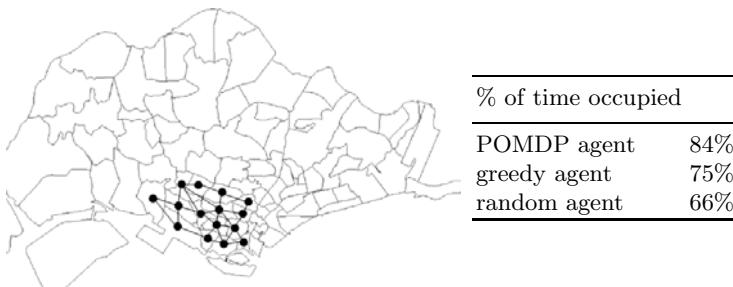


Fig. 4. Results of restricted area of Singapore

7 Future Works

We are aware of several limitations of the current model and its solution method. First is the issue of scalability. This is an inherent problem in practically all POMDP models for planning, and ours is no exception. It is thus interesting to see if the problem structure can be exploited for a more aggressive pruning strategy. It is also worthwhile to explore approximate solutions (for example by eliminating policy trees that are approximately dominated in the pruning step), to understand the trade-off between the quality of sub-optimal policies and the

gain in computation time. The second is the extension to multi-agent system. A possible extension is to consider a scenario where every taxi driver is equipped by a tool, telling them the optimal cruising policy, given the current state of the system and the policy of the other drivers. Information might be shared publicly, and each driver observe the state of the system completely, including the policy of the other drivers, or it might be kept private, and each agent has to guess the policy of the other drivers. The interesting questions are: What is the optimal policy of a driver in this case? What is the resulting state of the system if every agent execute its optimal policy, and how do we characterize equilibrium?

References

1. Becker, R., Zilberstein, S., Lesser, V., Goldman, C.V.: Transition-independent decentralized markov decision processes. In: Proc. 2nd Conf. on Autonomous Agents and Multi-agent Systems, pp. 41–48 (2003)
2. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of markov decision processes. Mathematics of Operation Research 27, 819–840 (2002)
3. Cassandra, A., Littman, M.L., Zhang, N.L.: Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In: Proc. 13th Conf. on Uncertainty in Art. Intell., pp. 54–61 (1997)
4. Hansen, E.A.: Solving pomdps by searching in policy space. In: Proc. 14th Conf. on Uncertainty in Art. Intell., pp. 211–219 (1998)
5. Hansen, E.A., Bernstein, D.S., Zilberstein, S.: Dynamic programming for partially observable stochastic games. In: Proc. 19th AAAI Conf. on Art. Intell., pp. 709–715 (2004)
6. Hansen, E.A., Feng, Z.: Dynamic programming for pomdps using a factored state representation. In: Proc. 5th Conf. on AI Planning Systems, pp. 130–139 (2000)
7. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Art. Intell. 101, 99–134 (1998)
8. Nair, R., Tambe, M., Yokoo, M., Pynadath, D., Marsella, S.: Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In: Proc. 18th Int'l Joint Conf. on Art. Intell., pp. 705–711 (2003)
9. Smallwood, R.D., Sondik, E.J.: The optimal control of partially observable markov processes over a finite horizon. Operations Research 21, 1071–1088 (1973)
10. Wong, K.I., Wong, S.C., Bell, M.G.H., Yang, H.: Modeling the bilateral micro-searching behavior for urban taxi services using the absorbing markov chain approach. Advanced Transportation 39, 81–104 (2005)
11. Wong, K.I., Wong, S.C., Yang, H.: Calibration and validation of network equilibrium taxi model for hong kong. In: Proc. 4th Conf. of the Hong Kong Society for Transportation Studies, pp. 249–258 (1999)
12. Wong, K.I., Wong, S.C., Yang, H., Wu, J.H.: Modeling urban taxi services with multiple user classes and vehicle modes. Transportation Research Part B: Methodologocal 42, 985–1007 (2008)
13. Yang, H., Wong, S.C.: A network model of urban taxi services. Transportation Research Part B: Methodological 32, 235–246 (1998)
14. Yang, H., Wong, S.C., Wong, K.I.: Demand-supply equilibrium of taxi services in a network under competition and regulation. Transportation Research Part B: Methodological 36, 799–819 (2002)

Next-Best-View Planning for 3D Object Reconstruction under Positioning Error

Juan Irving Vásquez and L. Enrique Sucar

Department of Computer Science, INAOE, Tonantzintla, Puebla, Mexico
`{ivasquez, esucar}@ccc.inaoep.mx`

Abstract. To acquire a 3D model of an object it is necessary to plan a set of locations, called views, where a range sensor will be placed. The problem is solved in greedy manner, by selecting iteratively next-best-views. When a mobile robot is used, we have to take into account positioning errors, given that they can affect the quality and efficiency of the plan. We propose a method to plan “safe views” which are successful even when there is positioning error. The method is based on a reevaluation of the candidate views according to their neighbors, so view points which are safer against positioning error are preferred. The method was tested in simulation with objects of different complexities. Experimental results show that the proposed method achieves similar results as the ideal case without error, reducing the number of views required against the standard approach that does not consider positioning error.

Keywords: Planning, View Planning, Next-Best-View, Object Reconstruction, Modeling.

1 Introduction

Three dimensional (3D) models of objects are used in a wide variety of tasks in robotics, such as manipulation, recognition, tracking, etc. Therefore, it is desirable that an autonomous robot can acquire 3D models from the objects in its environment by itself. One way to acquire a 3D model of an object is by locating a range sensor on different positions [2], often called *views*. Due to the sensor’s limited field of view and auto-occlusions of the objects, the robot must plan the views to observe the whole surface. This task is known as *view planning for object reconstruction*. Generally it is assumed that the robot has previous knowledge on the approximate size and position of the object. The problem is solved in greedy manner, via a cyclic process of sensing, registering, planning the next-best-view (NBV) and moving the sensor. The NBV is selected based on evaluating a set of potential positions by optimizing certain criteria, in particular to observe unknown areas and at the same time to have some overlap with previous ones for registration.

Previous work assumes perfect positioning of the sensor, which is a reasonable assumption when the sensor is positioned by a fixed manipulator or the object is moved by a turntable. However, if the sensor is mounted on a mobile or

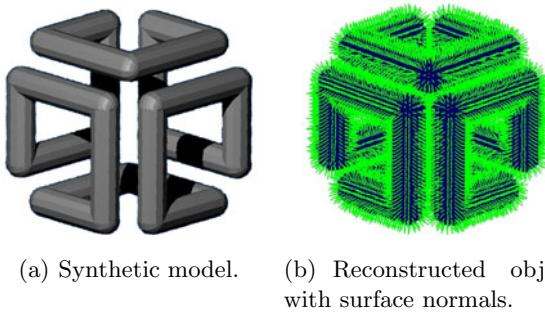


Fig. 1. Object SGI-Logo. This is an example of a complex object which represents a reconstruction challenge when there is positioning error.

humanoid robot there is no longer a perfect positioning. The positioning error depends on the platform, and is worst for low-cost mobile platforms [16]. One of the consequences of the positioning error is the failure of the registration step [13] [5], making impossible to fuse the new images with the previous images. Besides the failure of the registration, the positioning error increases the number views required for reconstruction, wasting time and energy of the robot.

The positioning error does not affect significantly the reconstruction of simple objects like a sphere, but for complex objects the reduction in performance is notable. This occurs because auto-occlusions and positioning error are combined. For example, let the object be the SGI-logo (Fig. 1). This object has several auto-occlusions and holes. To completely observe the surface we need to see the interior parts of the logo, but positioning error makes that the planned views will be occluded by other tubes, resulting in a waste of time and scans. Fig. 2 shows how the reconstruction percentage diminishes as the positioning error is incremented. So it is desirable that, besides fulfilling the required constraints, the planning algorithm considers the positioning error and selects the views that minimize the effects of this error.

We have developed a method to produce *safe views*, which are still good views despite of positionig error. The method consists of two phases. In the first phase all the candidate views are evaluated. Then, in a second phase, the views are reevaluated to select a safe view. The reevaluation in the second phase is based on the convolution of the utility function and a Gaussian function which takes into account the positioning error.

The method has been evaluated in simulation with several 3D objects of different complexities, varying the positioning error. The results show that by incorporating the positioning error model under the proposed scheme, there is a significant improvement with respect to the standard approach that does not consider this error. The performance of our method is close to the ideal performance when there is no error.

The rest of the paper is organized as follows. Section 2 summarizes related work on view planning with emphasis on those that consider positioning error.

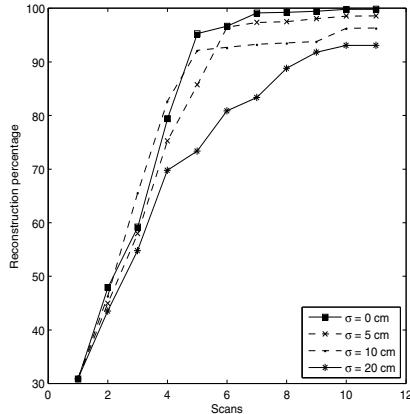


Fig. 2. Reconstruction percentage against number of views for the object SGI-logo with several amounts (indicated by the standard deviation or S.D.) of positioning error. The figure shows how the percentage of reconstruction decreases as the positioning error increases. For instance, at the 11th scan with a S.D. of 20 cm of positioning error, the reconstruction has decreased almost 10%.

Section 3 presents an overview of the proposed method, which is detailed in Sections 4 and 5. The experiments in simulation and the results are described in Section 6. Finally, Section 7 concludes the paper and provides ideas for future work.

2 Related Work

Since the 1980's the problem of view planning for object reconstruction has been addressed. There are two main review papers that make a thorough analysis of these algorithms, by Tarabanis et al. [14] and by Scott et al. [12].

Due to the high dimensionality of the problem, most of the previous work follows one of two strategies. The first strategy is to use a search-based method, where the search space is sampled and the candidate views are tested to find the NBV [3] [7] [15], or the NBV is obtained using optimization methods [4]. Under the second strategy, surface-based methods, the scanned surface is analyzed and the NBV is generated by an analytical method [8] [1]. All these methods are capable of finding the NBV fulfilling the required constraints; however, most of them have been focused on controlled environments where fixed robotic manipulators or turntables are used [3] [8] [1] [7].

The few approaches which face the positioning error consider it in the registration step [4] [17] [15], where the surfaces are merged and the robot is re-localized. However, the registration algorithms have their limitations; with high positioning error or not enough overlap they cannot fuse the surfaces [11] [9]. Additionally, as mentioned before, the positioning error increases the number of views required to reconstruct an object.

In the field of environment reconstruction, Low and Lastra in [5] describe an analytical method to estimate the absolute registration error, based on this estimation they can reject bad views. However, this approach differs from our method since they reject views before evaluation, and we consider all the views and select the safe ones.

3 Overview of the Method

The reconstruction process gets a 3D model of an object given a initial position for the robot and the position of the object. Algorithm 1 resumes the process in a cycle of four basic steps: (i) positioning, the robot place the sensor at an indicated configuration, (ii) scanning, a scan is taken from the current position (iii), updating, it consists of registering (alignment of the new scan) and merging (fusion of the scans into one unique model), and (iv) planning the next-best-view, where the next configuration of the sensor is selected. In this paper we address

Algorithm 1. Object Reconstruction

```

Data: Initial position  $p_0$ 
Result: 3D model  $\mu$ 
 $p = p_0;$ 
while True do
    Positioning( $p$ );
     $\iota = \text{TakeImage}();$ 
     $\mu = \text{UpdateModel}(\iota);$ 
     $p = \text{PlanNBV}();$ 
    if  $p$  ‘‘does not provide new information’’ then
        | Return  $\mu$ ;
        | Exit;
    end
end
```

the problem of planning the next-best-view when there is positioning error. Our approach can be summarized in four steps: (a) candidate views generation, (b) views evaluation, (c) convolution after evaluation, and (d) selection of the NBV. Each step is detailed below and algorithm 2 resumes the approach.

First a set of candidate views is generated. We assume that our scene is a cube, the robot and object are placed in it and we know the position of the object and its maximum size. With this information we establish a cube of unknown area where the object is encapsulated. Then we make a uniform sampling of the 3D space producing the set of candidate views V . Each generated view has two attributes, configuration and evaluation. The configuration is formed by the position and the orientation towards the object. The evaluation is a numerical value assigned by a utility function in the following step.

On the second step the candidate views are evaluated with a utility function given the current state of the model. The utility function gives a high numerical

value to those views considered as *good* views, based on several criteria. After evaluation, each view $v \in V$ has an evaluation $e \in E$. Common utility functions look for views which provide unknown surfaces, and at the same time overlap with previous scans. For this paper we have used a reduced version of the utility function proposed in [15]. It considers as the best view that which gives an 80% of unknown area and a 20% of overlap.

Given that the evaluation of all candidate views is computationally expensive, we consider only those which are between a minimum and maximum radius from the center of the object; this produces a *fat* view sphere, like a donut, around the object. See fig. 4. We also only consider views where the robot can be placed by incorporating positioning constraints.

Most previous approaches finish here, by selecting the view with the highest evaluation. Instead of that we go further, we analyze the set of evaluations and provide *safe* views, those that are more probable to be successful against positioning errors. To explain the method we will make an analogy between the set E of evaluations and a 2D image. In E there are good views but sometimes these views are surrounded by bad views, like a white pixel among black pixels. When the robot tries to reach that configuration, the white pixel, it is highly probably that it will fall into a *bad* configuration due to the positioning error. So, despite its high value, a view is not so good if it is surrounded by *bad* views. So it is preferable to chose a semi-optimal view surrounded by *good* views. Our method performs this reevaluation of views according to their neighbors. Returning to the 2D image analogy, our method blurs the image smoothing the contrast. The new *blurred* set is where the NBV is found.

Algorithm 2. Plan Next-Best-View

Data: Current model μ
Result: Next-Best-View p
 $V = \text{GenerateCandidateViews}()$;
 $E = \text{Evaluate}(V, \mu)$;
 $F = \text{Convolution}(E)$;
 $p = \text{SelectViewWithBestValue}(F)$;
Return p ;

4 Convolution

In order to find a safe view against positioning error we reevaluate the views according to their neighbors. To perform this task it is necessary a technique that increases the value of a view when the neighbors were good, besides the technique must reduce the value when the neighbors were bad. One way to perform this task is making a convolution, where based on two functions a third function is produced. The two basic functions are the utility function and a multidimensional Gaussian function. Here the utility function represents the goodness of a view and the Gaussian represents the distribution of the positioning

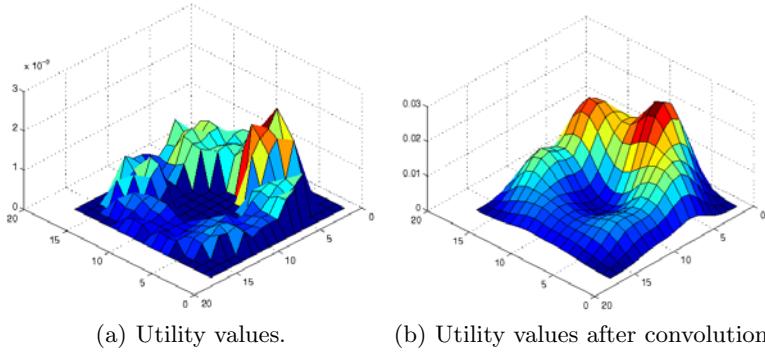


Fig. 3. Plot of the utility function in one slice of the search space. After convolution with a Gaussian the function is smoothed, integrating the effect of the positioning error in the value function.

error. Then after convolution we get a function which reevaluates the goodness of a view depending on the positioning error. Below we detail the performed convolution.

Let f be the utility function, g a multi-dimensional Gaussian function, then their convolution is given by equation (1).

$$[f * g](v) = \int f(\tau)g(v - \tau)d\tau \quad (1)$$

Function g is defined by equation (2).

$$g(\mu, \Sigma, X) = A \exp \left[-\frac{1}{2}(X - \mu)^T \Sigma^{-1} (X - \mu) \right] \quad (2)$$

where X is a vector with the error coordinates, μ the expected positioning error (mean) and Σ the covariance matrix. The dimension of g is established by the number of degrees of freedom of the robot. For convenience we have established the parameter $A = 1$ to keep the maximum value at one.

The search space for the NBV is defined by the co-domain of the convolution (1). Fig. 3 shows an example of the search space before and after convolution.

4.1 Sampling

Equation (1) gives a continuous search space for the NBV. However, given that addressing the problem of finding a solution in the continuous space is intractable, we have implemented our method using sampling and a discrete convolution.

It is worth to say that by now we are considering Cartesian space, so the Gaussian is defined by a three-dimensional function, which considers positioning error on the three axis. In future work we pretend to use multi-dimensional Gaussian according to the robot's degrees of freedom.

After the evaluation step we have a three-dimensional matrix ($n \times n \times n$) of evaluations E , where n is the number of divisions per side of the scene. Then we defined a second 3D matrix M ($n \times n \times n$) of uniformly distributed points, where the central element is the mean value μ ($\mu_x = 0, \mu_y = 0, \mu_z = 0$) and remaining elements are uniformly distributed coordinates, with the same distribution that the set of candidate views V . The covariance matrix is defined according to equation (3).

$$\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \quad (3)$$

Then we use M , μ , and Σ to generate a matrix G which contains the evaluations of the normal Gaussian function according to equation (2). Finally, a multi-dimensional convolution of the matrices E and G is done. Then we take the view with the highest value in C as the NBV.

5 View Planner

In this section we describe the representation and configuration of the scene and the view planner. We use a *voxel* map to represent the scene and store the partial models of the object at each iteration. Fig. 4 shows the configuration and size of the scene. The voxel map is conformed by a three-dimensional matrix of labeled voxels. Each voxel stores information about the space that it represents by means of three attributes [10]: a label indicating its type, a surface normal and a quality value. For this paper we are only using the first attribute, type. Each label has a color associated with it for display purposes. The labels are defined as follows [6]:

- *Unmarked*. A voxel that has not been observed yet by the range camera.
- *Occupied*. A voxel which position matches with acquired points in the 3D range image.
- *Empty*. A voxel in the observed area but for which there are no acquired points in this position.
- *Occluded*. A voxel in the sensor field of view but not seen because it is behind of an occupied voxel.
- *Occplane*. An occluded voxel that is adjacent with any of its six faces to an empty voxel.

To evaluate a view we perform a ray tracing from the view's position, similarly to taking a “synthetic image” inside the voxel map. Then the information obtained after the ray tracing is used by the utility function, equation (4), to get a numerical value. The utility function guarantees an overlap with previous scans and provides new information of unknown areas. See [15] for details.

$$f_{utility} = h(x, \alpha) + h(y, \beta) \quad (4)$$

where x is the percentage of occupied voxels in the image and y is the percentage of occplane voxels. The percentage of a voxel type is calculated as the amount of voxels of this type divided by total number of voxels in the ray tracing; empty voxels are not considered. In the experiments we used $\alpha = 0.8$ and $\beta = 0.2$. Function h is defined by equation (5).

$$h(x, \alpha) = \begin{cases} h_1(x, \alpha), & x \leq \alpha \\ h_2(x, \alpha), & x > \alpha \end{cases} \quad (5)$$

where

$$h_1(x, \alpha) = -\frac{2}{\alpha^3}x^3 + \frac{3}{\alpha^2}x^2 \quad (6)$$

and

$$h_2(x, \alpha) = -\frac{2}{(\alpha - 1)^3}x^3 + \frac{3(\alpha + 1)}{(\alpha - 1)^3}x^2 - \frac{6\alpha}{(\alpha - 1)^3}x + \frac{3\alpha - 1}{(\alpha - 1)^3} \quad (7)$$

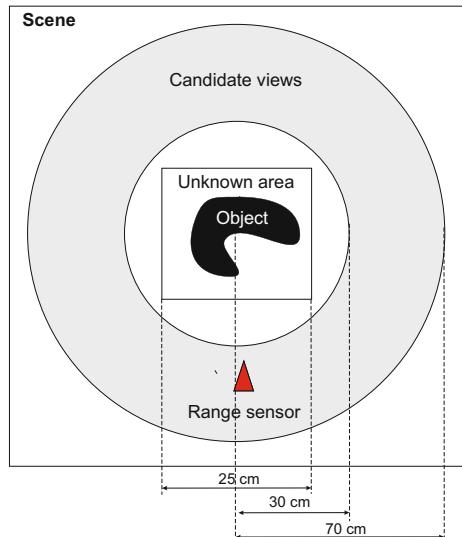


Fig. 4. 2D representation of the reconstruction scene. The object to be reconstructed (black) is placed in the center of the scene. Surrounding the object is the area (volume in 3D) where the sensor can be positioned, between the two concentric circles (light gray)

6 Experiments

The experiments pretend to show if using the convolution after evaluation is possible to improve the efficiency of the reconstruction (number of views and coverage). We call “basic method” to our same method but without convolution. The experiments compare the proposed method, convolution after evaluation (CAE), against: (i) the basic method without positioning error that serves as the desired objective; (ii) the basic method with positioning error that serves as a baseline, we expect that our method can improve it significantly.

The objects to be reconstructed are a sphere, a mug and the SGI-logo (Fig. 5), which provide a sample of objects of different complexities; from a very simple and regular object (sphere) to a complex one with holes and auto-occlusions (SGI-logo). The reconstruction scene has been configured with the object in the center and free space surrounding it. The sensor has five degrees of freedom ($x, y, z, pan, tilt$) and can be positioned inside the free space.



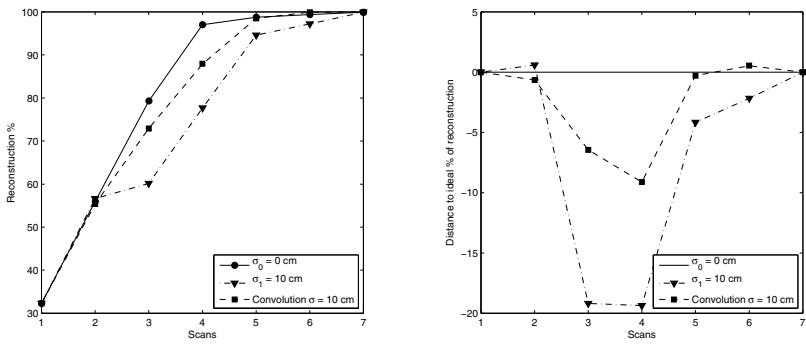
Fig. 5. Objects to be reconstructed

We have divided the scene in 17 points per dimension, given a total of 4,913 (17^3) candidate views, so $n = 17$ and $|V| = 4913$. Given that we only considered views that are between the interior and exterior radius, the evaluated views were 1,932. Our experiments considered only positioning error in the three axis (x, y, z), we did not include positioning error in *pan* or *tilt* which will be modeled in future work. Also, we have assumed the following standard deviations: $\sigma_x = 10cm$, $\sigma_y = 10cm$, $\sigma_z = 10cm$ and conditional independence between axis. Given that, we set the covariance matrix as:

$$\Sigma = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \quad (8)$$

The experiments consisted on running the planner three times with each object, one with no error ($\sigma_0 = 0cm$), the second with a positioning error with standard deviation of 10 cm ($\sigma_1 = 10cm$) and the third one using the convolution after evaluation (CAE) method and the same error. First we summarize the results for the sphere and mug, and then present in more detail the results for the SGI-logo, as it provides the more challenging case.

The results of the objects sphere and mug are summarized in figures 6 and 7, respectively. For the sphere, the reconstruction was completed in the same scan by all methods. The standard method with error was a little less greedy in the intermediate scans, but at the end it reconstructed a similar percentage of the object. However, the CAE method kept closer to the ideal reconstruction; so if the reconstruction was finished at an intermediate scan, more surface will be reconstructed with the CAE method. For the Mug, the same pattern is observed; where the CAE method keeps closer to the ideal case without error. In this case when the process ended, the largest percentage was covered with the ideal reconstruction, in second place was the CAE method and in third place was the standard method with error.

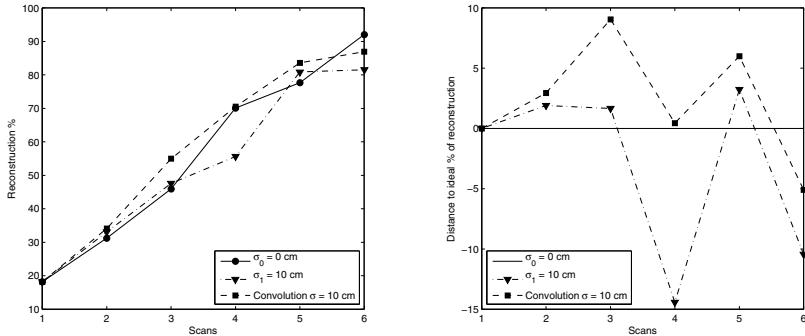


(a) Absolute reconstruction percentage (b) Relative reconstruction percentage for each method per scan, until the stop (with respect to the ideal percentage of criteria is reached).

a system with no positioning error): $y = x - \sigma_0$. The graph shows that the CAE method keeps closer to the ideal.

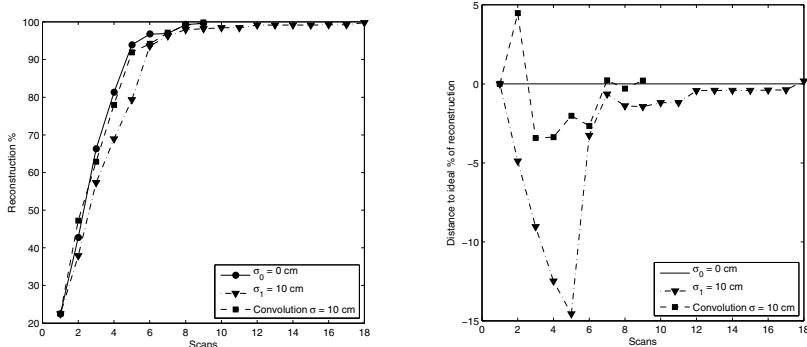
Fig. 6. Reconstruction of the Sphere object. Comparison of the percentages of reconstruction using: a) $\sigma_0 = 0 \text{ cm}$, $\sigma_1 = 10 \text{ cm}$ and “convolution after evaluation method”. The graphs depict the reconstruction percentage against the number of views or scans in absolute (a) and relative (b) terms.

Table 1 depicts the results of each method per column for the SGI-logo, in the same order as described above. It shows the percentage of surface observed after each step in the reconstruction process, until the termination condition of observing 99.5% or more is reached. This is shown graphically in Figure 8. It is clear in this experiment that the standard approach with positioning error requires two times as much views to reconstruct this object, while our approach can reconstruct it in the same number of steps as without error. We observe that the “safe” views provided by the CAE method keep the object reconstruction close to the ideal case without error even for a complex object. Figure 8(b) provides another perspective by plotting the differences in reconstruction percentage between the ideal case (no error) and both methods: the standard



- (a) Absolute reconstruction percentage for each method per scan, until the stop (with respect to the ideal percentage of criteria is reached).
- (b) Relative reconstruction percentage of a system with no positioning error: $y = x - \sigma_0$.

Fig. 7. Reconstruction of the Mug object. The graphs depict the reconstruction percentage against the number of scans in absolute (a) and relative (b) terms. The CAE method keeps closer to the ideal reconstruction. At the sixth view it is 5% below ideal, compared with the 10% below of the standard planner with positioning error. From the second to the fifth view the percentage of reconstruction of CAE is higher than ideal; this is because the reconstructions followed different paths, however all the time the overlap was satisfied and the final result is the same.



- (a) Absolute reconstruction percentage for each method until the stop criteria $y = x - \sigma_0$. The graph shows that CAE in this case always keeps closer to the ideal positioning error and convolution after evaluation converge at the same time in the nineteenth scan. The reconstruction with positioning error requires 18 scans.

Fig. 8. Reconstruction of the SGI-logo. The graphs depict the reconstruction percentage against the number of scans in absolute (a) and relative (b) terms.

Table 1. Reconstruction of the SGI-logo. The table shows the reconstruction percentage after each scan for the three methods until the stop criteria is reached, for the object SGI-logo. The first column corresponds to the NBV without error, the second with positioning error with $\sigma_1 = 10cm$, and the third with the same error incorporating the convolution technique.

Scan	Reconstructed Percentage		
	$\sigma_0 = 0cm$	$\sigma_1 = 10cm$	Convolution
1	22.51	22.51	22.51
2	42.79	37.91	47.26
3	66.33	57.28	62.9
4	81.36	68.88	77.99
5	93.93	79.37	91.91
6	96.82	93.55	94.16
7	96.91	96.26	97.13
8	99.36	97.96	99.06
9	99.62	98.17	99.83
10		98.42	
11		98.42	
12		99.19	
13		99.2	
14		99.2	
15		99.2	
16		99.22	
17		99.24	
18		99.79	

approach with positioning error and the proposed CAE method with positioning error. We observe that for the standard approach in the first views less surface is scanned, and more views are required to reach the same result. On the other hand, CAE keeps the reconstruction percentages close to that without error and converges in the same number of scans.

Based on these experiments, we can conclude that the proposed method is robust against positioning errors with a performance close to that of the ideal case without errors. The advantage is more significant for complex objects, while for simple objects there is no much difference against the standard approach. Thus, the main benefits of the CAE method are: (i) complex objects can be reconstructed in a smaller number of views when there is positioning error, (ii) registration errors could be prevented given that the overlap is kept all the time, (iii) if the reconstruction process has to be interrupted (for instance due to time limitations), the reconstruction percentage is higher than without taking into account the errors.

The main limitation of the method is that it is necessary to evaluate many views before the convolution step. In our implementation around 20 minutes are needed to compute a NBV in an Intel Core i5 laptop. However, the convolution

step takes only a few seconds. The computation time can be reduced by implementing a more intelligent sampling or by using a multi-resolution strategy, which are topics of future research.

7 Conclusions and Future Work

If we use a mobile robot, such as a mobile manipulator or humanoid robot, for object reconstruction, we have to take into account positioning errors, as these can affect the efficacy and efficiency of the reconstruction process. We have developed an extension to the traditional NBV planning for object reconstruction that considers positioning error, and tested it in simulation with several objects of different complexities. The experimental results show that the proposed method achieves similar results as the ideal case without error, reducing in a significant amount the number of views required for a complex object against the standard approach. The results also provide evidence that our method keeps a reconstruction percentage close to the ideal all the time, providing a kind of any-time algorithm.

In future work we plan to develop a multi-resolution strategy to accelerate the reconstruction process, and extend the method to consider the configuration space of a mobile manipulator.

Acknowledgments. This work was supported in part by Conacyt grant No. 47968 and the National Institute of Astrophysics, Optics and Electronics (INAOE). We appreciate the collaboration of Dr. Felipe Orihuela, Dr. Eduardo Morales and Dr. Rafael Murreita.

References

1. Banta, J.E., Wong, L.M., Dumont, C., Abidi, M.A.: A Next-Best-View System for Autonomous 3-D Object Reconstruction. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 30, 589–598 (2000)
2. Chen, S., Li, Y., Zhang, J., Wang, W.: Active Sensor Planning for Multiview Vision Tasks. Springer, Heidelberg (2008)
3. Connolly, C.: The determination of next best views. In: *IEEE International Conference on Robotics and Automation, ICRA 1985*, vol. 2, pp. 432–435 (1985)
4. Foissotte, T., Stasse, O., Escande, A., Kheddar, A.: A next-best-view algorithm for autonomous 3D object modeling by a humanoid robot. In: *Proc. of International Conference on Humanoid Robots 2008*, pp. 333–338 (2008)
5. Low, K.-L., Lastra, A.: Predetermination of icp registration errors and its application to view planning. In: *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, pp. 73–80 (2007)
6. Lozano, M.T., Devy, M., Sanchiz, J.M.: Perception Planning for an Exploration Task of a 3d Environment. In: *Proceedings of the 16th International Conference on Pattern Recognition, ICPR 2002*, vol. 3, pp. 704–707 (2002)
7. Massios, N.A., Fisher, R.B.: A Best Next View Selection Algorithm Incorporating a Quality Criterion. In: *British Machine Vision Conference, BMVC 1998*, pp. 780–789 (1998)

8. Maver, J., Bajcsy, R.: Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 417–433 (1993)
9. Salvi, J., Matabosch, C., Fofi, D., Forest, J.: A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput.* 25, 578–596 (2007)
10. Sanchiz, J., Fisher, R.: A next-best-view algorithm for 3d scene recovery with 5 degrees of freedom. In: British Machine Vision Conference 1999 (Nottingham, UK), pp. 163–172 (1999)
11. Scott, W.R., Roth, G., Rivest, J.-F.: Pose error effects on range sensing. Technical report, NRC Institute for Information Technology; National Research Council Canada (2002)
12. Scott, W.R., Roth, G., Rivest, J.-F.: View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.* 35, 64–96 (2003)
13. Scott, W.R., Yz, W.R.S., Roth, G., Rivest, J.-F.: View planning with positioning system error. Technical report, NRC Institute for Information Technology; National Research Council Canada (2004)
14. Tarabanis, K., Allen, P., Tsai, R.: A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation* 11, 86–104 (1995)
15. Vasquez, J.I., Lopez-Damian, E., Sucar, L.E.: View Planning for 3D Object Reconstruction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009), pp. 4015–4020 (2009)
16. Zhang, S., Yan, G., Sheng, W.: Viewpoint planning for automated 3d digitization using a low-cost mobile platform. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4369–4374 (2009)
17. Zhang, S., Yan, G., Sheng, W.: An efficient planning method for mobile robot based 3d digitization. In: 8th World Congress on Intelligent Control and Automation (July 2010)

Stochastic Learning Automata for Self-coordination in Heterogeneous Multi-Tasks Selection in Multi-Robot Systems

Yadira Quiñonez, Darío Maravall, and Javier de Lope

Computational Cognitive Robotics
Dept. Artificial Intelligence
Universidad Politécnica de Madrid
`ay.quinonez@alumnos.upm.es, dmaravall@fi.upm.es,`
`javier.delope@upm.es`

Abstract. This paper focuses on the general problem of coordinating multiple robots. More specifically, it addresses the self-election of heterogeneous specialized tasks by autonomous robots, as opposed to the usual multi-tasks allocation problem in multi-robot systems in which an external controller distributes the existing tasks among the individual robots. In this work we are considering a specifically distributed or decentralized approach in which we are particularly interested on decentralized solution where the robots themselves autonomously and in an individual manner, are responsible of selecting a particular task so that all the existing tasks are optimally distributed and executed. In this regard, we have established an experimental scenario and we propose a solution through automata learning-based probabilistic algorithm, to solve the corresponding multi-tasks distribution problem. The paper ends with a critical discussion of experimental results.

Keywords: Multi-robot Systems, Stochastic Learning Automata, Multi-tasks Distribution, Self-Coordination of Multiple Robots, Reinforcement Learning, Multi-Heterogeneous Specialized Tasks Distribution.

1 Introduction

In general, a Multi-Robot System (MRS) consists of a team of robots that, in the same environment, interact with each other to achieve a common goal [1], thus trying to improve the effectiveness, efficiency, performance and robustness of a robotic system. These systems provide greater flexibility in performing tasks and possible fault tolerance. To achieve that several robots coordinate with each other to perform a specific mission is not a trivial task, because, they must be designed to operate in dynamic environments in which we must also take into account the classical problems of autonomous robotics (e.g. uncertainty and unforeseen changes always present), new difficulties arising from the influence of the team robots on the environment and the task goal.

Currently, intensive research work is being focused on MRS; the interest in this field is that, these systems offer several advantages and a higher potential with respect to the use of a single robot in performing many tasks. We considered the taxonomy proposed in [2] about of task allocation methods for multi-robot systems, and our work is located in the category instantaneous assignment vs time-extended assignment because the available information concerning the robots, the tasks, and the environment permits only an instantaneous allocation of tasks to robots, with no planning for future allocations.

Some works focus mainly on the coordination of a set of robots using different techniques, in order to solve a specific problem. With regard to the optimal tasks allocation problem, we describe some strong potential research articles related to the coordination of multi-agent systems, swarm robots and multi-robot systems. For example, in [2] Price suggests a number of strategies to address problems of task allocation in multi-agent systems, based on the principle of self-organization of social insects through the mathematical model developed by Bonabeau. In the study by Yang et al. proposed a foraging mission in swarm robots using a task allocation algorithm based on the response threshold model developed by Bonabeau, the experiments have been implemented in TeamBots [4]. In [5] threshold functions were successfully applied to the self-distribution of tasks problem in MRS.

One approach based on auction market mechanisms for the coordination of multi-robot systems was introduced by Dias and Stentz [6] in 2000. They consider that in multi-robot systems based on auctions, the robots are designed as agents of their own interests operating in a virtual economy. The tasks are assigned to the robots through the auction market mechanisms, for each task the complete robot generates some income that are reflected in the form of virtual money for providing a service to the team. However, when executing a task, the robot consumes resources such as fuel or network bandwidth, therefore, requires some expenses to pay for the resources used to complete the task. In 2004 [7] Dias has developed a coordination mechanism called *Traderbots*, which is designed to inherit the effectiveness and flexibility of a market economy. In this approach, were made some improvements in relation to the estimated costs to improve the efficiency of the team, then, in 2006 [8] this mechanism was applied in teams of robots to search treasure in an unknown environment.

The research presented by Song et al. proposed a distributed bidirectional auction algorithm for multi-robot systems coordination. A task is divided into n sub-tasks, a robot can only run a sub-task, the allocation of sub-tasks is decided by both the auctioneer and bidder; the auctioneer chooses the pre-winners ordering the prices of offer, while the bidders chosen all tasks that pre-won the sub-task which has the lowest price. After the first round, the sub-tasks that were not chosen by any bidder enters a second round of auction depending on the initial price auction, this process is repeated until all sub-tasks have been completed [9]. Shiroma and Campos proposed a framework for coordination and distribution of tasks between a set of heterogeneous mobile robots called CoMuTaR (Coalition formation based on Multi-tasking robots), allowing the robots

to perform multiple tasks at the same time. It is based on the Contract Net Protocol to form coalitions concurrent through actions, use an auction process of a single round. They considered two specific experiments: (1) that two robots cooperate to push a box and (2) that a set of three tasks are performed by two robots [10].

Another important approach is presented in [11], they have proposed a coordination approach to swarm robots both navigation and task allocation based on RFID (Radio Frequency Identification, RFID). RFID devices are distributed a priori in the environment by building a navigation chart; each RFID device contains navigation instructions that allow the robots to run the routes from one place to another. Robots cannot communicate with each other, but may do so indirectly by writing and reading RFID devices. To perform the distributed task allocation algorithm defines an auction, where the central server takes work to be undertaken by a team of robots, analyzes and decides the number of robots, then robots are informed about the new tasks and the allocation is the result of negotiations that each robot makes of its own. Similarly using RFID devices to communicate, leaving registration messages between them, for example, messages and records assignments and out of zones.

In MRS, optimal task/job allocation or assignment is an active research problem, in which several central or global allocation methods have been proposed [12]. The probabilistic approaches have been used to solve major challenges of mobile robotics, getting some new and innovative solutions to important problems such as navigation, localization, tracking and robot control. This approach could be applied to the problem of coordinating multiple robots to the self-election of heterogeneous specialized tasks and lends itself naturally to the automata learning-based probabilistic algorithm.

2 Formal Definitions

2.1 Formal Description of the Problem

The optimal multi-task allocation problem in multi-robot systems can be formally defined as follows: “Given a robot team formed by N heterogeneous robots, and given K different types of heterogeneous specialized tasks or equivalently, given K different robots roles or robots jobs and given a particular time-dependent load or number of tasks to be executed $L = \{l_1(t), l_2(t), \dots, l_K(t)\}$ ”. Obtain an optimal distribution of the K Tasks/Jobs among the N robots in such a way that the robots themselves, autonomously and in an individual manner, select a particular Job/Task such that all the existing tasks are optimally executed.

Let L denote the task, which divided into j sub-tasks $\{l_1, l_2, \dots, l_K\}$ where $j = 1, 2, \dots, K$ tasks. Let $R = \{r_1, r_2, \dots, r_N\}$ be the set of N heterogeneous mobile robots. To solve the problem, we assumption that all members that all members $R = \{r_1, r_2, \dots, r_N\}$ are able to participate in any sub-task l_j .

2.2 Experimental Scenario

We have established the following experimental scenario (Fig. 1), with the objective of analyzing a concrete strategy or solution for the coordination of multi-robot systems as regards the optimal distribution of the existing tasks. Given a set of N heterogeneous mobile robots in a region, achieving an optimal distribution for a set of T tasks of different types, each task can be divided into j sub-tasks. So that the set of N robots to form sub-teams of the task j . The sub-teams are dynamic over time, i.e. the same robots will not be always part of the same sub-team, but the components of each sub-team can vary depending on the situation. Most of the proposed solutions in the technical literature are

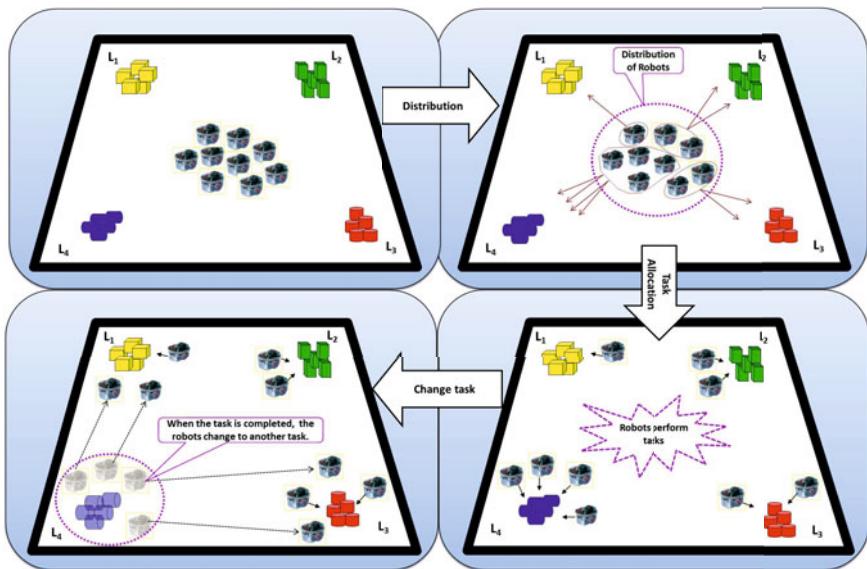


Fig. 1. Experimental Scenario

of a centralized nature, in the sense that an external controller is in charge of distributing the tasks among the robots by means of conventional optimization methods and based on global information about the system state [13]. However, we are mainly interested on truly decentralized solutions in which the robots themselves, autonomously and in an individual and local manner, select a particular task so that all the tasks are optimally distributed and executed. In this regard, we have experimented with stochastic reinforcement learning algorithms based on Learning Automata theory to tackle this hard self-coordination problem as described in the sequel.

3 Learning Automata-Based Probabilistic Algorithm

3.1 Learning Automata: A Brief Introduction

Learning automata have made a significant impact and have attracted a considerable interest in last years [14]. The first researches on learning automata models were developed in Mathematical Psychology, that describe the use of stochastic automata with updating of action probabilities which results in reduction in the number of states in comparison with deterministic automata. They can be applied to a broad range of modeling and control problems, control of manufacturing plants, pattern recognition, path planning for manipulator, among other. An important point to note is that the decisions must be made with very little knowledge concerning of the environment, to guarantee robust behavior without the complete knowledge of the system. In a purely mathematical context, the goal of a learning system is the optimization of a function not known explicitly [15].

Learning is defined as any permanent change in behavior as a result of past experience, an automata is a machine or control mechanism designed to automatically follow a predetermined sequence of operations or respond to encoded instructions [16]. The objective of stochastic learning automata is to determine how the choice of the action at any stage should be guided by past actions and responses, so when a specific action is performed the environment provides a random response which is either favorable or unfavorable [17].

3.2 Basic Definitions

A learning automaton is a sextuple $\langle x, Q, u, \vec{P}(t), G, \mathcal{R} \rangle$, where x is the finite set of inputs, $Q = \{q_1, q_2, \dots, q_m\}$ is a finite set of internal states, u is the set of outputs, $\vec{P}(t) = \|p_1(t), p_2(t), \dots, p_m(t)\|$ is the state probability vector at time instant t , $G : Q \rightarrow u$ is the output function (normally considered as deterministic and one-one), and \mathcal{R} is an algorithm called the reinforcement scheme, which generates $\vec{P}(t+1)$ from $\vec{P}(t)$ and the particular input at a discrete instant t .

The automaton operates in a random environment and chooses its current state according to the input received from the environment. The new state probabilities distribution $\vec{P}(t+1)$ reflects the information obtained from the environment. The random environment has a set of inputs u and its set of outputs is frequently binary $\{0, 1\}$, with ‘0’ corresponding to the reward response and ‘1’ to the penalty response. If the input to the environment is u_i the environment produces a penalty response with probability c_i .

Fig. 2 shows the feedback configuration of a learning automaton operating in a random environment. At each instant t the environment evaluates the action of the automaton by either a penalty ‘1’ or reward ‘0’. The performance of the automaton’s behaviors is the average penalty

$$I(t) = \frac{1}{m} \sum_{i=1}^m p_i(t)c_i \quad (1)$$

which must be minimized. In order to minimize the expectation of penalty (1), the reinforcement scheme modifies the state probability vector \vec{P} . The basic idea is to increase p_i if state q_i generates a reward and to decrease p_i when the same state has produced a penalty. A great number of reinforcement schemes for minimizing the expected value of penalty have been studied and compared. One of the most serious difficulties that arise in learning automata is the dichotomy between learning speed and accuracy. If the speed of convergence is increased in any particular reinforcement scheme, this action is almost invariably accompanied by an increase of convergence to the undesired state [18,19].

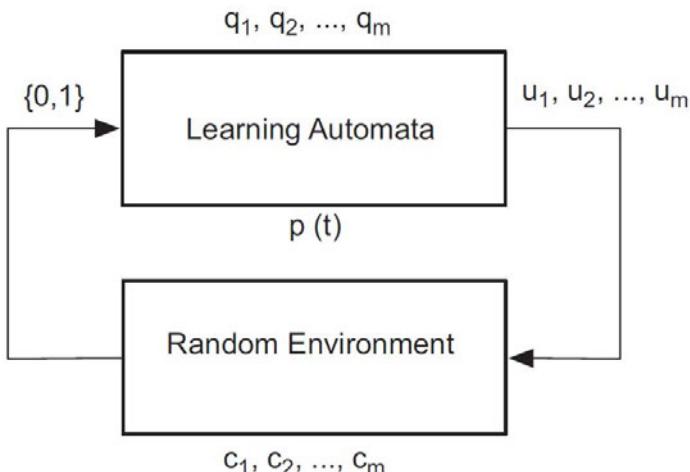


Fig. 2. Interaction of learning automaton with random environment

3.3 Stochastic Reinforcement Algorithms in Learning Automata Theory

In the technical literature a widely used stochastic reinforcement algorithms is L_{R-I} , which stands for Linear Reward-Inaction algorithm.

Let us suppose that the action chosen by the automaton at instant k is ϕ_i , for the L_{R-I} the updating of the action probabilities is as follows:

$$p_i(k+1) = p_i(k) + \lambda\beta(k)[1 - p_i(k)] \quad (2)$$

$$p_j(k+1) = p_j(k) - \lambda\beta(k)p_j(k) \quad \forall j \neq i, 1 \leq j \leq N \quad (3)$$

where $0 < \lambda < 1$ is the learning ratio and $\beta(k)$ is the environment's response: $\beta = 1$ (favorable response or reward) and $\beta = 0$ (unfavorable response or penalty in which case the algorithm do not change the probability, i.e. inaction). Let's suppose that there are K different specialized tasks, then we designate by $p_{ij}(k)$, the probability at instant k that robot R_i selects task L_j these probabilities hold:

$$0 \leq p_{ij}(k) \leq 1; \sum_{i=1}^N p_{ij}(k) = 1; i = 1, 2, \dots, N \text{ robots}; j = 1, 2, \dots, K \text{ tasks}$$

Initially, without previous robot's experience these probabilities are initialized at the "indifference" position as follow:

$$p_{ij}(0) = \frac{1}{K} \text{ for } i = 1, 2, \dots, N \text{ robots and } j = 1, 2, \dots, K \text{ tasks}$$

Afterwards it starts the learning process in which each robot updates its election probabilities according to the following conventional updating rule:

$$P_{ij}(k+1) = P_{ij}(k) + \lambda \beta(k) [1 - P_{ij}(k)] \quad (4)$$

where $0 < \lambda < 1$ is the learning rate with a fixed value of 0.2; $\beta(k)$ is the usual reward signal generated by the environment of the learning automata with the following interpretation: $\beta(k) = 1$; reward if and only if for the corresponding task T_j at instant k it holds that $\#R_j(k) \leq \#L_j(k)$, i.e. the number of robots performing task T_j is lower than the number of tasks T_j to be executed; $\beta(k) = 0$; penalty if and only if $\#R_j(k) \geq \#L_j$; i.e. when the number of robots performing task T_j is greater than the number of tasks T_j or whenever there are not pending tasks to be executed the automata receives a penalty signal. In few words: at each instant t the environment evaluates the action of the automata, when the response generated by enviroment is 1 means that the action is "favorable" and if the response value is 0 corresponds to an "unfavorable" as follow:

$$\beta_{L_j}(k) = \frac{\#R_j}{\#L_j} = \left\{ \begin{array}{l} \text{If } \frac{\#R_j}{\#L_j} \leq 1 \text{ then reward } \beta=1 \\ \text{If } \frac{\#R_j}{\#L_j} > 1 \text{ then penalty } \beta=0 \end{array} \right\}$$

4 Experimental Results

We have carried out a series of experiments to evaluate the system performance index through automata learning-based probabilistic algorithm, to solve the optimal distribution of the tasks among the N robots; so that all of them are executed by means of the minimum number of robots. The ideal objetive is that the performance index or learning curve corresponding to the load $L_j(t)$ of each task tend asymptotically to zero to that all curves in the minimun time and using the minimal possible number of robots for task execution.

In the simulations we have considered some variants such as: the multi-robot system size, different loads $L_j(t)$ for each type of task and the election of tasks for each robot. There are various methods of selection, to carry out the selection of tasks according to the results obtained with equation 4, we use two different methods to analyze the results of the algorithm:

1. The most likely: at each instant t choose the task that has the highest probability for all $P_{ij}(k)$.

2. The method of roulette: each probability $P_{ij}(k)$ at the instant t is divided into sectors, the sector with the largest area corresponds to the tasks that are most likely to be selected.

Fig. 3 shows the evolution of the system performance index obtained with the probabilistic algorithm for self-election of heterogeneous specialized tasks using the method of roulette and the most likely, with a team of robots formed by 20 – 50 heterogeneous robots and 4 types of heterogeneous specialized tasks with different loads:

- (a) $L = \{j_0 = 148, j_1 = 390, j_2 = 36, j_3 = 112\}$ and 20 robots
- (b) $L = \{j_0 = 266, j_1 = 365, j_2 = 419, j_3 = 150\}$ and 30 robots
- (c) $L = \{j_0 = 161, j_1 = 160, j_2 = 413, j_3 = 218\}$ and 40 robots
- (d) $L = \{j_0 = 497, j_1 = 88, j_2 = 160, j_3 = 249\}$ and 50 robots

It can be observed that learning curves corresponding to the load $L_j(t)$ of each task tend asymptotically to zero to both methods.

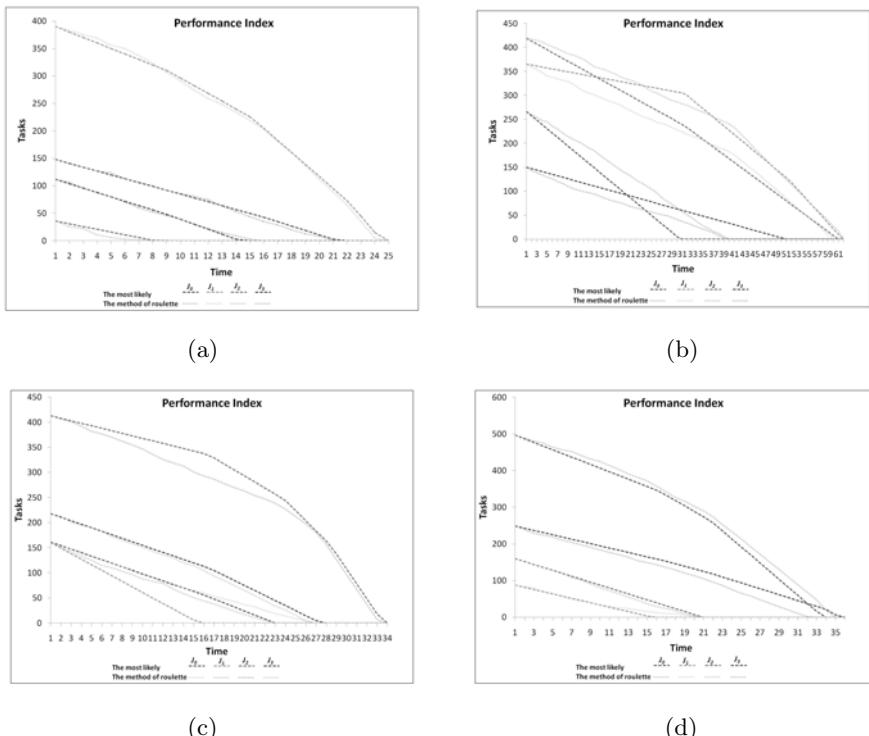


Fig. 3. Learning curves with the evolution of the system performance index for auto-election of tasks using the method of roulette and the most likely for different loads

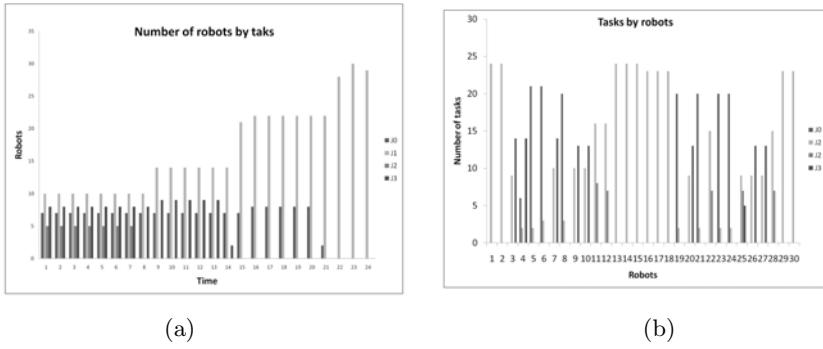
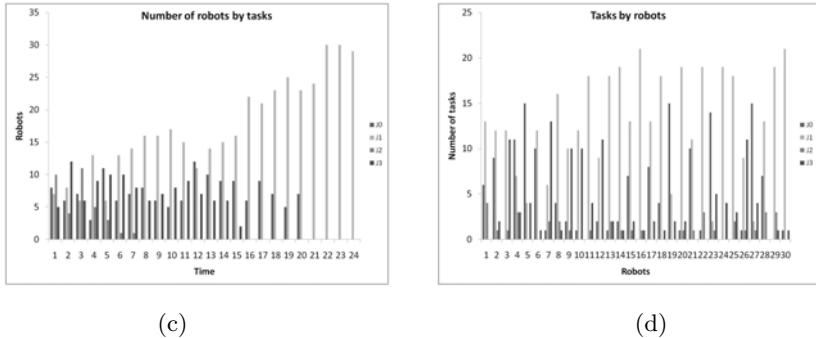
**Fig. 4.** The most likely**Fig. 5.** The method of roulette

Fig. 4 (a) shows the distribution of robots for each task l_j at the instant t , whereas Fig. 4 (b) shows a summary of the number of tasks l_j performed by each robot R_i through the method of the higher probability. Similarly, Fig. 5 (a) and (b) shows the number of robots in each task l_j and the number of tasks performed by each robot R_i using the method of roulette, respectively.

5 Conclusions and Further Work

In this paper we have proposed the automata learning-based probabilistic algorithm, applied to the self-coordination problem of multi-robot systems in the heterogeneous multi-tasks distribution. We have established an experimental scenario in which several heterogeneous and specialized tasks distributed to be executed by a team of heterogeneous mobile robots without the intervention of any global and central tasks scheduler. The results confirm that the robots are capable to select in an autonomous and individual manner the existing tasks without the intervention of any global and central tasks scheduler. We have shown that the algorithm can be efficiently applied to solve this self-coordination problem in multi-robot systems obtaining truly decentralized solutions.

We will study the evaluation of the robustness of probabilistic algorithm perturbing the number of pending load, to simulate the robot's error in estimating the real number of pending tasks. We also plan to experiment in the future with alternative methods like Ant Colony Optimization-based deterministic algorithms to compare both method.

References

1. Farinelli, A., Locchi, L., Nardi, D.: Multirobot Systems: A Classification Focused on Coordination. *IEEE Transactions on Systems, Man and Cybernetics*, 2015–2028 (2004)
2. Gerkey, B., Mataric, M.: A formal analysis and taxonomy of task allocation in multi-robot systems. *Intl. J. of Robotics Research*, 939–954 (2004)
3. Price, R., Tiño, P.: Evaluation of Adaptive Nature Inspired Task Allocation Against Alternate Decentralised Multiagent Strategies. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiño, P., Kabán, A., Schwefel, H.-P. (eds.) *PPSN 2004. LNCS*, vol. 3242, pp. 982–990. Springer, Heidelberg (2004)
4. Yang, Y., Zhou, C., Tian, Y.: Swarm Robots Task Allocation Based on Response Threshold Model. In: *IEEE International Conference on Autonomous Robots and Agents*, pp. 171–176 (2009)
5. Quiñonez, Y., De Lope, J., Maravall, D.: Bio-Inspired Decentralized Self-Coordination Algorithms for Multi-Heterogeneous Specialized Tasks Distribution in Multi-Robot Systems. In: Ferrández, J.M., Álvarez Sánchez, J.R., de la Paz, F., Toledo, F.J. (eds.) *IWINAC 2011, Part I. LNCS*, vol. 6686, pp. 30–39. Springer, Heidelberg (2011)
6. Dias, B., Stentz, A.: A Free Market Architecture for Distributed Control of a Multirobot System. In: *6th International Conference on Intelligent Autonomous Systems*, pp. 115–122 (2000)
7. Dias, B.: Traderbots: A new Paradigm for Robust and Efficient Multirobot Coordination in Dynamics Environments. Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh (2004)
8. Jones, E., Browning, B., Dias, B., Argall, B., Veloso, M., Stentz, A.: Dynamically Formed Heterogeneous Robot Teams Performing Tightly-Coordinated Tasks. In: *IEEE Conference on Robotics and Automation*, pp. 570–575 (2006)
9. Song, T., Yan, X., Liang, A., Chen, K., Guan, H.: A Distributed Bidirectional Auction Algorithm for Multirobot Coordination. In: *IEEE International Conference on Research Challenges in Computer Science*, pp. 145–148 (2009)
10. Shiroma, P., Campos, M.: CoMutaR: A Framework for Multi-robot Coordination and Task Allocation. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4817–4824 (2009)
11. Baglietto, M., Cannata, C., Capezio, F., Grossi, A., Sgorbissa, A.: A Multi-robot Coordination System Based on RFID Technology. In: *IEEE International Conference on Advanced Robotics*, pp. 1–6 (2009)
12. Huntsberger, T., Trebi-Ollennu, A., Aghazarian, H., Schenker, P.: Distributed Control of Multi-robot System Engaged in Tightly Coupled Tasks. *Journal Autonomous Robots*, 79–92 (2004)
13. Gerkey, B., Mataric, M.: Multi-Robot Task Allocation: Analyzing the Complexity and Optimality of Key Architectures. In: *IEEE International Conference on Robotics and Automation*, pp. 3862–3868 (2003)

14. Narendra, K., Thathachar, M.: Learning Automata: An Introduction. Prentice-Hall, Englewood Cliffs (1989)
15. Narendra, K., Thathachar, M.: Learning Automata: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 323–334 (1974)
16. Obaidat, M., Papadimitriou, G., Pomportsis, A.: Guest Editorial Learning Automata: Theory, Paradigms, and Applications. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 706–709 (2002)
17. Maravall, D., De Lope, J.: Fusion of Learning Automata Theory and Granular Inference Systems: ANLAGIS. Applications to Pattern Recognition and Machine Learning. *Neurocomputing* 74, 1237–1242 (2011)
18. Narendra, K., Wright, E., Mason, L.: Applications of Learning Automata to Telephone Traffic Routing and Control. *IEEE Transactions on Systems, Man, and Cybernetics*, 785–792 (1977)
19. Narendra, K., Viswanathan, R.: A Two-Level System of Schotastic Automata for Periodic Random Environments. *IEEE Transactions on Systems, Man, and Cybernetics*, 285–289 (1972)

Stochastic Abstract Policies for Knowledge Transfer in Robotic Navigation Tasks

Tiago Matos, Yannick Plaino Bergamo, Valdinei Freire da Silva,
and Anna Helena Reali Costa

Laboratório de Técnicas Inteligentes (LTI/EPUSP)
Escola Politécnica, Universidade de São Paulo
São Paulo, SP, Brazil

{tiago.matos,yannick}@usp.br, valdinei.freire@gmail.com,
anna.reali@poli.usp.br

Abstract. Most work in navigation approaches for mobile robots does not take into account existing solutions to similar problems when learning a policy to solve a new problem, and consequently solves the current navigation problem from scratch. In this article we investigate a knowledge transfer technique that enables the use of a previously known policy from one or more related source tasks in a new task. Here we represent the knowledge learned as a stochastic abstract policy, which can be induced from a training set given by a set of navigation examples of state-action sequences executed successfully by a robot to achieve a specific goal in a given environment. We propose both a probabilistic and a nondeterministic abstract policy, in order to preserve the occurrence of all actions identified in the inductive process. Experiments carried out attest to the effectiveness and efficiency of our proposal.

1 Introduction

Automated Planning and Learning are essential components of intelligent behavior. Broadly speaking, they deal with the methods by which an intelligent agent can determine the policy (action strategy) needed to successfully achieve a goal. However, often the determination of a policy for solving a certain task has to be done without any use of the acquired knowledge to solve similar tasks. Intuitively, generalization from closely related, but solved, problems may often produce policies that make good decisions in many states of a new unsolved problem. Knowledge transfer explores this intuition and could then be used in these situations. The core idea of knowledge transfer is that experience gained in performing one task can help solve a related, but different, task.

In this paper, we are concerned with the problem of transfer the knowledge that was learned in one problem to another problem so that this knowledge gives effective guidance when solving the new problem. Here we represent the knowledge learned as a stochastic abstract policy, which can be learned from a training set given by a set of state-action pairs obtained by solving one or more instances of a given problem. Such policies are not optimal but they are general and they can be applied effectively, producing good results.

More precisely, our approach starts from a set of navigation examples of state-action sequences executed successfully by a robot to achieve a specific goal in a given environment. These examples can be achieved through the execution of a plan developed to solve this task or provided by a teacher. Then, we induce a stochastic navigation policy in the form of a relational decision tree from these examples. This relational decision tree represents an abstract policy, which expresses preferences over navigation actions and can be used by the robot to decide the navigation actions both in the original task and in similar tasks or domains.

Our contribution lies in an extension of the work of Kersting et al. [1], so as not to consider in the leaf of the relational decision tree only the abstract action more frequently found in the inductive process, but to use a stochastic abstract action. We propose both a probabilistic and a nondeterministic abstract policy in order to preserve the occurrence of all ground actions identified in the inductive process.

Several papers have been looking at ways to abstract the knowledge acquired in solving a problem and use it to solve other problems. Taylor and Stone [2] provide a thorough survey on the transfer of learning using propositional descriptions of the states. In several works the source task is learned in some ground state space where an abstract policy is generated, and this policy is applied to the target task in order to improve learning performance, especially in the early episodes. Based on samples experienced by the agent when acting optimally in the source task, in [3] propositional symbolic learners generalize the optimal policy. In [4] action relevance is used in order to reduce the action set to be explored in the target task. In [5] a heuristic policy is used as a knowledge to be applied in the policy learning process in a new task or domain.

Regarding the first order state description, few examples are found in the literature for the domain of spatial navigation. In order to fully describe the relational situation in which an agent is, one must map the environment completely. So as not to do so, [1] and [6] rely on relational local perception and structured environments, where (near-) optimal abstract policies can be defined. Here we propose two different approaches for transfer knowledge from one problem to another, and this knowledge is represented by abstract policies that are based on local relational perceptions made in structured environments.

The remainder of this paper is structured as follows. In Section 2 we give an overview of Relational Markov Decision Processes and show how to learn and use abstract navigation policies. Section 3 covers a detailed specification of our approach, describing how to learn and to use stochastic abstract policies. We present experimental results and analysis in Section 4 and discuss the implications of these. Finally, in Section 5 we discuss how this technique can be expanded in the future, and draw conclusions based on what we have outlined.

2 Learning Relational Navigation Policies

Several domains are naturally described in terms of objects and relations among objects, therefore they can be compactly described with relational representations.

On the other hand, stochastic domains, such as robotic navigation domains that concern us here, are properly modeled by Markov Decision Processes. Relational Markov Decision Processes (RMDPs) combine relational logic with MDPs to powerfully exploit relational structures. An RMDP describes a mathematical model of interactions between an agent and a stochastic, dynamic environment when the world is described by objects and relations among them. The advantage of the relational representation is abstraction.

We give here a very brief review of the key aspects of RMDPs and their solutions. For a thorough treatment, we refer the reader to [7,8].

2.1 Relational Markov Decision Process

The key idea in Relational Markov Decision Processes (RMDPs) [8] is to merge relational logic with MDP. Let $\mathcal{P} = \{p_1/\alpha_1, \dots, p_n/\alpha_n\}$ be a set of predicates p_i with their arities α_i , $\mathcal{C} = \{c_1, \dots, c_k\}$ a set of constants, and $\mathcal{A}' = \{a_1/\alpha_1, \dots, a_m/\alpha_m\}$ a set of actions a_i with their arities α_i where $a_i = H_{a_i} \xleftarrow{p_{a_i}:A_{a_i}} B_{a_i}$, and A_{a_i} is an atom representing the name and the arguments of the action a_i , B_{a_i} is the precondition, H_{a_i} is a state representing the successful outcome, and p_{a_i} is the probability that an outcome succeed. An atom is called ground if it contains no variables. Let \mathcal{S}' be the set of all conjunctions of ground atoms over \mathcal{P} and \mathcal{C} , and \mathcal{A} the set of all ground atoms over \mathcal{A}' and \mathcal{C} . An RMDP is a tuple $M = \langle \mathcal{S}, \mathcal{A}, T, R \rangle$, where \mathcal{S} is a subset of \mathcal{S}' , $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is a probabilistic transition function, and $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow$ is a reward function. For a given state $s \in \mathcal{S}$ there is a corresponding set of feasible actions $\mathcal{A}(s) \subseteq \mathcal{A}$ which is calculate through the action precondition.

The task of the agent is to learn a policy $\pi : \mathcal{S} \times \mathcal{A}$ for selecting its next action $a \in \mathcal{A}(s)$ to be performed on the current state $s \in \mathcal{S}$, i. e., $\pi(s) = a$. The learned policy should maximize $V^\pi(s) = E[\sum_{t=0}^{\infty} \gamma^t r_t | \pi, s_0 = s]$, where $\gamma \in [0, 1]$ is a discount factor and r_t is the reward received in time t , i. e., the policy π^* is optimal if $V^{\pi^*}(s) \geq V^{\pi'}(s), \forall s \in S$ and $\forall \pi'$.

If the model of the problem is fully known, we can use planning techniques such as SPUDD [9] to determine π^* , otherwise we can use reinforcement learning techniques [10] for this purpose.

2.2 Modeling a Robotic Navigation Problem as an RMDP

A robotic navigation problem can elegantly be represented using an RMDP. Consider¹ the simple environment shown in Figure 3(a): we define a set of $n = 19$ constant locations $\mathcal{C} = \{11, 12, \dots, 119\}$. Unary predicates indicate the location type such as `isCorridor(Li)`, `isRoom(Li)`, `isCenter(Li)`, `isNearDoor(Li)`,

¹ Squares symbolize centers of rooms, e. g. (`isRoom(l2) ∧ isCenter(l2)`), triangles symbolize doors of rooms, e. g. (`isRoom(l3) ∧ isNearDoor(l3)`), circles symbolize doors of corridors, e. g. (`isCorridor(l5) ∧ isNearDoor(l5)`), and black pentagons symbolize corridors, but not in front of a door – here, for simplicity, we call center, e. g. (`isCorridor(l1) ∧ isCenter(l1)`).

whereas a binary predicate `isConnected(Li,Lj)` indicates that L_i is connected to L_j . A special unary predicate `in(Li)` indicates the location of the robot. Terms starting in lowercase represent constants, and in uppercase are variables.

We consider a set \mathcal{A}' of specialized navigation actions that move the robot from one location to another, e. g., `gotoRCRD(Li,Lj)` moves the robot from L_i in the center of a room to a location L_j near a door in a room, `gotoRDCD(Li,Lj)` moves the robot from a location L_i near a door in a room to a location L_j near a door in a corridor, etc. The generic action `gotoXXYY(Li,Lj)` shortens the conditions required for the current location XX and the next location YY , e. g., $XX = RC$ when $(\text{in}(L_i) \wedge \text{isRoom}(L_i) \wedge \text{isCenter}(L_i))$, $XX = CD$ when $(\text{in}(L_i) \wedge \text{isCorridor}(L_i) \wedge \text{isNearDoor}(L_i))$ and so on; likewise, $YY = CC$ when $(\text{isCorridor}(L_i) \wedge \text{isCenter}(L_i))$, and so on.

The precondition of `gotoXXYY(Li,Lj)` is $(\text{in}(L_i) \wedge \text{isConnected}(L_i,Lj))$ and both locations L_i and L_j attend the required conditions XX and YY .

If the result of an action is successful the robot reaches the desired location ($\text{in}(L_j)$) with probability p , and if the action fails the robot remains in the same location ($\text{in}(L_i)$) with probability $1 - p$.

The use of RMDPs offers many possibilities for abstraction. An abstract state X is defined by the conjunction of predicates that have at least one term as a variable. The ground state $s \in S$ is an instance of X if there is a substitution $\theta = \{V_1/t_1, \dots, V_k/t_k\}$ such that $X\theta = s$, where V_i is a variable and t_i is a term. For instance, given the abstract state $X = (\text{in}(L) \wedge \text{isRoom}(L) \wedge \text{isNearDoor}(L))$ and the substitution $\theta = \{L/11\}$, then the state $s = X\theta = (\text{in}(11) \wedge \text{isRoom}(11) \wedge \text{isNearDoor}(11))$ is an instance of X . Abstract actions can be described similarly by introducing variables in their arguments, preconditions, and postconditions.

2.3 Generating an Abstract Policy

An abstract policy is a mapping from abstract states to abstract actions. An abstract policy captures a generalization of optimal (or sub-optimal) policies according to the structural characteristics of the problem. Abstract policies may be induced from a set of examples E . Here, examples are formed by state-action pairs taken from a set of navigation experiences generated by applying an optimal policy. Given a goal state in the robot navigation problem, we find an optimal policy in the ground MDP. Then we define a set of initial states from which the robot acts optimally to generate state-action sequences that lead to the goal state [1]. Each state-action pair of these sequences is inserted in E . The task then is to induce an abstract policy based on E .

TILDE is an inductive algorithm that makes abstraction of the experiences, and represents the abstract policy induced in a first order logical decision tree (FOLDT) [11]. FOLDT is an adaptation of a decision tree for first order logic where the tests in the nodes are conjunctions of first order literals. The TILDE algorithm [11] is shown in Algorithm 1. TILDE creates a FOLDT based on the set of training examples E . Test candidates are created (step 2 of the algorithm) from a set of refinement operators [11], which are previously defined by an expert.

Each refinement operator generate a set of first order literals as test candidate for the division of the set of examples E . The best test to divide the set E is the test that reduces a measure of entropy. The optimal test is chosen (step 3) using the default gain ratio [12]. If the partition induced on E indicates that the division should stop (procedure STOP_CRIT in step 5), a leaf is created. The tree leaves created (step 6) contain atomic sentences that represent abstract actions. If more than one atomic sentence represents the remaining examples in E , TILDE chooses the atomic sentence that represents the largest number of examples. If the partition induced on E does not indicate that the division should stop, for each one of the partitions induced (step 8) the function TILDE is recursively called (step 9). An internal node is created using the optimal test τ as test, and the two children are the sub-trees induced by each call of TILDE (in step 9). Figure 1 shows the FOLDT induced from the examples generated by applying the optimal policy for the task of the robot going from any location L to location l2 in the environment shown in Figure 3(a).

Algorithm 1. Algorithm TILDE

```

1: function TILDE (E: set of examples): returns a decision tree
2:   T = GENERATE_TESTS_FOL(E)
3:    $\tau$  = OPTIMAL_SPLIT_FOL(T,E)
4:    $\epsilon$  = partition induced on E by  $\tau$ 
5:   if STOP_CRIT(E, $\epsilon$ ) then
6:     return leaf(INFO_FOL(E))
7:   else
8:     for all  $E_j$  in  $\epsilon$  do
9:        $t_j$  = TILDE( $E_j$ )
10:      end for
11:      return inode( $\tau$ , ( $j, t_j$ ))
12:    end if
13: end function

```

2.4 Using the Abstract Policy to Guide a Robot

The FOLDT induced by TILDE represents an abstract policy to the robot navigation problem. Each node of the FOLDT captures a logical test, which either succeeds or fails when applied to a particular (ground) state. If it succeeds, we use the left subtree; otherwise the right one. Leaves in the FOLDT induced by TILDE represent a single abstract action to be taken in the abstract state, which is defined by the conjunction of all preceding tests from the root to the leaf. Note that each abstract action can represent a set of ground actions.

We perform a forward search guided by the learned abstract navigation policy. The robot starts in some location, observes the current (ground) state, and decides the (ground) action to perform next by evaluating the abstract policy on the observed state. The observed ground state imposes a substitution θ to the abstract state and to the abstract action suggested by the FOLDT. In order to

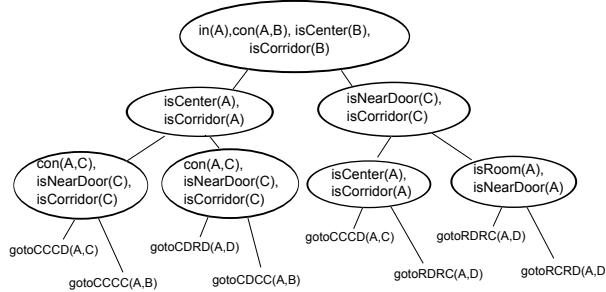


Fig. 1. An abstract policy induced by the TILDE algorithm for the navigation problem in the environment shown in Figure 3(a). The conjunction of predicates in a path from the root to the leaf defines an abstract state. Tree leaves define abstract actions.

decide the ground action to be performed next by the robot, we choose uniformly among all ground actions defined by the substitution θ to the abstract action suggested in the FOLDT leaf.

However, it may happen that the substitution θ to the chosen abstract action generates a set of ground actions $A\theta$ whose elements do not belong to the set of executable actions in the observed ground state, i. e., $A\theta \cap A(s) = \emptyset$. In this case we choose uniformly among all ground actions in the set $A(s)$. Finally, having decided what ground action to perform, the robot performs the action and repeats the observation-decision-actuation cycle.

Note that TILDE creates a tree that contains only one abstract action in each leaf. Unfortunately, important information is lost in this abstraction process. Therefore, the application of an abstract policy may cause some problems. In the next section we discuss these issues and propose a modification in the TILDE algorithm to overcome these problems.

3 Generating a Stochastic Abstract Policy

The process of generalization causes loss of some important information to solve the problem, since it groups together different actions to be applied in the same abstract state. For example, in the environment of the Figure 3(a) where the goal is to reach the location 12, the abstract state that groups together ground states defined by locations 15 and 18 will leads to the same tree leaf (represented by the third leaf from left to right in Figure 1). However, the training examples given when the robot was at location 15 indicated that the optimal ground action was `gotoCDRD(15, 13)`, and when the robot was at location 18 the optimal ground action was `gotoCDCC(18, 15)`. In this case, during the creation of the leaf, TILDE will choose the abstract action that represents the largest number of examples, and the distinction between these situations will be lost.

Also related to this same problem, if the abstract policy generated in this way is used to directly control the robot navigation, there may be cases in which the goal is never reached. To illustrate these cases, consider again the environment shown in Figure 3(a) and the abstract policy shown in Figure 1: this abstract policy will never drive the robot to the goal location 12 whenever he comes to the location 18, since the abstract policy will guide the robot to the location 17 – the abstract policy gives action `gotoCDRD(18,17)` – rather than to the location 15 (which leads to the goal).

Due to these facts we propose a modification to the abstraction algorithm TILDE that can handle the problems showed previously. The modification (named X-TILDE) enables the construction of a tree that represents a stochastic abstract policy. Each leaf now contains a set of abstract actions. We propose two versions of the X-TILDE algorithm: P-TILDE (Probabilistic TILDE) in which we associate a probability of choosing each abstract action belonging to a leaf, and ND-TILDE (Nondeterministic TILDE) in which there is no preference in the choice of an abstract action in a leaf. Kersting et al. [1] also noted some of these problems that we raised, and in order to circumvent them, they proposed the use of an auxiliary structure and some heuristics in the process of using the policy.

In the leaf level, the TILDE algorithm can face two situations: i) the actions of all the training examples corresponding to the same abstract state indicate the same abstract action; or ii) the actions of some of them suggest another abstract actions. In the first case, the choice is trivial, while in the latter case, the algorithm TILDE chooses to associate to the leaf the abstract action whose corresponding ground action occurred more often in the examples.

We propose changing TILDE so that less information is lost. Instead of assigning only one abstract action to each tree leaf (step 6 of the Algorithm 1), our proposal is to associate a set of abstract actions to each leaf (see Figure 2). We propose two variations of versions, called synthetically X-TILDE: P-TILDE for when X is P, and ND-TILDE in the other case. P-TILDE keeps the abstract actions and their respective frequency of occurrence in the tree leaf, and we make use of it whenever we reuse the abstract policy. On the other hand, the ND-TILDE keeps in the leaf all the abstract actions derived from the training examples, and they have an equal chance of being chosen when we reuse the abstract policy. Using the X-TILDE algorithm the problems previously cited are circumvented. The stochastic abstract policy learned for the states derived for locations 15 and 18 suggests two abstract actions as policy: `gotoCDRD(A,D)` and `gotoCDCD(A,C)`, as shown in Figure 2(c).

Therefore, when using the abstract policy we can envisage three different ways for the robot to decide which ground action to apply, depending on the algorithm used to generate the FOLDT:

- TILDE: to choose uniformly among all ground actions defined by the substitution θ to the abstract action suggested in the FOLDT leaf.

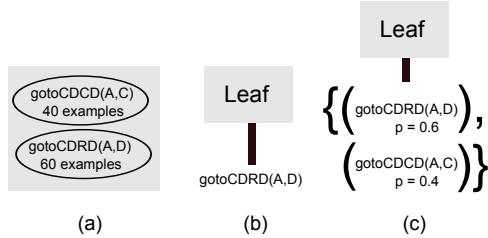


Fig. 2. (a) Number of examples and their respective action candidates for the third leaf (from left to right) of the FOLDT of Figure 1. (b) Leaf created by TILDE using only the abstract action `gotoCDRD(A,D)`. (c) Leaf created by P-TILDE, with both abstract actions, `gotoCDRD(A,D)` and `gotoCDCD(A,C)`, and their respective frequency of occurrence. In the case of ND-TILDE there is no value associated to each abstract action in the set of abstract actions in the leaf.

- P-TILDE: first to choose probabilistically among all abstract actions in the FOLDT leaf and then uniformly among all ground actions defined by the substitution θ to the chosen abstract action;
- ND-TILDE: first to choose uniformly among all abstract actions in the FOLDT leaf and then also uniformly among all ground actions defined by the substitution θ to the chosen abstract action.

4 Experiments

The states in the maps of Figures 3(a) and 3(b) were defined based on the map's morphological skeleton. Each state has a reference point which is the point where three or more segments of the morphological skeleton intersect. Having chosen the reference point, the states are delimited by the Voronoi diagram of the reference point, i.e., any given point (x, y) of the map will belong to the state S with the nearest reference point.

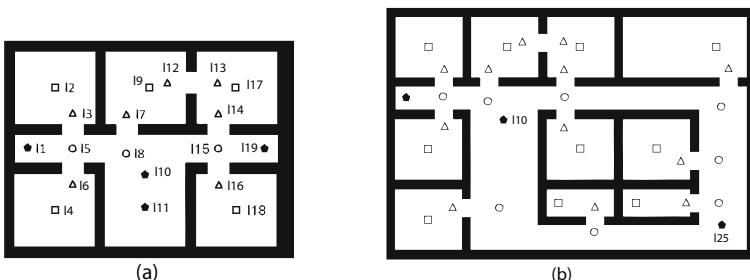


Fig. 3. (a) An example of a relational representation of a simple navigation problem. (b) New navigation problem used to show the policy reuse.

In all experiments it was used the same FOLDT, generated by the examples taken from the optimal policy application, whose goal was to reach the center of the upper right room of the map in Figure 3(a) (location 12).

The experiments were divided in three parts: in the first one we applied the abstract policy to the same environment and the same task used to build the tree; in the second one we kept the same environment but considered different goals; in the third one we considered a new environment (Figure 3(b)). In all cases we compare the abstract policies with a random policy which, for every given state s , the robot selects uniformly among the actions in the set $A(s)$ and executes it.

The use of the abstract policy in a new task can lead to situations that did not exist in the source problem used to induce the abstract policy. Because of this, the abstract policy can not suggest an adequate action for this situation. One occurrence of this problem occurs in the second set of experiments, where for the abstract state $(\text{in}(A) \wedge \text{isConnected}(A,B) \wedge \text{isCenter}(B) \wedge \text{isCorridor}(B))$ the abstract policy gives $\text{gotoCDCC}(A,B)$ as abstract action. This abstract action is appropriated in four of the goal states (12, 14, 19 and 117), but is not the correct action in the last one (118). In this situation, during the induction of the abstract policy the abstraction algorithm has not received an example covering this situation. Every time the location 115 is reached the abstract policy will guide the robot to a center of a corridor, and the correct goal 118 will never be reached.

The same kind of problem occurs in the third set of experiments. In this case the abstract policy faces a new situation where for the abstract state $(\text{in}(A) \wedge \text{isConnected}(A,B) \wedge \text{isCenter}(B) \wedge \text{isCorridor}(B) \wedge \text{isConnected}(A,C) \wedge \text{isNearDoor}(C) \wedge \text{isCorridor}(C))$ the abstract action $\text{gotoCDCC}(A,D)$ should be considered to allow the robot to reach all centers of rooms in the environment. The abstract policy lacks this information and because of this the environment is divided in two distinct regions: the first encompassing most of the top states and the second the lower states. Both regions are connected by the locations 110 and 125, but there is no path linking a location in one region to another location of the other region, i. e., there is only a path connecting locations of the same region. Because of this, in this set of experiments we selected the pairs of initial and goal states belonging to the same region.

By letting N be the number of transitions used to achieve the goal, we show for every experiment and for every policy a plot of the cumulative distribution of N , i.e. $P(N \leq t)$, the mean $\mu(N)$, the standard deviation $\sigma(N)$ and the number E_{succ} of episodes where the robot successfully reached the goal, with the corresponding total number of episodes E_{tot} .

Since the mean of N will depend on the relative distance of each $\langle \text{initialstate}, \text{goalstate} \rangle$ pair used for each episode, we consider as a measure of performance the ratio $\mu(N)_{X-TILDE}/\mu(N)_{RANDOM}$. Hence a number less than one will represent an advantage for the abstract policy with respect to the random policy. The lower this number, the better the performance.

4.1 Experiment 1: Same Environment – Same Goal State

In the first experiment we compare the abstract policies against the random policy in the same conditions that were used to generate the training examples in the FOLDT induction, i. e., the robot should navigate from any location to location 12 in the environment of Figure 3(a). We randomly selected five initial states for the task, and ran 2000 episodes for each one. Figure 4 shows the cumulative distribution, $\mu(N)$ and $\sigma(N)$ for each policy. The performance ration was: $\frac{\mu(N)_{P-TILDE}}{\mu(N)_{RANDOM}} = 0.69$ and $\frac{\mu(N)_{ND-TILDE}}{\mu(N)_{RANDOM}} = 0.60$.

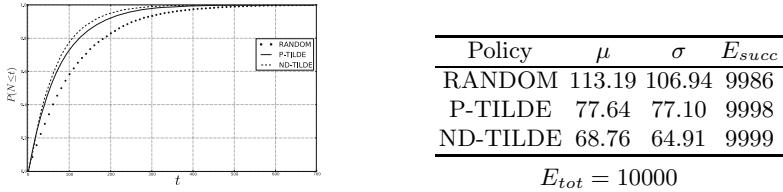


Fig. 4. Cumulative distribution, $\mu(N)$ and $\sigma(N)$ for the first set of experiments

4.2 Experiment 2: Same Environment – Different Goal States

In the second experiment we kept the same environment but changed the goal state. We chose as goal states the remaining reachable center of rooms in the environment shown in Figure 3(a) (14, 19, 117, 118). For each goal we ran 2000 episodes with randomly selected initial states. Figure 5 shows the cumulative distribution, $\mu(N)$ and $\sigma(N)$ for each policy. The performance ration was: $\frac{\mu(N)_{P-TILDE}}{\mu(N)_{RANDOM}} = 0.75$ and $\frac{\mu(N)_{ND-TILDE}}{\mu(N)_{RANDOM}} = 0.70$.

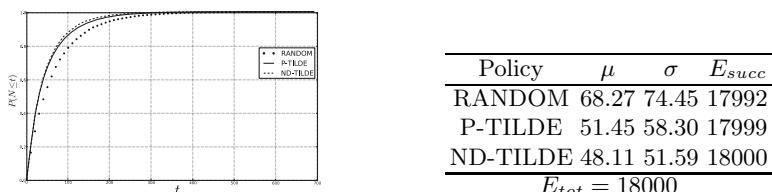
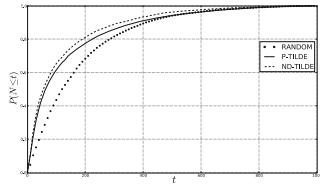


Fig. 5. Cumulative distribution, $\mu(N)$ and $\sigma(N)$ for the second set of experiments

4.3 Experiment 3: New Environment

For each $\langle initialstate, goalstate \rangle$ pair we ran 2000 episodes. Figure 6 shows the cumulative distribution, $\mu(N)$ and $\sigma(N)$ for each policy. The performance ratio was: $\frac{\mu(N)_{P-TILDE}}{\mu(N)_{RANDOM}} = 0.78$ and $\frac{\mu(N)_{ND-TILDE}}{\mu(N)_{RANDOM}} = 0.68$.



Policy	μ	σ	E_{succ}
RANDOM	177.97	172.73	7971
P-TILDE	138.84	176.39	7901
ND-TILDE	120.49	155.25	7974
$E_{tot} = 8000$			

Fig. 6. Cumulative distribution, $\mu(N)$ and $\sigma(N)$ for the third set of experiments

5 Conclusion and Future Works

We presented a new approach for the policy reuse in robot navigation problems. The policy reuse is obtained by using a FOLDT that represents an abstract policy. The FOLDT is induced using the TILDE algorithm. Our approach, named X-TILDE, extends the TILDE algorithm and creates an FOLDT that represents a stochastic abstract policy. This stochastic abstract policy retains more information about the structure of the navigation problems and solves situations that the abstract policy generated by the original TILDE algorithm can not solve.

The quality of the abstract policy generated by the TILDE algorithm depends on the structure of the problem and the relational description chosen. Whereas the TILDE algorithm does not provide solution to many problems, our experiments showed that our approach can solve a much larger set of problems. The experiments also showed that the abstract policy generated by X-TILDE has a better performance than a random policy in a new task, which leads us to conclude that there was indeed a transfer of knowledge between tasks.

As the experiments showed, the performance of P-TILDE is worse than the performance of ND-TILDE. We believe it is because the P-TILDE algorithm classifies states based on the ground policy, but aggregates them based only on their relational descriptions. Since ND-TILDE does not take into account the distribution of occurrence of actions, the results showed that it is better to use a more conservative approach. On the other hand, the random policy, which is the most conservative of all policies, showed worse results. This leads us to investigate in the future a better compromise between these policies and to find ways to determine the best balance between the approaches.

Despite being more robust, our approach does not guarantee that every task can be performed. This depends on the experience in the source task, and may fail in situations that did not occur in the source task. In [4] random perturbations are applied to the source task so that even in unrepresentative tasks we can have the transfer of knowledge robustly. Another important topic to be investigated is to define how the source problems should be selected, which should be simple but should able to generate useful policies for more complex tasks.

The main motivation for reusing a policy in a robot navigation domain is to guide the robot in solving similar problems. If the abstract policy learned is not perfect, the agent must learn from the knowledge that was transferred. In this case, the abstract policy can be used as a advice to guide the exploration in

RL algorithms. This combination aims to accelerate the learning process of an optimal political in the target task [13].

Acknowledgments. This research was partially supported by FAPESP (08/03995-5, 09/04489-9, 09/14650-1, 10/02379-9) and by CNPq (475690/2008-7, 305512/2008-0).

References

1. Kersting, K., Plagemann, C., Cocora, A., Burgard, W., Raedt, L.D.: Learning to transfer optimal navigation policies. *Advanced Robotics: Special Issue on Imitative Robots* 21, 1565–1582 (2007)
2. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10, 1633–1685 (2009)
3. Madden, M.G., Howley, T.: Transfer of experience between reinforcement learning environments with progressive difficulty. *Artif. Intell. Rev.* 21, 375–398 (2004)
4. Sherstov, A.A., Stone, P.: Improving action selection in MDP's via knowledge transfer. In: Proc. of the 20th National Conference on Artificial Intelligence (2005)
5. Bianchi, R., Ribeiro, C., Costa, A.: Accelerating autonomous learning by using heuristic selection of actions. *Journal of Heuristics* 14, 135–168 (2008), doi:10.1007/s10732-007-9031-5
6. Lane, T., Wilson, A.: Toward a topological theory of relational reinforcement learning for navigation tasks. In: Proc. of the 18th Int. Florida Artificial Intelligence Research Society Conference (2005)
7. Kersting, K., Otterlo, M.V., Raedt, L.D.: Bellman goes relational. In: Brodley, C.E. (ed.) Proc. of the 21st Int. Conference on Machine Learning, Banff, Alberta, Canada, pp. 465–472 (2004)
8. Otterlo, M.V.: The logic of adaptive behavior: knowledge representation and algorithms for the Markov decision process framework in first-order domains. PhD thesis, University of Twente, Enschede (2008)
9. Hoey, J., St-Aubin, R., Hu, A.J., Boutilier, C.: Spudd: Stochastic planning using decision diagrams. In: Proc. of Uncertainty in Artificial Intelligence, Stockholm, Sweden (1999)
10. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
11. Blockeel, H., Raedt, L.D.: Top-down induction of first-order logical decision trees. *Artificial Intelligence* 101, 285–297 (1998)
12. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
13. Matos, T., Bergamo, Y.P., da Silva, V.F., Cozman, F.G., Costa, A.H.R.: Simultaneous abstract and concrete reinforcement learning. In: Proc. of the 9th Symposium on Abstraction, Reformulation and Approximation (2011)

The Evolution of Signal Communication for the e-puck Robot

Fernando Montes-Gonzalez and Fernando Aldana-Franco

Department of Artificial Intelligence, Universidad Veracruzana,
Sebastian Camacho 5, Col. Centro, Xalapa, Veracruz, Mexico
fmontes@uv.mx, faldana72@hotmail.com
<http://www.uv.mx/fmontes>

Abstract. In this paper we report our experiments with the e-puck robots for developing a communication system using evolutionary robotics. In order to do the latter we follow the evolutionary approach by using Neural Networks and Genetic Algorithms. The robots develop a communication scheme for solving tasks like: locating food areas, avoiding obstacles, approaching light sources and locating sound-sources (other robots emitting sounds). Evorobot* and Webots simulators are used as tools for computing the evolutionary process and optimization of the weights of neural controllers. As a consequence, two different kinds of neural controllers emerge. On one hand, one controller is used for robot movement; on the other hand the second controller process sound signals.

Keywords: Evolutionary Robotics, Genetic Algorithms, Mobile Robots.

1 Introduction

In robotics the implementation of obstacle avoidance is an initial task for beginning to experiment with a mobile robot. In order to solve this problem, it is necessary to implement an agent that relates percepts to actions. One of many approaches consists in directly connecting sensors to motors. Then, infrared or sonar sensors control robot motors for avoiding obstacles. Initially, a single sensor mounted on the front of a robot pointing forward is sufficient to measuring distance to an unknown object. If a sensor detects an object within certain distance, then the robot turns left or turns right to avoid it. In our laboratory we have small robots such as the e-puck robot [1], and we would like to test this robot in a realistic environment. The e-puck is an educational and research robot, its size is about 7cm diameter; the battery provides movement autonomy for about 3 hours, two stepper motors with a 20 steps per revolution and a 50:1 reduction gear. The robot has been fitted with a ring of eight infrared sensors measuring ambient light and proximity of obstacles in a range of 4 cm. A Microchip microprocessor dsPIC 30F6014A at 60MHz (about 15 MIPS) controls the robot. Additionally, the robot is equipped with a VGA camera with a field of view of 36° pointing in the direction of forward motion, a wireless Bluetooth

interface, and a ground sensor. The robot can be simulated under free and commercial applications. One of the main research areas for the e-puck is behavioral and evolutionary robotics. As for the controller of the robot we employ a Neural Network (NN) optimized with Genetic Algorithms (GAs). An overall discussion on the development of Evolutionary Robotics (ER) and GAs is provided in section 2. In section 3 we discuss some topics related to robot simulation software. Section 4 describes the parameters for the neural network, genetic algorithm, and the e-puck robot. The description of the results of experimenting with the robot is presented in Section 5. Finally, we provide a general conclusion in section 6.

2 Evolutionary Robotics and Genetic Algorithms

In writing several examples of solutions have been provided for the development of robot behavior. Commonly, the implementation of a particular behavior is carried out once the experimental setup is established. For example, robots can be set in a semi-structured environment where they solve particular tasks. Take for instance the work of Trefzer, et al. [2], where an e-puck has been provided with a general purpose obstacle avoidance controller both in simulation and on a real robot. In the mentioned work and similar works, the use of Genetic Algorithms [3] is preferred over existent evolutionary methods like: Evolutionary Strategies, Genetic and Evolutionary Programming and Co-evolution. In general the Evolutionary Robotics approach, for optimization, relies on the use of Genetic Algorithms. As for the elements to evolve, ER uses a robot controller that after several simulator interactions will produce the fittest individuals. Therefore a population of these controllers forms the elementary building blocks [4] that will serve as the necessary leverage to increment the overall fitness population. Parameter interactions in the implementation of a robot controller epistemically increment the chances to find near-optimal solutions. Most of the times a single objective evaluation function is employed in ER due to the fact that the resultant robot behavior comes from a dynamic system made with the robot and its environment [5]. However, in evolutionary computation is possible to simultaneously optimize several objectives without aggregating them as a single monolithic fitness function [6].

Recent research relies on the concept of domination and generates the so-called Pareto front to bootstrapping a light-seeking robot behavior [7]. In our example we use a single objective function to reward the performance of a global behavior to be achieved. As a consequence the use of GAs is sufficient to model complex robot behavior that provides a solid approach for starting to work on ER. In this approach is important to notice that tuning the obstacle avoidance behavior, which is shaped and nearly optimized by the use of the GA, ultimately depends on the right choice of the fitness function [8]. The use of GAs and Neural Networks offers a good solution to the problem of modeling behavior in maze-like environments [9]. In addition, neural networks have many applications in robotics due to their benefits as powerful classifiers. These processing units are noise and fault tolerant, which facilitates the robot to be driven in dynamical environments

[10]. However, neural networks require the setup of a chosen topology, and this can be done by the use of some rules of thumb. Once the topology is chosen the weights of the neural controller have to be configured. A common approach for neural networks relies on the use of backpropagation training, which is a form of supervised learning where the network has to learn a known response to a particular configuration of input data. The general misclassification error is calculated and decreased over time when the NN is trained. However, this kind of learning requires the design of training and validation data. On the other hand, the use of genetic algorithms is a form of gradient ascent approach that refines at each step of the optimization the quality of initial random solutions.

The optimization of neural controllers with genetic algorithms requires the representation, as a vector, of the weights of the neural controller. Then, a common practice consists of a direct encoding of the neural weights as an array that represents the genetic material to be manipulated by artificial evolution. A single neural controller represents one of the many individuals that form a population, which in turn are candidates for providing a good solution to the task that is to be solved. On the other hand, the fittest individuals of one population are used to breed the children that will be evaluated in the next generation. Therefore, the quality of a solution (“fitness”) is measured to acknowledge whether a candidate solution is or not a good solution to the behavior we are trying to model. If the fitness of all candidate solutions is plotted, we will end up with a convoluted space where all possible fitness solutions can be represented. Therefore, mountains and valleys, where landmarks in the mountains represent good quality solutions and landmarks near valleys are poor solutions, form this fitness landscape.

The search of the best solution within a fitness landscape requires the guidance of the genetic algorithm to move uphill to find improved solutions. Nevertheless, a few downhill steps may be necessary in order to climb to the highest mountain. Therefore, exploration is guided by the use of a fitness formula that defines the behavior to be shaped, and three *standard* genetic operators are employed to create new solutions from existent ones. Then, the current evaluated population spawns a new generation by the *selection* of a subset of the best individuals, the reproduction of the best individuals in pairs by the *crossover* of their genetic material, and the *mutation* of some of the material genetic of the individuals in the new population. The application of these operators to an initial random population of weights will produce refined solutions over the time; next the fitness evaluation will shape the final behavior through the breeding of the fittest individuals. However, several computer iterations are needed before this may occur.

3 Robot Software Simulator

EvoRobot* is a software simulator [11] developed by Stefano Nolfi and Onofrio Gigliotta at the Laboratory of Artificial Life and Robotics (ISTC-CNR) in Rome Italy. This robot simulator is a powerful tool for experimenting with the e-puck

robot. Additionally, it allows the simulation of individual and collective tasks. This software employs neuronal networks for robot control and genetic algorithms for the optimization of robot behavior. In order to run the experiments is necessary to configure parameters related to the definition of the simulated world where the robot is to be set. Some of these parameters define: the number, size and position of surrounding walls; number, position and size of food-zones; position and number of lights; and also the number and position of related landmarks. On the other hand, the characteristics of the simulated robot are defined with parameters for various sensors, topology of the neuronal controller, the lifetime of the robot, the fitness function and also parameters for the genetic algorithm. Once these parameters are determined, evorobot tests several robo-controllers using evolution. The weights of the neuronal network on each individual are modified according to an already defined fitness function. Therefore, evolution shapes behavior by analyzing the fitness of both the best individuals and the average of all the individuals on the actual generation of individuals. Then, it is possible to add new sensors by modifying the open source code of evorobot.

On the other hand, the use of a commercial robotics simulator such as Webots [12] allows simulating many experiments in robotics through many models of various robots like: Aibo, Boebot, e-puck, Khepera, Koala, Lego Mindstorm, Nao, and Pioneer. Webots is a fine tool for developing experiments in evolutionary robotics because the environmental richness of 3D and the variety of simulation models of sensors, actuators, environments, and robots. Also, it is possible to use Webots with other programming software, through communication via sockets and text files. Webots does not have implemented toolboxes for developing experiments in evolutionary robotics. However, it is possible to exploit the strengths of webots to setup an experimental environment such as the one we are proposing.

4 Robotic Experimental Setup

Initially, we based our work on that of Marocco and Nolfi [13] and de Greef and Nolfi [14]; thus we started developing the robot environment in evorobot* (see Fig. 1). The robot neural controllers of the robot team consist of neural networks with 16 sensory neurons that encode the activation states of the corresponding 8 infrared sensors, 4 communicative sensors, and 2 ground sensors. Also, an extra neuron receives the activation state of the communication actuator at times t-1, directly connected to the communication unit. The four communication neurons act as feature detector for the same kind of signal emitted by the four robots. The neural controllers also include four internal neurons that receive connections from the sensory neurons and from themselves and send connections to the motor and communicating neurons (Fig. 2). The communication sensors can detect signals produced by other robots up to a distance of 100 cm from four corresponding directions (i.e. frontal [$315^\circ - 45^\circ$], rear [$135^\circ - 225^\circ$], left [$225^\circ - 315^\circ$], right [$45^\circ - 135^\circ$]). The output of motor neurons used a sigmoid transfer function and internal neurons were computed as leaky integrators. The configuration of

internal leaky integrators with recurrent connections allows the evolving robots to integrate sensory-motor information through time and also to remember and possibly communicate previous sensory states [15,16].

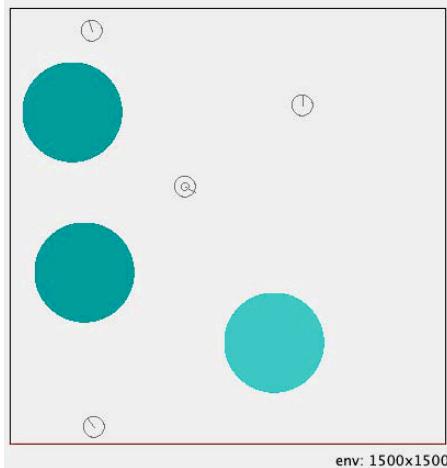


Fig. 1. The world for robot communication in evorobot*. Four e-puck robots are set in arena of 150x150 cm. Additionally, three target areas (foodzones) are included and robots are rewarded if they are able to locate and emit signals when they enter these areas.

Robots were optimized, using the evolutionary robotics approach [9], for the ability to find and remain in the four target areas (foodzones). The initial population consisted of 100 randomly generated genotypes that encoded the connection weights of the corresponding neural controllers (each parameter is encoded with 8 bits). Each genotype is translated into 4 identical neural controllers that are embodied in the four corresponding robots. The 20 best genotypes of each generation were allowed to reproduce by generating five copies each, with 2% of their bits replaced with a new randomly selected value. The fitness of the team of robots consists of the sum of 0.25 scores for each robot located in a target area and a score of -1.00 for each extra robot located in the same target area. The total fitness of a team is computed by summing the fitness gathered by the four robots in each time step. The experiment was replicated 10 times. Evolution was stopped after 1,000 iterations (Fig. 3).

Due to some limitations in evorobot, mainly caused for the lack of simulation of robot microphones and speakers; we had to develop another simulation consistent with the real e-puck sensory conditions. It is important to mention that running evolved robots in simulation and then testing them in the real environment qualitatively produce further optimized solutions. For our experiments we started a simulation in Webots with pre-defined sensory conditions in order to face the robot to similar sensory-conditions without having to use any extra

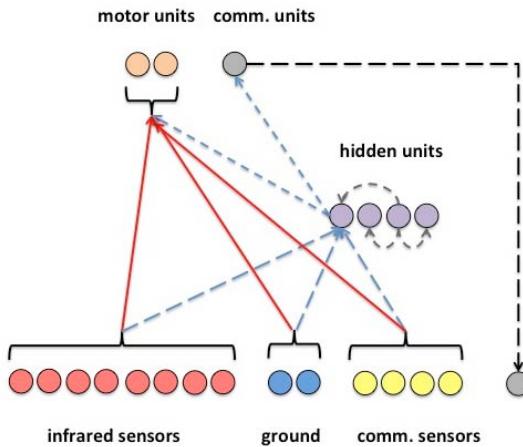


Fig. 2. The Neural Network for the communication of four robots in evorobot*. The lower, middle and top layers indicate sensory, internal, and motor neurons respectively. A straight line represents a direct connection to the motor neurons (not all the connections are shown). Notice the recurrent connections in the hidden units.

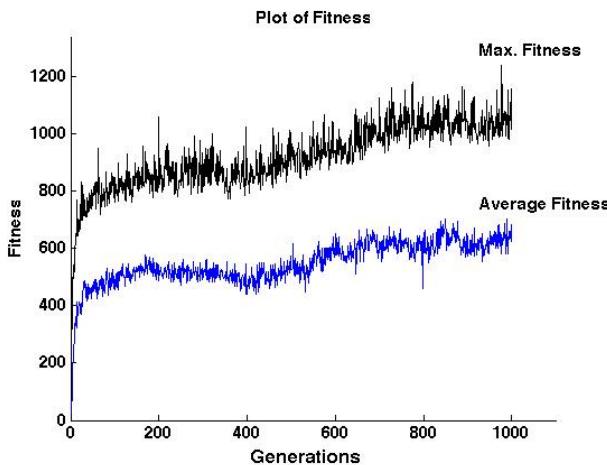


Fig. 3. Robot fitness in evorobot*. Values are plotted across 1,000 generations. For each generation the highest fitness of one individual was obtained from the averaged fitness of ten replications under similar conditions. The maximum fitness of all individuals was averaged as a measure of the population fitness.

robot hardware. Next, we set a general scenario as follows: four e-puck robots interact in a squared area that is delimited by four walls; there are three circular target areas and two groups of light-leds (see Fig. 4). Robot controllers are evolved in simulation using staged evolution, as an initial step three different kinds of controllers are created. Hence, one neural controller is set for looking and finding target areas, we labeled this controller as a general purpose controller (*gen-*pc**). A general and specific purpose 1 (*gen-spec-p1*) controller is set for those individuals that travels to groups of lights. Additionally, a different general and specific purpose 2 (*gen-spec-p2*) controller is set for robots remaining close to one of the other robots that acts as a sound-source because of the emitting sound signals. The three different neural controllers (*gen-*pc**, *gen-spec-p1*, and *gen-spec-p2*) are composed of two modules: one for motion control and the other for the production of sound signals. During the initial stage in evolution controllers are left with free neural parameters for motion and weights for sound-signals are locked with random values. At the second stage in evolution motion parameters are locked and sound-signal neuron parameters are optimized. As a final step, evolved individuals are combined to verify that signals are employed as a communication channel.

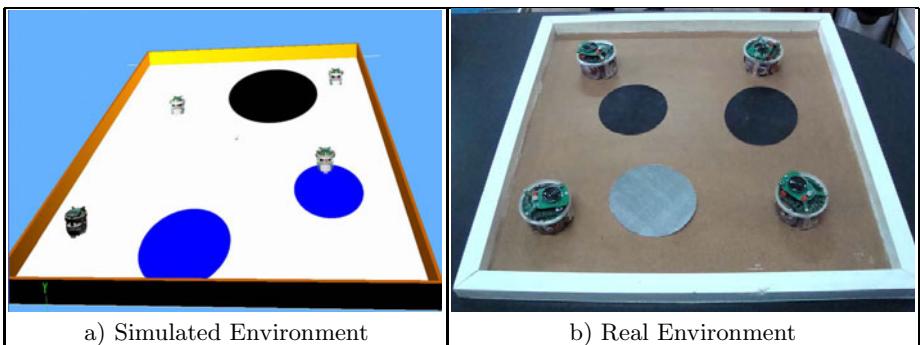


Fig. 4. a) The virtual scenario at Webots simulator composed by 4 e-puck robots, 3 target areas, 2 sets of lights, and an arena delimited by four walls. b) The real scenario composed by 4 e-puck robots, 3 target areas, and 2 sets of lights.

The neural controllers is defined as a fully connected, feedforward network with three layers (see Fig. 5). The Input layer is composed by twenty two neurons (8 infrared sensors, 8 light sensors, 3 ground sensors, and 3 microphones). The hidden layer is composed by eight neurons which represent a sensory-motor integration module. Finally, the output layer is composed by two neurons for motion control and five neurons for the signal emitting process.

The optimization process is computed by a standard genetic algorithm with a population of one hundred individuals, and the algorithm is left to run until an optimal level is reached first for the motion neurons and then for the sound-signal neurons. Then, the genetic operators mutation, selection, crossover and elitism

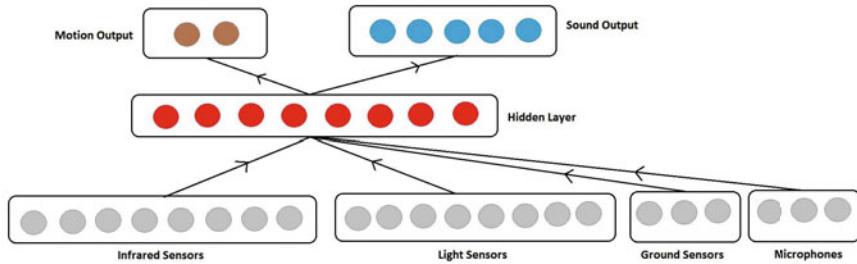


Fig. 5. Neural network used as a controller for four e-puck robots, composed by the input layer with infrared sensors, light sensors, ground sensors, and microphones. The hidden layer has eight neurons that work as a sensory-motor device. The output layer consists of 2 neurons for controlling robot-motion and 5 neurons for controlling the generation of sound signals (notice that not all connections are shown).

are used to create new individuals. Elitism replicates the current best individual in order to preserve the quality of recently produced solutions. Next tournament selection produces ninety-nine individuals by breeding winning parents in pairs. One random crossover point is applied to breed new individuals for the next generation. Mutation is applied to one percent of the total bits number of the new generation. The use of two fitness functions regulates the optimization process.

An initial fitness function looks for individuals able to locate target areas and avoid collisions. Additionally, a specific goal can be defined for those individuals able to come closer to both a group of lights or other robots emitting sound-signals.

$$f_1 = \sum_{i=0}^n target + (1 - max_ir_i) + sp \quad (1)$$

where the *target* value is 0 when a robot is not in the foodzone and 1 when a robot is in a target area, *max_ir* is the highest value of the infrared sensors, and *sp* is a specific purpose goal that takes the value of 1 in case of fulfilling a specific condition (either approaching a light or sound source) and 0 in the other case. The total time of each replication is *n*.

At the second stage a fitness function evolves individuals able to emitting particular sound signals under pre-defined conditions, for example, when a robot is in a target area or when a robot is near an obstacle.

$$f = \sum_{i=0}^n (voc + tx) \quad (2)$$

where *voc* is 0 when the robot emits a wrong signal and 1 in case of a correct signal, *tx* is 1 when the robot is emitting a signal and 0 in the other case, and *n* is the time of each trial.

In summary, three different neuro-controllers were evolved in order to test signal-presence under different situations. One gen-pc neural controller is used for robots that have to look and find target areas while avoiding obstacles. The second controller, gen-spec-p1, is used for those robots that look and find target areas, avoid obstacles, and travel near groups of lights. The last controller, gen-spec-p2, is used for robots that have to look and find target areas, avoid obstacles, and locate sound sources (another robot emitting sounds). Two fitness functions were used to evolve these controllers.

5 Results

The evolution of two kinds of neuro-controllers is carried out (Eqs. 1 and 2). We employ staged evolution to evolve three neural controllers with initial free parameters for motion and then free parameters for sound-signal neurons. Evolution is stopped until we observe the appearance of at least three sound-signals. Each individual is evaluated across ten replications with random positions and orientations. Robot controllers are tested in simulation and the genetic algorithm uses the generated data, in webots, to produce new individuals at every step of the evolution. Once individuals start emitting sound-signals, they are tested in the real environment. In order to verify the quality of solutions, experiments are replicated at least ten times. After evolution ends we observe that robots present the capability of producing specific signals under pre-defined situations such as finding target areas and/or groups of lights. Furthermore, pre-defined situations serve as an initial step in the development of more elaborated signals under different situations. Hence, robots are able to emit signals under pre-defined and emergent conditions (see Fig. 6).

Robots with a gen-p1 neuro-controller (robots that have to avoid obstacles, look and find target areas) emit three different signals under pre-defined conditions. For example, signal number 1 is emitted when a robot is in a target area without surrounding obstacles; signal number 2 is emitted when a robot is not in a target area and no nearby obstacles; signal number 3 is emitted when a robot is in a target and there is a nearby obstacle. Furthermore, signal 1 is employed as an emergent behavior when a robot is not in a target area but senses a nearby obstacle.

On the other hand, the e-puck robot team with gen-spec-p1 controllers emit five different sound-signals under pre-defined situations, which are included as a part of fitness function (Eq. 1) during the evolutionary process. Robot signals are numbered from 1 to 5 and can be defined as follows: signal number 1 is emitted when an individual is in a target area without nearby obstacles; signal number 2 is emitted if a robot is not in a target area and no nearby obstacles are sensed; signal number 3 is emitted when a robot is in a target area and a nearby obstacle is detected; signal number 4 is produced if a robot is near a group of lights and no nearby obstacle; a different signal, number 5, is produced when one robot is in a target area and a group of lights is detected. Also, an emergent signal 1 is present if a robot is not in the target area and in the presence of a nearby obstacle.

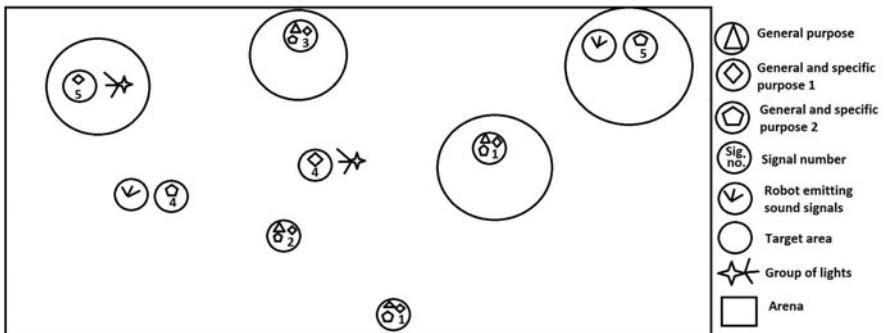


Fig. 6. Signals produced by robots under pre-defined and not pre-defined situations (emergent conditions). In the real environment emitted signals are those stored in the programming e-puck library. A general purpose controller (*gen-p1*) is used for robots that have to look and find target-areas avoiding collisions. A particular purpose controller (*gen-spec-p1*) is employed for robots that look and find target areas, avoid collisions, and come near groups of lights. Additionally, another particular purpose controller (*gen-spec-p2*) was developed for those individuals that locate target areas, avoid collisions, and go near sound-sources (another robot emitting sound).

As for the robots with a *gen-spec-p2* controller; they also emit five different pre-defined sound-signals. These signals can be identified as signal number 1 which is emitted when a robot is in a target area without any nearby obstacles; signal number 2 is shown when a robot is not in a target area and no obstacle is sensed; the presence of a robot in a target area and obstacle detection is highlighted using signal number 3; then signal 4 occurs if a robot is in the range of a sound-source; in case a robot is in a target area and a sound-source is detected then signal number 5 is emitted. Additionally, an emergent signal number 1 appears when a robot is not in a target area and senses an obstacle.

6 Discussion

The evolution of controllers in evorobot produced individuals with the ability to be concurrently located in target areas and to switch between areas. The production of robot signals in evorobot requires some extra hardware for the physical implementation [17]. In contrast, in our work, we aim to develop a realistic model that employs sound signals generated by regular sound sources (microphones and speakers). In general, three important differences were shown between the evorobot model and our own model implemented in webots. The hardware implementation of the robot team evolved in evorobot requires a duplex communication, via radio-frequency (UHF), using a board extension for the e-puck robot. In contrast, our model was designed to take advantage of duplex communication via standard sound sources (emitted by microphones and the

speaker in the e-puck robot). Furthermore, communication in evorobot has been based in radio-frequency, which is not audible as regular sound communication. The communication for radio-frequency communication models uses transceptrors that work at broad distances. Conversely, sound models, the e-puck microphones, are unable to pick signals at large distances but at audible distances. Additionally, modeling microphones and speakers requires writing and recompiling the evorobot source code. On the other hand, webots provides a limited model of sound communication for the e-puck robot. As a consequence, we chose to implement our model using webots. Another difference of our model with models developed using evorobot is that for the latter the use of one single communication unit is sufficient to modulate the output signal. In our model more than one neuron can be used to generate sound signals; taking advantage of the five pre-defined sounds stored in the programming e-puck library.

The work in this paper shows three important aspects for the production of multiple sound signals. Firstly, our model uses a controller that produces sound signals allowing robots to emit, from 3 to 5, different signals under pre-defined conditions. There is a significant relation between the number of pre-defined conditions and the number of present neurons in the output layer for the sound controller. Secondly, signal emitting controllers and their fitness function allow robots to learn pre-defined conditions through the evolutionary process. As a consequence, a group of robots learns how to emit signals according to a collective context (a basic language). Thirdly, as shown in section 5, pre-defined conditions promote the development of emergent signals, and situations, which is an important issue in evolutionary robotics.

Finally, our work aims to contribute to the development of elaborated communication in mobile robots and artificial evolution is one suitable tool for achieving this goal. Here, we have shown that new behavioral and communication signals can be developed from basic controllers, which form a communication shared context. Hence, facilitating the production and exploitation of robot signals that are not just predefined in the robot sensory-motor states but also emerge in the overall robot behavior.

7 Conclusions

In this paper we have shown a novel approach for building a basic communication system for a group of e-puck robots. The emergence of communication is the result of artificial evolution after optimizing the neural weights of the neurons in the output layer. In our model, neural controllers facilitate the robot to learn some basic rules for emitting communication signals. As a consequence, robots develop a basic language for solving a collective task.

Acknowledgments. This work has been sponsored by CONACyT–MEXICO grant SEP No. 0100895.

References

1. Mondana, F., Bonani, M.: The e-puck education robot, <http://www.e-puck.org/>
2. Trefzer, M.A., Kuyucu, T., Miller, J.F., Tyrrell, A.M.: Evolution and Analysis of a Robot Controller Based on a Gene Regulatory Network. In: Tempesti, G., Tyrrell, A.M., Miller, J.F. (eds.) ICES 2010. LNCS, vol. 6274, pp. 61–72. Springer, Heidelberg (2010)
3. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
4. Goldberg, D.E.: Genetic Algorithms. In: Search Optimization & Machine Learning. Addison Wesley Longman Inc. (2001)
5. Doncieux, S., Mouret, J.: Behavioral diversity measures for Evolutionary Robotics. In: IEEE Congress on Evolutionary Computation (2010)
6. Deb, K.: Multiobjectives optimization using evolutionary algorithms. Wiley (2001)
7. Mouret, J., Doncieux, S.: Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In: Proceedings of the Eleventh Conference on Congress on Evolutionary Computation, Trondheim, Norway, May 18–21, pp. 1161–1168. IEEE Press, Piscataway (2009)
8. Reynolds, C.W.: Evolution of Obstacle Avoidance Behaviour: Using Noise to Promote Robust Solutions. In: Kinnear Jr., K. (ed.) Advances in Genetic Programming. MIT Press, Cambridge (1994)
9. Nolfi, S., Floreano, D.: Evolutionary Robotics. The MIT Press, USA (2000)
10. Santos, J., Duro, R.: Artificial Evolution and Autonomous Robotics. Ra-Ma Editorial, Spain (2004) (in Spanish)
11. Evorobot*, Evolutionary Robotics Simulator,
<http://laral.istc.cnr.it/evorobotstar/>
12. Cyberbotics, Webots: Commercial Mobile Robot Simulation Software,
<http://www.cyberbotics.com>
13. Marocco, D., Nolfi, S.: Self-Organization of Communication in Evolving Robots. In: Proc. 10th Int. Conf. Artificial Life: Alife, pp. 199–205. MIT Press, Bloomington (2006)
14. De Greeff, J., Nolfi, S.: Evolution of communications in robots. In: Vargas, P., Di Paolo, E., Harvey, I., Husbands, P. (eds.) The Horizons for Evolutionary Robotics. The MIT Press (in press) (to appear)
15. Nolfi, S. and Marocco, D.: Evolving robots able to integrate sensory-motor information over Time. In: Theory of Biosciences, pp. 120:287–120:310 (2001)
16. Beer, R.D.: The dynamics of active categorical perception in an evolved model agent (with commentary and response). Adaptive Behavior 11(4), 209–243 (2003)
17. Cianci, C.M., Raemy, X., Pugh, J., Martinoli, A.: Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics. In: Sahin, E., Spears, W.M., Winfield, A.F.T. (eds.) SAB 2006 Ws 2007. LNCS, vol. 4433, pp. 103–115. Springer, Heidelberg (2007)

An Hybrid Expert Model to Support Tutoring Services in Robotic Arm Manipulations

Philippe Fournier-Viger¹, Roger Nkambou¹, André Mayers²,
Engelbert Mephu Nguifo³, and Usef Faghihi⁴

¹ Dept. of Computer Sciences, University of Quebec in Montreal, Canada

² Dept. of Computer Sciences, University of Sherbrooke, Canada

³ Dept. of Mathematics and Computer Sciences, Université Blaise-Pascal Clermont 2, France

⁴ Dept. of Computer Sciences, University of Memphis, U.S.A.

fournier_viger.philippe@courrier.uqam.ca,
nkambou.roger@uqam.ca, andre.mayers@usherbrooke.ca,
mephu@isima.fr, jfaghihi@yahoo.com

Abstract. To build an intelligent tutoring system, a key task is to define an expertise model that can support appropriate tutoring services. However, for some ill-defined domains, classical approaches for representing expertise do not work well. To address this issue, we illustrate in this paper a novel approach which is to combine several approaches into a hybrid model to support tutoring services in procedural and ill-defined domains. We illustrate this idea in a tutoring system for operating Canadarm2, a robotic arm installed on the international space station. To support tutoring services in this ill-defined domain, we have developed a model combining three approaches: (1) a data mining approach for automatically building a task model from user solutions, (2) a cognitive model to cover well-defined parts of the task and spatial reasoning, (3) and a 3D path-planner to cover all other aspects of the task. Experimental results show that the hybrid model allows providing assistance to learners that is much richer than what could be offered by each individual approach.

1 Introduction

To build an *intelligent tutoring system* (ITS), a key task is to define an expert model that can support appropriate tutoring services to assist learners during problem-solving activities. There are three main approaches for providing this domain expertise to an ITS. The first one is cognitive task analysis, which aims at producing effective problem spaces or task models by observing expert and novice users [1, 15] to capture different ways of solving problems. However, cognitive task analysis is a very time-consuming process [1] and it is not always possible to define a complete or partial task model by hand, in particular when a problem is ill-structured. According to Simon [2], an *ill-structured problem* is one that is complex, with indefinite starting points, multiple and arguable solutions, or unclear strategies for finding solutions. Domains that include such problems and in which tutoring targets the development of

problem-solving skills are said to be *ill-defined domains* (within the meaning of Lynch et al. [3]). The second approach, constraint-based modeling (CBM), was proposed as an alternative to cognitive task analysis [4]. It consists of specifying sets of constraints on what is a correct behavior instead of providing a complete task description. Though this approach was shown to be effective for some ill-defined domains, it can be very challenging for a domain expert to design a complete set of constraints. The third approach consists in integrating an expert system into an ITS (e.g. [5, 6]). However, developing an expert system can be difficult and costly, especially for ill-defined domains, and expert systems sometimes do not generate explanations in a form that is appropriate for learning.

Recently, a fourth approach was proposed [7, 8]. It consists of applying data mining algorithms to automatically extract partial task models from users interactions recorded in an ITS. The partial task model can then be used to offer assistance to learners. Although this approach was shown to be useful, task models that are produced are partial and thus the approach cannot support assistance in all situations.

Given that each approach has advantages and limitations, especially for ill-defined domains, we argue in this paper for the combination of several approaches to overcome their respective limitations for ill-defined domains. We illustrate this idea with CanadarmTutor, a tutoring system for operating the Canadarm2, a robotic arm installed on the international space station. The paper is organized as follows. Section 2 introduces CanadarmTutor. Section 3, 4 and 5 successively introduce the different approaches that we have tried to represent domain expertise in CanadarmTutor (an expert system, a cognitive model, and data mining algorithms for learning partial task models) and discuss their advantages and limitations. Section 6 explains how we have combined the three approaches in a hybrid model. Finally, section 7 and 8 respectively presents a comprehensive experimental evaluation of CanadarmTutor and the conclusion. Results show that the tutoring services provided by our proposed hybrid model are relevant and useful to learners.

2 The CanadarmTutor Tutoring System

CanadarmTutor [9] (cf. Figure 1.a) is a simulation-based tutoring system to teach astronauts how to operate Canadarm2, a 7 degrees of freedom robotic arm deployed on the International Space Station (ISS). The main learning activity in CanadarmTutor is to move the arm from a given configuration to a goal configuration. Operating Canadarm2 is a difficult task since the astronauts who control it have a limited view of the environment. The environment is rendered through only three monitors, each showing the view obtained from a single camera while about ten cameras are mounted at different locations on the ISS and on the arm. To move the arm, the operator must select at every moment the best cameras for viewing the scene of operation. Moreover, an operator has to select and perform appropriate joint rotations for moving the arm, while avoiding collisions and dangerous configurations. Operators also have to follow an extensive protocol that comprises numerous steps. A single mistake, such as neglecting to lock the arm into position, for example, can lead to catastrophic and costly consequences.

Operating Canadarm2 is an ill-defined task according to the definition of Simon [2]. The reason is that for each Canadarm2 manipulation, there is a huge number of possibilities for moving the robotic arm to a goal configuration and because one must also consider the safety of the manoeuvres and their easiness, it is very difficult to define a “legal move generator” for generating the moves that a human would execute. In fact, some arm movements are preferable to others depending on several criteria that are hard to formalize such as the view of the arm given by the chosen cameras, the relative position of obstacles to the arm, the arm configuration (e.g. avoiding singularities) and the familiarity of the user with certain joint manipulations over others. In practice, much of the skills to operate the arm are learned by practice. For all these reasons, it is a difficult problem to represent the domain expertise in CanadarmTutor.

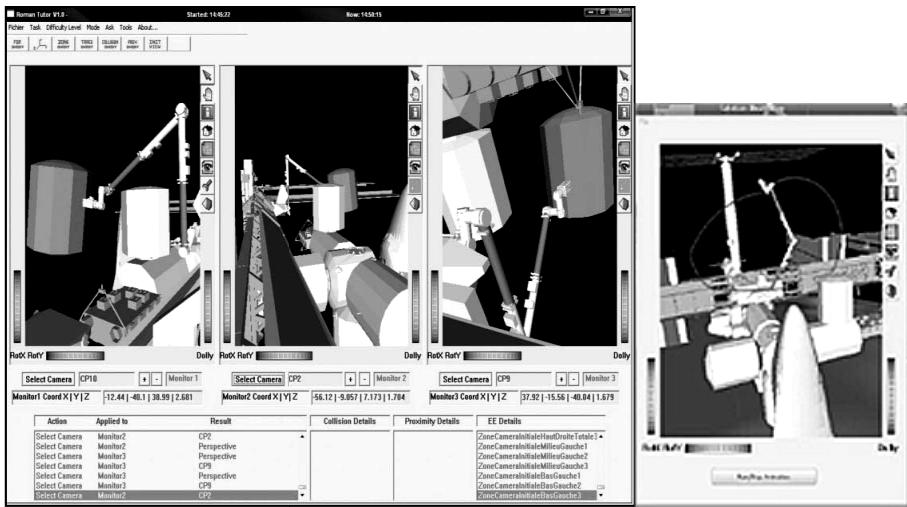


Fig. 1. (a) The CanadarmTutor user interface, (b) a path-planner demonstration

3 Integrating a Path-Planner for Automatic Path Generation

To provide the domain expertise in CanadarmTutor, we first based our work on the expert system approach. A custom path-planner named FADPRM was integrated into CanadarmTutor [9]. FADPRM is an efficient algorithm for robot path-planning in constrained environments. It can calculate a trajectory in space between any two robotic arm configurations while avoiding obstacles and considering constraints such as dangerous and desirable zones.

The integration of FADPRM in CanadarmTutor provides the following benefits. First, CanadarmTutor uses FADPRM to automatically produce demonstrations of correct and incorrect motions in training by generating a path between two arm configurations (cf. Figure 1.b), while considering the obstacles (the ISS modules) and constraints on predefined desirable and dangerous zones. Second, CanadarmTutor can generate a path and calculate an approximate distance with a learner solution to get a rough estimation of

how good the learner solution is. Although the path-planner can provide these useful tutoring services, our experiments with learners showed that the generated paths are not always realistic or easy to follow, as they are not based on human experience. Moreover, they do not cover some important aspects of the arm manipulation task such as selecting cameras and adjusting their parameters and other part of the security protocol. Also, the path-planner cannot support important tutoring services such as estimating knowledge gaps of learners because there is no representation of the knowledge or skills that a learner should possess to operate the arm.

4 Integrating a Cognitive Model for a Fine Assessment of Learner's skills and Evaluate Spatial Reasoning

Facing the aforementioned problems, we decided to apply the cognitive task analysis approach [14]. We observed several users operating Canadarm2 to understand how they operate it and also to identify common mistakes that learners do. To do this, members of our lab attended two-week training with astronauts at the Canadian Space Agency and also interviewed the training staff.

To encode the information about how users operates the robotic arm, we used a custom cognitive model [14], similar to the one used in CTAT [1], which is the reference model in the ITS community to build “model-tracing tutors”. The main difference between CTAT and our model is that given that spatial reasoning is crucial for manipulating Canadarm2, our model is provides spatial reasoning. To take into account the spatial dimension, we reviewed the research on spatial cognition. We found that most researchers in experimental psychology, neurosciences and neuropsychology agree that spatial knowledge necessary for complex spatial reasoning (“allocentric representations”) are encoded as declarative knowledge [10, 11, 12], and that it could be represented in the form of relations “*a r b*”, where “*a*” and “*b*” are symbols designating objects and “*r*” is a spatial relationship between the objects [13, 16].

Based on these facts, to model the spatial knowledge in CanadarmTutor, we discretized the 3D space into 3D sub spaces that we name elementary spaces (ES) so that the continuous space can be represented as discrete symbols. After examining different possibilities, it was determined that the most realistic types of ES for mental processing are ESs configured with an arm shape. Figure 2.a illustrates 6 of the 30 ESs that we have defined in CanadarmTutor. For example, from ES 1, it is possible to obtain ES 2, ES 4 and ES 6. Each ES is composed of three cubes. Spatial knowledge was then encoded as four types of relationships such as (1) a camera can see an ES or an ISS module, (2) an ES contains an ISS module, (3) an ES is next to another ES, and (4) a camera is attached to an ISS module. The procedural knowledge of how to move the arm to a goal position was modeled as a loop where the learner must recall a set of cameras for viewing the ESs containing the arm, select the cameras, adjust their parameters, retrieves a sequence of ESs to go from the current ES to the goal, and then move to the next ES.

Based on this task model, we integrated six new tutoring services in CanadarmTutor. Due to space limitation, we only give a brief description. First, a learner can consult the task model to learn how to operate the arm. The learner can

also consult the declarative knowledge associated to the task model to learn about properties of the ISS, the cameras and Canadarm2. Second, model-tracing [1, 15] allows evaluating the learner knowledge during arm manipulation exercises. After a few exercises CanadarmTutor build a detailed learner profile that shows the strength and weakness of the learner in terms of mastered, missing and buggy knowledge. This is done by comparing the task model with the learner solutions to see which knowledge is used by the learner. The third tutoring service is to evaluate declarative knowledge linked to the task model with direct questions such as “Which camera can be used to view the Node02 ISS module?”. The fourth tutoring service is to provide hints and demonstrations on request during arm manipulation exercises. Suggesting the next step is done by model-tracing. CanadarmTutor can give messages such as “If you have finished adjusting the third monitor, then you should start moving the robotic arm”. Demonstrations are generated dynamically also thanks to model-tracing. The fifth tutoring service is to generate personalized exercises based on the student model. During a training session, CanadarmTutor relies on the student model to generate exercises that progressively involves new knowledge and knowledge that is judged not yet mastered by the learner. For instance, CanadarmTutor can suggest an exercise involving a camera that has been rarely used by the learner. The sixth and last tutoring service is to offer proactive help to the learner. When proactive help is activated, CanadarmTutor will for example warn the learner that another camera should be selected if the arm has moved to an area that is not visible by the currently selected cameras. This type of help which is also implemented based on model-tracing is particularly appreciated by beginners and intermediate learners.

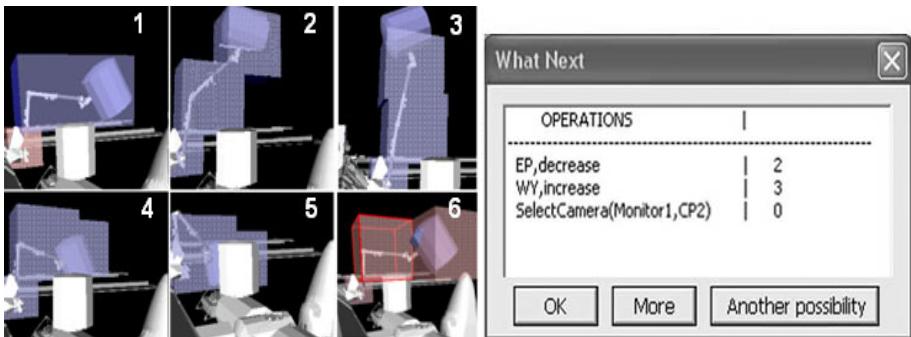


Fig. 2. (a) Six Elementary Spaces, (b) Hint Offered by the Data Mining Approach

Although the task model created by cognitive task analysis provides a fine cognitive assessment of a learner's knowledge for the main steps of the manipulation task, it does not go into finer details such as how to select joint rotations for moving Canadarm2. The reason is that for a given robotic arm manipulation problem, which is to move the arm from a given position to another on the ISS, there is a huge number of possibilities –good or bad– to solve the problem and for moving the robot one must also consider the safety of the manoeuvres and their easiness, as we explained in section 2. It is thus not possible to define a complete and explicit task model for this task (this task is ill-defined according to the definition of Simon [2]). On the other

hand, the path-planner sometimes provides paths that are too complex and difficult to be executed by users, as they are not based on human solutions.

5 Using Data Mining Techniques to Learn Partial Task Models

Facing these difficulties, we have decided to apply the fourth approach, which is the automatic acquisition of partial task models [8]. The idea is to automatically extract a partial task model from user solutions by using data mining algorithms instead of defining a task model by hand. The goal is to provide tutoring services for parts of the task of operating the arm that are ill-defined and could not be represented easily with the cognitive model (mainly, how to select the joint rotations to move Canadarm2). Furthermore, an advantage of this approach over the path-planner is that it is based on real user data.

The approach is applied in three phases in CanadarmTutor [8]. The first phase was to record many user solutions for each exercise of moving the robotic arm. As said previously, in CanadarmTutor, an exercise consists of moving the robotic arm from an initial configuration to a goal configuration. For each attempt, a *sequence of events* is created in a database. We define an event as a set of actions done by a learner that are considered simultaneous. In CanadarmTutor, we defined 112 such actions that can be recorded including (1) selecting a camera, (2) performing an increase or decrease of the pan/tilt/zoom of a camera and (3) applying a rotation value to one of the seven arm joints. An example of a partial action sequence recorded for an user in CanadarmTutor is $\langle(0, \text{rotateSP}\{2\}), (1, \text{selectCP3}), (2, \text{panCP2}\{4\}), (3, \text{zoomCP2}\{2\})\rangle$ which represents decreasing the rotation value of joint SP by two units, selecting camera CP3, increasing the pan of camera CP2 by four units and then its zoom by two units. Furthermore, we annotated each sequence with information called “dimensions”. Table 1 shows an example of a toy database containing six learners solutions annotated with five dimensions. In this Table, single letters a , b , c , and d denote actions. The first dimension “*Solution state*” indicates if the learner solution is a successful or a buggy solution. Values for this dimension are produced automatically by CanadarmTutor. The four other dimensions are examples of dimensions that can be added manually. Whereas the dimension “*Expertise*” denotes the expertise level of the learner who performed a sequence, “Skill_1”, “Skill_2” and “Skill_3” indicate whether any of these three specific skills were demonstrated by the learner when solving the problem. This example illustrates a five dimensions database. However, any kind of learner information or contextual information can be encoded as dimensions. In CanadarmTutor, we used 10 skills that we selected to be the most important, and the “solution state” and “expertise level” dimensions to annotate sequences.

The second phase is to extract partial task models from user solutions. For this, we have developed a custom sequential pattern mining algorithm¹ [8] which takes as input a sequential database having this format and a threshold named *minsup*. The algorithm then output the set of all subsequences that appears in at least *minsup* user

¹ The custom sequential pattern mining algorithm can be downloaded as part of the Sequential Pattern Mining Framework (SPMF): <http://www.philippe-fournier-viger.com/spmf/>

solutions. The idea is to extract sequences of actions that are common to many learners. We have designed the custom algorithm in previous works specifically to accept the dimensions and also different types of constraints useful in our context. For example, it is possible to specify time constraints on subsequences to be discovered to avoid discovering too short or too long patterns. Table 2 shows some subsequences (also called patterns) found from the database shown in Table 1 with $\text{minsup} = 33\%$. Consider pattern P3. This pattern represents doing action *b* one time unit (immediately) after action *a*. The pattern P3 appears in sequences S1 and S3 of Table 1. It has thus a *support* of 33 % or two sequences. Moreover, the annotations for P3 tell us that this pattern was performed by expert users who possess skills “Skill_1”, “Skill_2” and “Skill_3” and that P3 was found in plan(s) that failed, as well as plan(s) that succeeded.

The third phase is to use the partial task models for supporting tutoring services. We have implemented three tutoring services in CanadarmTutor based on the partial task models. First, CanadarmTutor can assess the profile of the learner (expertise level, skills, etc.) by looking at the patterns applied. If for example a learner applies patterns with value “intermediate” 80 % of the time for dimension “expertise”, then CanadarmTutor can assert that the learner expertise level is “intermediate”. In the same way, CanadarmTutor can diagnose mastered and missing/buggy skills for users who demonstrated a pattern by looking at the “skills” dimensions of patterns applied (e.g. “Skill_1” in Table 2).

Second, CanadarmTutor can guide the learner. This tutoring service consists in determining the possible actions from the set of patterns and proposing one or more

Table 1. An example toy database containing 6 user solutions

ID	Dimensions					Sequence of actions
	Solution state	Expertise	Skill_1	Skill_2	Skill_3	
S1	successful	Expert	yes	yes	yes	<(0,a),(1,bc)>
S2	successful	novice	no	yes	no	<(0,d)>
S3	buggy	expert	yes	yes	yes	<(0,a),(1,bc)>
S4	buggy	intermediate	no	yes	yes	<(0,a),(1,c), (2,d)>
S5	successful	expert	no	no	yes	<(0,d), (1,c)>
S6	successful	novice	no	no	yes	<(0,c), (1,d)

Table 2. Some frequent patterns extracted from the dataset of Table 1 with a *minsup* of 33 %

ID	Dimensions					Sequence of actions	Support
	Solution State	Expertise	Skill_1	Skill_2	Skill_3		
P1	*	expert	yes	yes	yes	<(0,a)>	33 %
P2	*	*	*	yes	yes	<(0,a)>	50 %
P3	*	expert	yes	yes	yes	<(0,a), (1,b)>	33 %
P4	successful	*	no	*	*	<(0,d)>	50 %
P5	successful	expert	*	*	yes	<(0,c)>	33 %
P6	successful	novice	no	*	no	<(0,d)>	33 %

actions to the learner. In CanadarmTutor, this functionality is triggered when the student selects "What should I do next?" in the interface menu. CanadarmTutor then identifies the set of possible next actions according to the matching patterns found and make a recommendation to the learner (cf. Figure 2.b). For example, if the learner performed a rotation of the joint SP followed by a rotation of the joint EP and ask to CanadarmTutor "What Should I do next?", CanadarmTutor will look for pattern that match with SP, EP to suggest what is the next action that the learner should do.

Finally, a tutoring service that has been implemented in CanadarmTutor is to let learners explore patterns to learn about possible ways to solve problems. CanadarmTutor provides an interface that lists the patterns and their annotations, and provides sorting and filtering functions (for example to display only patterns leading to success or beginner patterns).

6 Combining All Approaches in a Hybrid Model

Learning partial task models from user solutions in context of manipulating Canadarm2 allows assisting learners how to choose joint rotations –which was impossible to achieve with the cognitive model, and allows providing advices based on real user data unlike the path-planner. However, an important problem with the partial task model is that if part of a solution path was previously unexplored by other users no help can be offered to learners. Thus each of three approaches that we have applied separately in CanadarmTutor has different advantages and disadvantages. Based on this observation, we decided to combine them in a hybrid model. The goal is to combine their advantages to overcome their respective limitations. This hybrid model works as follows.

When a learner performs an arm manipulation exercise with CanadarmTutor, it uses the cognitive model to update the student model by model-tracing. The student model is a list of knowledge units from the cognitive model, each annotated with a probability that indicates if the knowledge is mastered by the learner. Moreover, the student model is also updated when a learner answers questions asked by CanadarmTutor (cf. section 4).

When a user completes a robotic arm manipulation exercise (fail or succeed), the solution is added to a sequence database of user solutions for that exercise (a database similar to the one in Table 1). The solution is annotated with the dimension "Solution State" to indicate the success or failure. Moreover, the skills from the cognitive model are used to annotate sequences as dimensions (if the mastery level is higher than 0.8 in the student model, the skill is considered mastered). Thereafter, when a minimum of 10 sequences have been recorded for an exercise, the data mining algorithm is applied for extracting a partial task model for the exercise.

During an exercise, when CanadarmTutor detects that a learner follows a pattern from the corresponding partial task model, dimensions of the pattern are used for updating the student model. For example, if a learner applies a pattern common to learners possessing "Skill_1", the mastery level of "Skill_1" in the student model will be heightened by a small increment (we use 0.05 in CanadarmTutor). In this way, the partial task models are also used for updating the student model (the student model is shared by the cognitive model and the partial task model approach).

During a learning session, CanadarmTutor uses the student model for generating exercises that progressively involves new knowledge or knowledge that is judged not yet mastered by the learner (this is done as explained in section 4). The exercises that are generated are either questions about declarative knowledge of the cognitive model or robotic arm manipulation exercises.

When a learner asks for help about what should be done next during a robotic arm manipulation exercise, the system generates a solution using the three aforementioned approaches (cf. Figure 3). First, the cognitive model gives the general procedure that should be followed for moving the arm such as “You should select a camera and then adjusts its parameter for monitor 2” (cf. Figure 3.A). This help is generated by model-tracing with the cognitive model. Then, in the same window, the patterns from the partial task model that match the current user solution are displayed to the learner. For example, three patterns are presented on Figure 3.B. The learner can view a pattern as an animation by using the arrow buttons. Patterns give mainly the information about the joint rotations that should be performed for moving the arm. If no pattern matches the current learner solution, a demonstration is generated by the path-planner that demonstrates possible paths as solutions (cf. Figure 3.C).

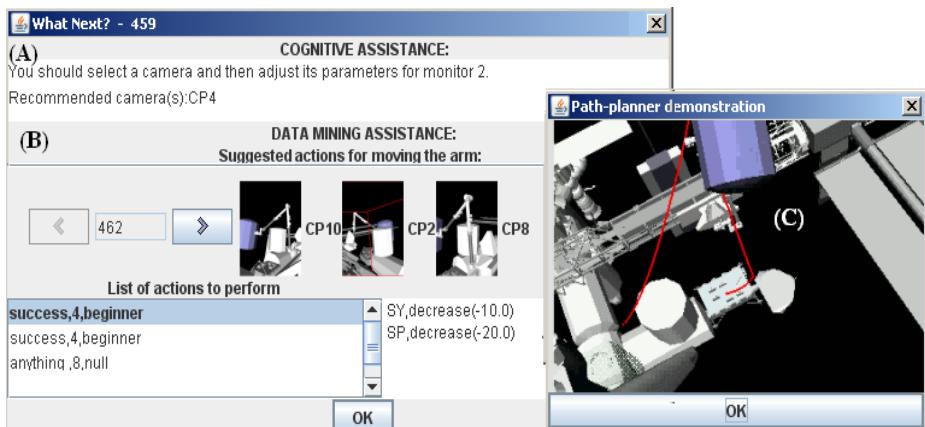


Fig. 3. An Example of Hint Offered by the Hybrid Approach

The learner can also explore patterns from the partial task models as explained in section 5 to learn about different ways to solve problems. The learner can also explore the cognitive model for learning about the general procedure for moving the arm, and the learner can request demonstrations at any time from the path-planner (cf. section 3) or the cognitive model (cf. section 4).

Lastly, as explained, CanadarmTutor can use the cognitive model to generate proactive help to learner such as assisting the learners to choose the best cameras (cf. section 4).

Table 3 provides a summary of the tutoring services offered by each approach and the hybrid model. It clearly shows the rich tutoring services that the hybrid model offers compared to each individual approach.

Table 3. Comparison of tutoring services offered by each approach

	Path-planner	Cognitive model	Data mining approach	Hybrid model
Generate path demonstrations and evaluate the path followed by the learner	X			X
Free exploration of the knowledge, generate demonstrations, hints, proactive help and evaluate skills (for the well-defined part of the task)		X		X
Evaluate declarative knowledge with questions (including spatial knowledge)		X		X
Free exploration of the knowledge, provide hints and evaluate skills (for the ill-defined part of the task)			X	X
Integrated help covering all aspects of the task				X

7 Evaluation of CanadarmTutor

To evaluate the hybrid model, we conducted an evaluation of the latest version of CanadarmTutor with 10 users. The goal of the evaluation was to measure if the tutoring services (1) help the learners to learn and (2) if they generally offer relevant information to the learners according to the context they were used.

Before performing the experiment, we recorded 30 solutions for each robotic arm manipulation exercise to make sure that some patterns are extracted for each exercise by the data mining approach to avoid the cold-start problem that no patterns are available to offer tutoring services initially.

The experiment was performed as follows. We first gave to each participant a document explaining the objectives of the experiments and what kind of data will be collected. Then, we asked each participant to perform 15 procedural exercises. Completing the exercises took about 1 hour for each participant. During this session, we allowed participants to use all aforementioned tutoring services. We set CanadarmTutor to record all solutions so that they can be examined after the experiment. During the experiment, we observed the participant and took notes to evaluate (1) if the tutoring services gave relevant help when they were used and (2) if the learner corrected their mistakes after using the tutoring services or if they were more confused. Lastly, we performed a five minutes interview with each learner to see their opinion on the same two aspects, and also their general opinion about the tutoring services and how CanadarmTutor could be improved.

The results of the experiment are as follows. All participants completed the 15 exercises. Most participants used all tutoring services. We also found that participants relied more on the tutoring services for the most difficult exercises, which is what we expected. All participants mentioned that they found the tutoring services very useful and that the tutoring services helped them learn how to manipulate Canadarm2, which is also what we have observed (we saw many times that learners did not repeat their

mistakes after receiving tutoring feedback). Users also agreed that the set of tutoring services would be less interesting if some were removed, which confirm that the hybrid model is superior to using each individual approach.

Besides, we received several comments for improvements and bug fixes. For example, two participants said they would like that CanadarmTutor has more elaborated pedagogical strategies and also the capability of generating complex tutorial dialogues. Since this work focused on the domain expertise representation, we kept these comments for future works.

8 Conclusion

In this paper, we have argued for the use of hybrid approaches to represent domain expertise in procedural and ill-defined domains. The motivation is that different approaches are sometimes better suited for different parts of the same ill-defined task. We have presented this idea with the case study of CanadarmTutor. We have first described how we have experimented three different approaches to represent domain expertise in CanadarmTutor, and discussed their respective benefits and limitations. We have then explained how we have created a hybrid model that combines the three approaches in the latest version of CanadarmTutor to solve limitations of each approach. The result is tutoring services that greatly exceed what all previous versions of CanadarmTutor offers. An evaluation was performed and shows that the hybrid model allows providing relevant and helpful tutoring services that are appreciated by users.

For future work, we will enhance CanadarmTutor with more elaborated pedagogical strategies and plan to provide additional tutoring services. We will also fix bugs and minor issues that were raised by users.

Acknowledgments. Our thanks go to the FQRNT and NSERC for their logistic and financial support. The authors would also like to thank all members of the GDAC/PLANIART teams who participated in the development of CanadarmTutor.

References

- [1] Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.: The Cognitive Tutor Authoring Tools (CTAT): Preliminary evaluation of efficiency gains. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 61–70. Springer, Heidelberg (2006)
- [2] Simon, H.A.: Information-processing theory of human problem solving. In: Estes, W.K. (ed.) Handbook of Learning and Cognitive Processes. Human information, vol. 5, pp. 271–295. John Wiley & Sons, Inc. (1978)
- [3] Lynch, C., Ashley, K., Aleven, V., Pinkwart, N.: Defining Ill-Defined Domains; A literature survey. In: Proc. Intelligent Tutoring Systems for Ill-Defined Domains Workshop at ITS 2006, pp. 1–10 (2006)
- [4] Mitrović, A., Mayo, M., Suraweera, P., Martin, B.: Constraint-Based Tutors: A Success Story. In: Monostori, L., Váncza, J., Ali, M. (eds.) IEA/AIE 2001. LNCS (LNAI), vol. 2070, pp. 931–940. Springer, Heidelberg (2001)

- [5] Clancey, W.: Use of MYCIN's rules for tutoring. In: Buchanan, B., Shortliffe, E.H. (eds.) Rule-Based Expert Systems. Addison-Wesley (1984)
- [6] Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N.: Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor. Interactive Learning Environments 8, 149–169 (2000)
- [7] Barnes, T., Stamper, J.: Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In: Woolf, B.P., Aimeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 373–382. Springer, Heidelberg (2008)
- [8] Fournier-Viger, P., Nkambou, R., MephunGuifo, E.: Learning Procedural Knowledge from User Solutions To Ill-Defined Tasks in a Simulated Robotic Manipulator. In: Romero, et al. (eds.) Handbook of Educational Data Mining, pp. 451–465. CRC Press (2010)
- [9] Belghith, K., Kabanza, F., Hartman, L.: Anytime Dynamic Path-planning with Flexible Probabilistic Roadmaps. In: Proc. 12th Intern. Conf. on Robotics and Automation (ICRA 2006), pp. 2372–2377 (2006)
- [10] Burgess, N.: Spatial memory: how egocentric and allocentric combine. Trends in Cognitive Sciences 10(12), 551–557 (2006)
- [11] Nadel, L., Hardt, O.: The Spatial Brain. Neuropsychology 18(3), 473–476 (2004)
- [12] Tversky, B.: Cognitive Maps, Cognitive Collages, and Spatial Mental Models. In: Campari, I., Frank, A.U. (eds.) COSIT 1993. LNCS, vol. 716, pp. 14–24. Springer, Heidelberg (1993)
- [13] Gunzelmann, G., Lyon, D.R.: Mechanisms for Human Spatial Competence. In: Barkowsky, T., Knauff, M., Ligozat, G., Montello, D.R. (eds.) Spatial Cognition 2007. LNCS (LNAI), vol. 4387, pp. 288–307. Springer, Heidelberg (2007)
- [14] Fournier-Viger, P., Nkambou, R., Mayers, A.: Evaluating Spatial Representations and Skills in a Simulator-Based Tutoring System. IEEE Transactions on Learning Technologies 1(1), 63–74 (2008)
- [15] Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A.: Intelligent Tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education 8, 30–43 (1997)
- [16] Carruth, D., et al.: Symbolic Model of Perception in Dynamic 3D Environments. In: Proc. 25th Army Science Conf. (2006)
- [17] Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: Proc. ICDE, pp. 3–14 (1995)

Inverse Kinematics Solution for Robotic Manipulators Using a CUDA-Based Parallel Genetic Algorithm

Omar Alejandro Aguilar and Joel Carlos Huegel

Tecnológico de Monterrey - Campus Guadalajara
Biomechatronics Lab

Av. General Ramón Corona No. 2514. Zapopan, Jal. México
{o.aguilar,jhuegel}@itesm.mx

Abstract. Inverse kinematics is one of the most basic problems that needs to be solved when using robot manipulators in a work environment. A closed-form solution is heavily dependent on the geometry of the manipulator. A solution may not be possible for certain robots. On the other hand, there may be an infinite number of solutions, as is the case of highly redundant manipulators. We propose a Genetic Algorithm (GA) to approximate a solution to the inverse kinematics problem for both the position and orientation. This algorithm can be applied to different kinds of manipulators. Since typical GAs may take a considerable time to find a solution, a parallel implementation of the same algorithm (PGA) was developed for its execution on a CUDA-based architecture. A computational model of a PUMA 500 robot was used as a test subject for the GA. Results show that the parallel implementation of the algorithm was able to reduce the execution time of the serial GA significantly while also obtaining the solution within the specified margin of error.

1 Introduction

The use of industrial robots on the work environment has become more important than ever. Hazardous and uncomfortable environments, repeatability and consistency, tedious work, and high productivity requirements are some of the situations in which the use of robots instead of a humans is more suitable [7]. The robot manipulator is formed by a series of joints and links that allows the manipulator to reach a point within a certain workspace, by allowing the relative motion between two parts of the body. Each joint provides the robot with a degree of freedom (DOF). Robot manipulators must have at least 6 DOFs in order to reach any position and orientation on a three-dimensional space. More than 6 DOFs provide redundancy so that there can be multiple solutions and thus avoid obstacles, including the robot itself, in the workspace.

Industrial robots may vary in size, reach, degrees of freedom, type of joints among other characteristics. Yet, for reaching or determining the position of a robot's end effector in space (without considering the forces involved) two problems must be solved for any kind of manipulator: the direct kinematics and

inverse kinematics. The former is concerned with, given the position of the joints, determining the position of the end effector. In contrast, inverse kinematics determines the required position of every joint to obtain a given a position and rotation of the end effector.

Obtaining the direct kinematics for any manipulator is a very straightforward procedure, whether it is done by geometric or analytic methods. On the other hand, inverse kinematics is a much more difficult problem since, for a given position and orientation, there could be no solution, multiple solutions or even an infinite number of solutions in manipulators with redundant DOFs. Inverse kinematics is heavily dependent on each manipulator's geometry; therefore we propose the implementation of a genetic algorithm (GA) that searches for a solution to the inverse kinematics problem, within a margin of error. Since traditional serial GAs may take a considerable amount of time to find a solution, a parallelized version of the same GA was implemented on a CUDA-based architecture running on an NVIDIA Tesla C1060 card. A 6-DOF PUMA 500 robot was used as case study for the implemented GA. In the following sections we will discuss briefly the nature of GAs and how they work, followed by a description of the GA designed for this specific problem and its parallelization.

1.1 Genetic Algorithms

Directed evolutionary algorithms are a form of meta-heuristic search and optimization techniques used to find an optimal or near-optimal solution to a given problem using an iterative approach. One of the most used forms of directed evolutionary algorithms is the genetic algorithm (GA) proposed by Holland [8]. By being *directed* these algorithms differ from the traditional Darwinian evolution. GAs can be applied to solve a wide variety of problems using directed random searches to locate an optimal or near-optimal solution. These applications cover fields such as robotics, scheduling, simulation and modeling, engineering design, pattern recognition, among others [9]. These algorithms have gained interest as computing power has increased making the interative process feasible withing a reasonable amount of time.

Genetic algorithms transform an initial population into a new generation by allowing a population to adjust to the environment and produce *fitter* individuals. GAs start with an initial group of individuals (population), where each one of these individuals encode a possible solution to a particular problem. Each individual is represented by a set of pre-defined characteristics (chromosomes) which correspond to the individual's *genetic* code. This initial population is generated by giving random values to each individual's characteristics. Once the initial population is set, there must be a mechanism to evaluate if an individual is an actual solution to the problem to be solved, and if not, how close or far it is from it. Most of the time a solution to the problem will not be among the initial population because of the randomness in which it was generated, yet there will be some individuals who are closer to the solution than the rest of the population. This is represented by a fitness value given by a problem-specific function designed by the author to evaluate the *correctness* of a proposed solution.

The goal of the GA will be to evolve the aforementioned initial population so that each subsequent generation has individuals with higher fitness values and eventually find an approximate solution to the problem within a predefined error established by the author. There must be a selection mechanism to guarantee that the best individuals have a higher chance to survive and generate offspring. When generating offspring, a set of genetic operations is performed on the current generation to produce newer, fitter individuals which resemble the biologic phenomena that occurs from one generation to another. These operations are termed: crossover and mutation. There are several ways in which these operations are executed and are heavily dependent on the nature of the problem and the encoding used by the author to represent the solutions but in general, the former refers to the exchange of genetic code between two individuals, while the latter refers to a small change on the characteristics of the individual.

1.2 Simple Genetic Algorithm

The first genetic algorithm described by Holland is known as the Simple Genetic Algorithm (SGA). On the SGA the individuals are represented by a string of binary bits, each bit representing a characteristic. As previously mentioned, the initial population is generated randomly, while for selection it uses a proportionate selection scheme. In this scheme, the fitness function is represented by f , and each individual has an expected number of offspring given by f_i/\bar{f} where \bar{f} is the average fitness value of the current population. The method used to follow the proportionate selection scheme is called the roulette wheel selection. In this method, each string is given a slot of a roulette wheel defined by an angle $2\pi f_i/\bar{f}$. A random number is generated between 0 and 2π and if the number falls in the section of a string's angle, that specific string will generate offspring. As it can be seen, strings with a higher f_i/\bar{f} ratio will generate more offspring, causing subsequent generations to have more individuals derived from the previous generation's fitter individuals.

To generate the next generations, the SGA selects pair of parents using the aforementioned technique, generating a new individual which inherits the genetic code form its parents. At this stage, crossover and mutation operations are applied to modify the genetic code of the children. The SGA uses a single-point crossover, where a random point is chosen between 1 and $l - 1$, where l is the length of the string. All the bits following the randomly selected bit are exchanged between the two strings. Furthermore, mutation is performed by flipping a single bit. Both of these operations, crossover and mutation, are only performed each time with a probability of p_c and p_m respectively. A flow diagram to describe the steps followed by the SGA is shown in Fig. 1.

The main disadvantage in implementing GAs to solve real-world problems is the computational cost required to evaluate fitness functions and perform genetic operators. This is specially true with harder and bigger problems where the fitness function is computationally heavy. Parallel implementations of GAs have shown to be a good way to reduce the execution time [4]. Recently, computational power scaling has been achieved by using multi-core architectures

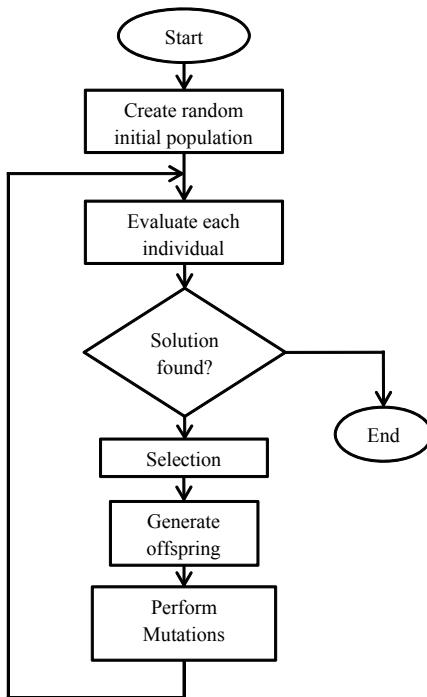


Fig. 1. Simple Genetic Algorithm Flow Diagram

rather than just increasing processor clock speeds. For this reason, and the parallel nature of some of the operations on GAs, their implementation over today's parallel/distributed computing paradigms [11].

1.3 CUDA

GPUs are considered massively parallel processors, which means they are formed by several individual processing units, each of which has their own memory and can communicate through some mechanism. Following Flynn's taxonomy [6] they are classified as SIMD (Single Instruction stream, Multiple Data stream) computers. This means that several processors perform the same instructions on different data at the same time. This characteristic makes them ideal when large amounts of data need to be processed applying the same operations with no concern on the order in which the data needs to be processed. Some aspects of the GA, like the fitness function evaluation are said to be *embarrassingly* parallel, thus making them ideal for SIMD machines. Selection and both genetic operators may need to be modified from the ones used in the SGA to take advantage of such architecture.

Table 1. DH Parameters

Parameter	Meaning
θ	Angle about z_{i-1} , from x_{i-1} to x_i
d_i	Distance along z_{i-1} from the origin of the $(i-1)-th$ coordinate system to the intersection with x_i
a_i	Distance along x_i from the intersection of z_{i-1} with x_i to the origin of the $i-th$ coordinate system
α_i	Angle about x_i , from z_{i-1} to z_i

CUDA is NVIDIA's general purpose parallel computing architecture. It provides programmers with an efficient way to develop highly parallel applications running on the GPU (Graphics Processing Unit). Before the release of CUDA in 2006 [13], GPUs were very difficult to program since programmers required to use graphic APIs, such as DirectX and OpenGL, to develop programs that ran on the GPU. Developers needed to think about the problem in such a way that it resembled a rendering process using shaders [10]. Still, directed evolutionary algorithms were developed prior to the release of CUDA. In [17], Wong et al. developed an hybrid master-slave fine grained evolutionary algorithm [4] where fitness evaluation, mutation and reproduction were performed on the GPU, whereas competition and selection were done on the CPU side. Wong and Wong implemented a genetic algorithm running almost in its entirety on the GPU, only the random number generation was left to the CPU [16].

After the introduction of the CUDA architecture, a wide range of applications have been developed, including computational chemistry, sparse matrix solvers, sorting, searching, and physics models [12]. In the field of evolutionary algorithms, several researchers have developed and implemented algorithms using a CUDA-based architecture. In the following section we will describe the proposed GA and the method used for its parallelization on a CUDA-based architecture.

2 Methodology

2.1 Direct Kinematics

Before defining the structure of the GA, we need to determine the direct kinematics equations for the manipulator which we are going to use. The Denavit and Hartenberg convention [2,5] provides an easy way of modeling a kinematic chain by carefully assigning a coordinate system to each joint. By following a set of rules, each joint can be represented by 4 parameters (see Table 1). These four parameters describe 4 basic transformations that allow to obtain a transformation matrix (see Eq. 1) for each joint.

$${}^{i-1}\mathbf{A}_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

In Eq. 1 S and C represent a sine and cosine respectively. By substituting the Denavit-Hartenberg parameters of the PUMA 500 manipulator for each joint (see Table 2), we can obtain the matrix T , which represents the homogeneous transformation matrix for the entire manipulator, by multiplying the 6 transformation matrices that correspond to the six joints of the manipulator (see Eq. 2). The vector \mathbf{p} resulting from this equation gives the final position of the end effector, while the orientation is given by vectors \mathbf{n} , \mathbf{o} and \mathbf{a} .

Table 2. PUMA 500 DH Parameters

Axis	θ	d	a	α
1	θ_1	0	431.80mm	$\pi/2$
2	θ_2	0	20.30mm	0
3	θ_3	150.05mm	0	$-\pi/2$
4	θ_4	433.07mm	0	$\pi/2$
5	θ_5	0	0	$-\pi/2$
6	θ_6	55.588mm	0	0

$$\begin{aligned} \mathbf{T} &= {}^0\mathbf{A}_1{}^1\mathbf{A}_2{}^2\mathbf{A}_3{}^3\mathbf{A}_4{}^4\mathbf{A}_5{}^5\mathbf{A}_6 \\ &= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2)$$

2.2 Genetic Algorithm

The proposed GA is similar in structure to the original SGA described by Holland. There are a few differences, however, in the implementation of each stage. The first step is to define the encoding of the problem, that is how we represent each individual and what the characteristics are going to look like. On the SGA individuals were binary strings, where each bit represented a characteristic. For this particular problem, this representation will not work. Real values can be represented as binary strings, in fact they will be represented by binary bits since the computer can only deal with such data. However, by treating the values as such, we face a big problem, which is that each bit on the string does not carry the same weight. This will eventually have repercussions when doing genetic operations. For example, let's assume we use 16 bit strings, if a mutation occurs on the second bit of the string, the final value of the string will be changed by $2^1 = 2$, whatever that value represents on the encoding; however if the mutation occurs in bit 8 of the string, its value will change $2^7 = 256$. This will lead to very disruptive genetic operations, preventing the algorithm to converge. Because of the nature of the problem we need to use continuous variables to represent the strings. Each possible solution is composed of a set of angles θ_i where $i = 1, 2, \dots, 6$. That means each individual encodes a possible solution, which is a set of values for each joint angle (characteristics).

As a result of this encoding scheme, the mutation operation differs from the SGA. When we were dealing with binary characteristics, a mutation was performed in the form of a negation of a particular bit. Since our characteristics are real variables, we cannot flip their values. By definition, mutations are small variations on the genetic code, hence the mutation operation on our GA will translate to a small random rotation within a range in the joint angles of the robot's manipulator. On the other hand, crossover will remain essentially the same, a series of joint angles will be exchanged between the individuals.

Before establishing a selection scheme, we need to define the fitness function to evaluate each individual in the population. The inverse kinematics problem on a three-dimensional space can be seen as a multi-objective optimization problem, since translation and rotation must be satisfied. Eq. 5 describes an overall fitness value f_a , where α and β are constants to represent an overall-fitness value which takes both, the space and orientation errors, into consideration. Eq. 3 and Eq. 4 refer to the magnitude of the space and orientation errors respectively. Both are obtained by getting the difference between the desired point in space and the result from evaluating the direct kinematics for the individual's joint angle values.

$$f_s = \sqrt{{\Delta_x}^2 + {\Delta_y}^2 + {\Delta_z}^2} \quad (3)$$

$$f_o = \sqrt{{\Delta_{Rx}}^2 + {\Delta_{Ry}}^2 + {\Delta_{Rz}}^2} \quad (4)$$

$$f_a = \alpha f_s + \beta f_o \quad (5)$$

The most important step, and the one that will influence the most to the success or failure of the GA is the selection scheme. Srinivas and Patnaik [15] describe various issues that are present when only the traditional roulette wheel selection scheme, used by the SGA, is implemented. An improper selection scheme may cause the GA to converge prematurely due to the lack of variation in the population. This issue could also prevent the algorithm to look on the entire search space and get stuck on some local optima instead of searching for the global optimum. Another problem that can happen using only roulette wheel selection occurs at the later stages of the GA, when most individuals have converged to certain area of the search space, each string has approximately the same chance to generate offspring, thereby diminishing the capability of the GA to obtain fitter individuals. Since the roulette wheel selection scheme used in the SGA requires the knowledge of the complete population to calculate the expected offspring, a tournament selection is preferred when parallelizing the GA [3]. It prevents that very few individuals, with a very low error, saturate the next generation's genetic code because of the proportional selection scheme [15]. In the tournament selection scheme, a set of n individuals is selected randomly from the population and a competition is held between those individuals and the fitter one survives. Each member of the population has to participate in at least one tournament, hence for each member we select $n - 1$ other individuals

to compete with him. Consequently, there will be n number of tournaments, where n is the size of the population. Normally, the best individual in the tournament is chosen deterministically, however we added an stochastic element to this selection process by occasionally allowing a weaker individual to win the tournament. This is accomplished by selecting the fitter individual with a fixed probability $0.5 < pT < 1$. The reason behind this added element of stochasticity is to prevent the algorithm from converging prematurely and getting stuck on a subspace of the entire search space because of the loss in diversity in the genetic code, causing it to move only by mutation. A problem that may arise when using the stochastic tournament selection scheme is that the fitter individuals in the generation can occasionally disappear from the population. To prevent this, the number of tournaments was reduced to $n - 3$, where the remaining spots in the population are occupied by the individuals with the highest f_a , f_s and f_o , ensuring that the fitter individuals always survive.

Preliminary results showed that, even with the stochastic tournament selection, the algorithm may occasionally converge prematurely to a point that is close to the solution but not entirely optimal, since it is not within the accepted margin of error. Therefore we implemented an adaptive mutation operator. Traditionally, mutation occurs with a low probability, usually less than 0.05. Mutation is used seldom used because it is usually a way to recover lost genetic code. However, we implemented mutation in such a way that it resembles an inverse simulated annealing algorithm, meaning that with each iteration of the algorithm solutions tend to move at a higher pace. Hence the probability that a mutation occurs increases exponentially with each iteration of the algorithm. At the same time we reduce the magnitude of the mutation in the same way to increase the randomness in the searching method but on a reduced search space while it is being directed towards the solution. Eq. 6 and Eq. 7 show the value of the probability of performing a mutation pm and the maximum change in the angle mut for the current iteration $cont$ of the algorithm with a maximum number of iterations k . Variables $minPm$ and $maxPm$ represent the minimum and maximum probabilities of mutation respectively, while $minM$ and $maxM$ represent the angle each joint can turn at the most for each iteration.

$$pm = minPm \cdot \left(\frac{maxPm}{minPm} \right)^{cont/k} \quad (6)$$

$$mut = maxM \cdot \left(\frac{minM}{maxM} \right)^{cont/k} \quad (7)$$

The pseudo-code shown in Fig. 2, illustrates the implemented GA with the proposed selection scheme and genetic operators. In this pseudo-code a random initial population p is generated, then the forward kinematics is evaluated for each individual i in the population with function FWDKINE. Then $fitness_i$ is calculated by function ERROR. The next step is to add the fittest individual f to the next generation. For each of the individuals, an opponent is selected

randomly by function `SELECTRANDOM`, the first individual is replaced by the opponent if it is fitter and a random number is bellow the current value of *pm*. The cycle repeats `MAXTRIES` times or until the fittest individual is within the allowed tolerance.

2.3 CUDA Implementation

The parallel version of the GA is exactly the same as the serial version. Initial random population, fitness evaluation, selection and genetic operators are all carried out on the device. The random numbers were generated using the CURAND library. For initializing the seed for each thread, the method proposed by Arora et al.[1] was implemented. In this method, an array containing a set of random numbers generated on the host side with C `rand()` function is copied to the device. Each value is then used as a seed value for each thread's random number generator.

A kernel wrapper is called within a C++ program which receives the input (position and orientation) and gives a solution, if any, for the inverse kinematics problem. The wrapper calls a single kernel that performs the whole algorithm on the device and returns the best solution it finds until the stop condition is met. A single grid with one block of 512 threads, where each thread manages a single individual of the population, was used to replicate the same type of algorithm as the serial version but executed in a parallel fashion on the GPU. The kernel starts by initializing the random number generator using the values stored in the global memory for that purpose. Each thread generates a single random candidate solution within the limits of the robot workspace by giving a value for each angle inside the joint limits. Then, fitness values are calculated by each thread for every member of the population. Both the candidate solution and fitness value are stored on shared memory so that they can be seen by the rest of the threads in the thread block. The fitter individuals in terms of orientation, position and overall fitness are found using parallel reduction on the entire thread block. Each of the three fitter individuals is compared to the target point and if any of them fall within the tolerance it is copied to the global memory so that it can be retrieved by the main program, thus ending the kernel execution. If no solution has been found tournament selection is performed were each thread selects a random index to choose the individual with whom the tournament will be held. After the tournament selection ends, crossover is performed in a similar manner followed by the mutation.

In the following section we will present the outputs of the algorithm for a set of inputs formed by a point in space and an rotation. Each input is represented by the set $\{x, y, z, \theta_x, \theta_y, \theta_z\}$, where x, y, z represent the Cartesian coordinates and $\theta_x, \theta_y, \theta_z$ the euler angles to specify the orientation of the end effector. We established the stop condition on the GA where there is a candidate solution with an error that is less than 1mm in position and less than 1 in orientation. Performance of the original GA and the parallel version are compared by

Algorithm 1. Genetic Algorithm

```

procedure INVKINE(position)
    p  $\leftarrow$  random initial population
    n  $\leftarrow$  empty next generation
    for cont = 0 to MAX_TRIES do
        mut = maxM · (minM/maxM)(cont/k)
        pm = minPm · (maxPm/minPm)(cont/k)
        for each i in p do
            pos  $\leftarrow$  FWDKINE(i)
            fitnessi = ERROR(pos)
        end for
        f  $\leftarrow$  fittest i in p
        if FWDKINE(f) within error then
            return r
        end if
        while n NOT full do
            p1  $\leftarrow$  POP(p)
            p2  $\leftarrow$  SELECTRANDOM(p)
            if p2.fitness > p1.fitness AND randnum  $\leq$  pTR then
                n  $\leftarrow$  PUSH(p2)
            else
                n  $\leftarrow$  PUSH(p1)
            end if
        end while
        for each i in p do
            p1  $\leftarrow$  POP(n)
            if randnum  $\leq$  pc then
                p2  $\leftarrow$  SELECTRANDOM(n)
                p1  $\leftarrow$  CROSSOVER(p1, p2)
            end if
            for each joint in i do
                if randnum  $\leq$  pm then
                    joint  $\leftarrow$  MUTATION(joint, mut)
                end if
            end for
        end for
        p = n
    end for
    return failure
end procedure

```

Fig. 2. Genetic Algorithm for determining an inverse kinematics solution to an industrial robot arm

comparing number of iterations of the algorithm and computing time using a Core 2 Duo E670 @2.66 GHz. and an Nvidia Tesla C1060 for the serial and parallel versions respectively.

3 Results

Two points were chosen arbitrarily and were given as input to both, the parallel and serial versions of the GA. The first point was set at $\{0.0293, -0.1639, 0.3430, -2.8565, 1.8428, -2.7641\}$ and the second in $\{-0.2847, -0.8033, 0.1860, -1.3748, 1.535, 0.8731\}$. The algorithms were executed seven times for each point until an answer was found within the specified error. Each run was executed with a different set of pseudo-random numbers. In terms of the programming language, a different seed was set for the random number generator for each run. The number of iterations and execution time were registered and compared.

Table 3. Algorithm results for first desired point

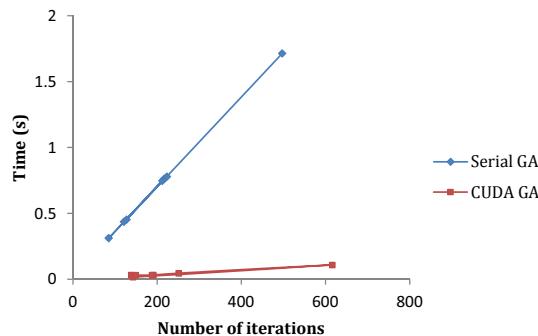
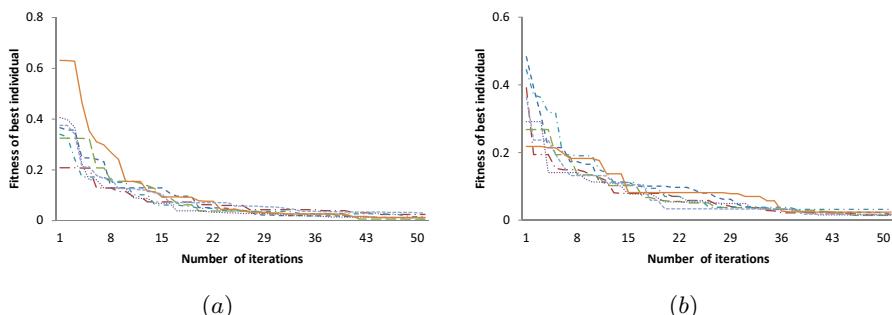
	Serial		CUDA	
	Iterations	Time(s)	Iterations	Time(s)
1	140	0.492	91	0.015
2	646	2.199	209	0.031
3	573	1.953	387	0.062
4	169	0.605	328	0.047
5	662	2.264	225	0.031
6	131	0.465	417	0.077
7	448	1.535	168	0.031
Average	395.571	1.359	260.714	0.042

As shown in tables 3 and 4, the CUDA-based algorithm shows a significant improvement in term of execution speed. On average, it performed approximately 32 times faster on the first point in space and close to 20 times faster on the second one. Due to the stochastic nature of the algorithm, the number of iterations needed to find an appropriate solution varies from one run to another. Still, even the slowest run on the parallel version was faster than every single try on the serial GA. Figure 3 shows the amount of time needed for both algorithms with respect to the number of iterations the algorithm performed. As it can be seen, there is a clear difference in execution time when using the CUDA-based algorithm that grows as the number of iterations increases.

As stated in the previous section, the best individual was always kept intact for the next generation. Figure 4 shows how the fitness value of the best individual decreases with each iteration of the algorithm. It can be seen that both, the parallel and serial versions of the GA have a very similar behavior in terms of population evolution for each generation. At the beginning, individuals evolve rapidly towards the optimal solution, yet as the population moves toward the solution, the speed in which this occurs diminishes. This is because the entire population is converging toward a single point, preventing the algorithm from exploring different parts of the search space. From this point on, mutation plays a key role in ensuring that the algorithm searches on neighboring parts of the

Table 4. Algorithm results for second desired point

	Serial		CUDA	
	Iterations	Time(s)	Iterations	Time(s)
1	121	0.435	139	0.032
2	217	0.762	143	0.016
3	85	0.311	616	0.109
4	212	0.747	252	0.046
5	223	0.779	191	0.032
6	127	0.452	188	0.031
7	497	1.714	150	0.031
Average	211.714	0.743	239.857	0.042

**Fig. 3.** Time to find a solution for both algorithms on each of the seven runs**Fig. 4.** Evolution of the best individual's fitness value in the population on each of the seven runs for the serial (a) and parallel (b) genetic algorithms

space. Our adaptive mutation scheme proved to be quite effective in getting closer to an optimal solution by preventing the population from getting stuck on a close but non-optimal solution.

4 Discussion

A Genetic Algorithm may be suitable for obtaining a solution to the inverse kinematic problem of highly redundant manipulators, where no closed-form solution can be found or there are an infinite number of possibilities. These highly redundant manipulators are usually controlled by a human who controls each joint individually. While the execution time of the typical GA is not suitable for on-line operation, the parallelization using consumer level graphic cards may accelerate the execution time enough for a practical use on these situations. Furthermore, a multi-objective GA may be developed using the presented algorithm as a base for avoiding obstacles in certain environments. The proposed GA can be used on an educational platform, where any robot can be defined by giving the Denavit-Hartenberg parameters so that a solution to the inverse kinematics can be found without the need of any extra calculations.

Previous implementations of PGAs on CUDA-based architectures have shown different results. Wong et al. [17] implemented an hybrid master-slave fine grained model obtaining speedups varying from 1.13 to 4.24 with respect to the CPU implementation. Arora et al. [1] implemented a parallelized version of a GA, testing its performance on binary and real-coded problems, their obtained speedups of 40 to 400 times compared to the serial GA. In [18], Yu et al. implemented a fine-grained GA obtaining speed gains of up to 20 depending on population size. In [14], Pospichal et al. developed an island based GA with unidirectional ring migrations. In their work they obtained speedups up to seven thousand times with respect to a linear GA executed on the CPU.

In terms of speedups going from a serial GA running on the CPU to a parallel implementation on the GPU, our results are similar to what the aforementioned researches have obtained. Unlike the work of Pospichal et al. [14], our parallel implementation uses the same structure than the serial version. Speedups are achieved by parallelizing those operations on the GPU. To improve our results, further optimizations to the CUDA C Code need to be done to fine-tune the execution on the CUDA architecture. Different models such as island-based or overlapping demes [4] can improve the performance of the GA by taking full advantage of the massively parallel nature of the GPU cards.

5 Conclusions

While the inverse kinematics can be solved on a fraction of a second using a closed solution obtained by analytical or geometric methods, it is a problem which usually needs to be solved independently for each robot. This document presented a Genetic Algorithm that can be used to solve the same problem for different robots by just giving the Denavit-Hartenberg parameters for each joint. The GA searches for a point in space close enough to the desired position by iteratively reducing the search space getting better candidate solutions with each run mimicking the natural selection processes. The proposed GA proved to be quite efficient in finding a solution within the tolerance converging on each run of the GA. The GA may be useful in the case of highly redundant manipulators

but further research needs to be done in order to determine if the implemented GA can be used in those cases. A parallel version of the same algorithm was implemented and executed on a CUDA-based architecture. This version of the GA is 20 to 30 times faster than the serial GA. Further optimizations need to be done to the CUDA C code to achieve better performance.

References

1. Arora, R., Tulshyan, R., Deb, K.: Parallelization of binary and real-coded genetic algorithms on gpu using cuda. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (July 2010)
2. Barrientos, A., Peñín, L.F., Balaguer, C., Aracil, R.: Fundamentos de Robótica, 2nd edn. McGraw-Hill, Spain (2007)
3. Cantú-Paz, E.: A summary of research on parallel genetic algorithms (1995)
4. Cantú-Paz, E.: A survey of parallel genetic algorithms. *Calculateurs Paralleles, Réseaux et Systèmes Repartis* 10(2), 141–171 (1998)
5. Craig, J.J.: Robótica, 3rd edn. Prentice Hall, Mexico (2006)
6. Flynn, M.J.: Some computer organizations and their effectiveness. *IEEE Transactions on Computers* C-21(9), 948–960 (1972)
7. Groover, M.P.: Automation, Production Systems, and Computer-Integrated Manufacturing, 3rd edn. Prentice Hall, New Jersey (2008)
8. Holland, J.H.: Adaptation in natural and artificial systems. MIT Press, Cambridge (1992)
9. Koza, J.R.: Genetic programming (1997)
10. Macedonia, M.: The gpu enters computing’s mainstream. *Computer* 36(10), 106–108 (2003)
11. Munawar, A., Wahib, M., Munetomo, M., Akama, K.: A survey: Genetic algorithms and the fast evolving world of parallel computing. In: 10th IEEE International Conference on High Performance Computing and Communications 2008, HPCC 2008, pp. 897–902 (2008)
12. Nickolls, J., Buck, I., Garland, M., Skadron, K.: Scalable parallel programming with cuda. *Queue* 6, 40–53 (2008)
13. Nvidia: Nvidia cuda c programming guide (September 2010), http://developer.download.nvidia.com/compute/cuda/3_0/toolkit/docs/NVIDIA_CUDA_ProgrammingGuide.pdf
14. Pospichal, P., Jaros, J., Schwarz, J.: Parallel Genetic Algorithm on the Cuda Architecture. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcazar, A.I., Goh, C.-K., Merelo, J.J., Neri, F., Preuß, M., Togelius, J., Yannakakis, G.N. (eds.) *EvoApplicatons 2010*. LNCS, vol. 6024, pp. 442–451. Springer, Heidelberg (2010)
15. Srinivas, M., Patnaik, L.: Genetic algorithms: a survey. *Computer* 27(6), 17–26 (1994)
16. Wong, M.L., Wong, T.T.: Parallel hybrid genetic algorithms on consumer-level graphics hardware. In: IEEE Congress on Evolutionary Computation, CEC 2006, pp. 2973–2980 (2006)
17. Wong, M.L., Wong, T.T., Fok, K.L.: Parallel evolutionary algorithms on graphics processing unit. In: The 2005 IEEE Congress on Evolutionary Computation, vol. 3, pp. 2286–2293 (September 2005)
18. Yu, Q., Chen, C., Pan, Z.: Parallel Genetic Algorithms on Programmable Graphics Hardware. In: Wang, L., Chen, K., S. Ong, Y. (eds.) *ICNC 2005*. LNCS, vol. 3612, pp. 1051–1059. Springer, Heidelberg (2005)

MFCA: Matched Filters with Cellular Automata for Retinal Vessel Detection

Oscar Dalmau¹ and Teresa Alarcon²

¹ Centro de Investigación en Matemáticas,
A.C. Jalisco S/N Col. Valenciana, 36240, Guanajuato, Gto. México

² Centro Universitario de los Valles
Carretera Guadalajara - Ameca Km. 45.5 C.P. 46600. Ameca, Jalisco, México
dalmau@cimat.mx, teresa.alarcon@profesores.valles.udg.mx

Abstract. Blood vessel extraction is an important step for abnormality detection and for obtaining good retinopathy diabetic diagnosis in digital retinal images. The use of filter bank has shown to be a powerful technique for detecting blood vessels. In particular, the Matched Filter is appropriate and efficient for this task and in combination with other methods the blood vessel detection can be improved. We propose a combination of the Matched Filter with a segmentation strategy by using a Cellular Automata. The strategy presented here is very efficient and experimentally yields competitive results compared with others methods of the state of the art.

Keywords: Matched filter, cellular automata, blood vessel detection, diabetic retinopathy.

1 Introduction

Diabetic Retinopathy (DR) is a common complication associated with diabetes. Diabetic Retinopathy is the main cause of blindness among adult people. Then, early Diabetic Retinopathy detection is crucial for decreasing vision loss. Many techniques have been developed in order to detect DR, see for instance [19,17] and references therein. Most of the automatic DR detection methods also detect normal fundus regions, such as optic disk, fovea, macula and blood vessels. In order to detect retinal lesions, this information (normal fundus regions) is removed due to their similarity, in color and shape appearance, with abnormal retinal areas.

In this work, we propose an automatic blood vessel detection method. Our method has three stages. In the first stage, we enhance the blood vessel regions by using a filter bank. In particular, we use the Matched Filter (MF) reported in [2] due to its efficiency. The resulting image is then thresholded to produce a binary segmentation (second stage). However, thresholding techniques may yield different segmentation results when applying to retinal images with different luminance conditions, or even when using different fundus camera. In order to get good segmentation results, we pay special attention to thresholding techniques

[15]. Finally, in the third state we improve the previous segmentation by using a Cellular Automata [20,7].

There are several strategies for blood vessel extraction. Some of them are based on: morphological operators [9,21], Neural Networks [13], region growing [12], Gabor Filters [14], second-order Gaussian Filter [3] and Matched Filter [2,5,22,23]. Among these methods, the Matched Filter [2] has proven to be very efficient. Most of these methods combine different strategies in order to improve the vessel extraction results. Recently, Zhang et al. [23] proposed a combination of Matched Filter [2] with first-order Gaussian derivative [8]. This method is very efficient and achieves good vessel detection results. The reader can find a study of optimal parameters for obtaining a Matched Filter bank for vessel detection in Ref. [1]. There are two main differences between the proposal in Ref. [23] and our approach. In Ref. [23] Zhang et al. use first-order derivative of Gaussian in order to reduce the problem of strong response of Matched Filter in non-vessel edges. In our algorithm, we pay special attention to the thresholding technique [15]. Instead of using a threshold directly, we relax an automatically obtained threshold value in order to obtain two thresholds: one for the object (vessels) and the other for the background (non-vessels). After applying both thresholds, we are in condition to use a segmentation method. In particular, we use a Cellular Automata [20] as segmentation method. These differences are the principal contributions of this paper in the context of blood vessel detection. We show experimentally that these modifications yield competitive vessel detection results compared with algorithms of the state of the art.

This paper is organized as follows. Section 2 describes our algorithm. Here, we briefly review the Matched Filter and the Cellular Automata algorithm. In Section 3 we present the experimental results. Finally, Section 4 presents our conclusions.

2 Our Proposal

The proposed method contains three stages. Firstly, the contrast between vessel and non-vessel is enhanced by applying a filter bank. Then, we relax an automatic threshold value, in order to classify pixels with high confidence of belonging to blood vessel or non-vessel. Pixels with low confidence are finally classified by using a segmentation method.

2.1 Filter Bank

Many algorithms use filter banks for increasing the contrast of the original image: Gabor Filter, Second order derivative of Gaussian, ... etc. We choose the Matched Filter because is simple and very efficient. The Matched Filter was proposed in [2] and was designed for blood vessel detection in fundus images. The Matched Filter takes into account that the grey level profiles of cross section of blood vessel are similar to a Gaussian function. Although, the intensity profiles can vary in width and intensity, they can be modeled using the Gaussian function:

$$f(x, y) = a \left(1 - b e^{-\frac{d^2}{2\sigma^2}} \right), \quad (1)$$

where a, b, σ are parameters that allows us to model the intensity and width of the vessels, d is the distance from (x, y) to the line passing through the center of the vessel. The reader can find the details in Ref. [2]. Based on the previous function, Eq. (1), the kernel for the Matched filter can simply be designed as follows:

$$K(x, y) = -e^{-\frac{d^2}{2\sigma^2}}. \quad (2)$$

Now, considering a set of orientations θ_i $i = 1, 2, \dots, m$ and rotating the kernel in Eq. (2) using these orientations, we obtain the following set of convolution kernels K_i at a given scale σ

$$\mathbf{t} = [x, y]', \quad (3)$$

$$R_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{pmatrix}, \quad (4)$$

$$\mathbf{z}_i = [u, v]' = R_i \mathbf{t}, \quad (5)$$

$$K_i(x, y) = -e^{-\frac{u^2}{2\sigma^2}}, \forall \mathbf{z}_i \in \mathcal{V}, \quad (6)$$

where the neighborhood $\mathcal{V} \stackrel{\text{def}}{=} \{(u, v) : |u| < 3\sigma, |v| < \frac{L}{2}\}$, L is the length of the neighborhood along the y axis, σ represents the scale of the filter and is related with the vessel profile width and R_i is a rotation matrix with orientation θ_i . Finally, a zero-mean normalization is applied to the Kernel in Eq. (6) to obtain the Matched Filter at orientation θ_i and scale σ :

$$\mu_i = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{z}_i \in \mathcal{V}} K_i(x, y), \quad (7)$$

$$\hat{K}_i(x, y) = K_i(x, y) - \mu_i, \forall \mathbf{z}_i \in \mathcal{V}, \quad (8)$$

where $|\mathcal{V}|$ is the cardinality of \mathcal{V} and μ_i is the mean value of the Kernel K_i in the neighborhood \mathcal{V} . In [2] is assumed that the width of the blood vessel is equal to 2σ .

In summary, the Matched Filter depends on three parameters σ , L and θ , i.e., scale, length and orientation respectively. As the scale σ is related with the vessel width then, the value of the scales are typically taken in the interval $[1, 3]$, L is in $[7, 15]$ and $\theta \in [0^\circ, 180^\circ]$. In practice, one uses a discretization of these intervals, for example, $\sigma \in \{1.0, 1.1, 1.2, \dots, 2.9, 3.0\}$, $L \in \{7.0, 7.1, 7.2, \dots, 14.9, 15.0\}$ and $\theta \in \{0^\circ, 15^\circ, \dots, 165^\circ\}$. Since we discretize the parameter space, then the parameter search space is finite and not very large. So, an exhaustive search in the parameter space is enough to find the best parameters, i.e., a parameter training step.

In order to build a Matched Filter Bank we must provide a set of scales $\{\sigma_i\}_{i=1,2,\dots,n}$, a set of lengths $\{L_i\}_{i=1,2,\dots,n}$ and a set of orientations $\{\theta_i\}_{i=1,2,\dots,m}$.

The number of scales and lengths is the same and is equal to n , the number of orientation is m and the total number of convolution kernels \hat{K}_i of the filter bank is equal to $n * m$. The parameter L_i can be selected based on σ_i . If σ_i is small then L_i is relatively small and viceversa [23]. Using this strategy we can eliminate one parameter. The orientations can simply be defined as the number of orientations (m) or the angular resolution ($\frac{180^\circ}{m}$). According to our experiments and to previous works [2,1] 12 orientations, $m = 12$, produce good experimental results, i.e., an angular resolution of 15° . Finally, the blood vessels are enhanced by convolving the fundus image with the set of kernels of the Matched Filter Bank and retaining only the maximum response, denoted here as M .

2.2 Thresholding Relaxation

A natural way for detecting the blood vessels is to binarize the maximum response M of the Matched Filter. However, this simple idea does not always afford good blood vessels extraction. For that reason, different strategies has been proposed for detecting the retinal vessels. For example, Zhang et al. in [23] propose a pixel-wise threshold based on the first-order Gaussian derivative whereas Hoover et al. [5] propose an iterative threshold probing technique in which a sequence of thresholds are applied to the maximum response.

In this work, we propose another strategy. First, we compute an automatic threshold T [15] for segmenting the maximum response of the Matched Filter, Subsection 2.1. In the experiment we use the *minimum error thresholding* technique proposed by Kittler and Illingworth [6] because it produces the best vessel detection. Then, we relax the previous threshold obtaining two new thresholds T_v, T_n for vessels and non-vessels respectively. These thresholds are defined in the following way:

$$\begin{aligned} T_v &= \mu_v T, \\ T_n &= \mu_n T, \end{aligned}$$

where μ_n, μ_v are parameters that satisfy $0 < \mu_n < \mu_v$. T_v is a *threshold above* [16] for vessel region, i.e., pixels with maximum response greater than T_v have high confidence to belong to the vessel region and are classified as *vessel*. T_n is a *threshold below* [16] for non-vessel region, i.e., pixels with maximum response less than T_n have high confidence to belong to the non-vessel region and are classified as *non-vessel*. Pixels with maximum response in $[T_n, T_v]$ are not classified yet, because they have low confidence to belong to vessel or non-vessel region. The non-classified region is a band between the vessel and non-vessel region. In order to classify these pixels we propose a segmentation stage, see next Section.

2.3 Segmentation

Pixels with low Matched Filter maximum response still need to be classified. To this end, we suggest using a segmentation algorithm that allows us to incorporate spatial information and that also takes into account the contrast between

blood vessels and non-vessels. In particular, in this work we use the “GrowCut” Cellular Automata algorithm [20,7] because it is computationally efficient and experimentally yields good results.

A Cellular Automata is a triplet $A = (S, \mathcal{N}, \delta)$ where S is a non-empty state set, \mathcal{N} is a neighborhood system and δ is a transition function from the state t to state $t + 1$. Let \mathcal{L} be the lattice of the image and $r \in \mathcal{L}$ a pixel of the image. A cell state $S_r = (l_r, \theta_r, F_r)$ is a triplet where l_r is the current state, $\theta_r \in [0, 1]$ is the ‘strength’ and F_r is a feature vector, defined by the image, for details see Refs [20,7]. If \mathcal{N}_r denotes the set of neighbors of the pixel r and $g : + \rightarrow +$ is a decreasing function bounded to $[0, 1]$, then the transition function of the “GrowCut” is given by the following algorithm:

Algorithm 1. Cellular Automata evolution rule

```

for all  $r \in \mathcal{L}$  do
     $l_r^{t+1} = l_r^t$ 
     $\theta_r^{t+1} = \theta_r^t$ 
    for all  $s \in \mathcal{N}_r$  do
        if  $g(\|F_r - F_s\|)\theta_s^t > \theta_r^t$  then
             $l_r^{t+1} = l_s^t$ 
             $\theta_r^{t+1} = g(\|F_r - F_s\|)\theta_s^t$ 
        end if
    end for
end for

```

In particular, we use as \mathcal{N}_r the set of the first neighbors of r , i.e., $\mathcal{N}_r = \{s \in \mathcal{L} : |s - r| = 1\}$ and as $g(\cdot)$ the following function:

$$g(x) = \frac{\gamma}{\gamma + x^2},$$

where γ is a positive parameter, in experiments we use $\gamma = 10^{-2}$. The feature vector is just the maximum response, i.e., $F_r = M_r$, $\forall r \in \mathcal{L}$. The initial state $\forall r \in \mathcal{L}$ is set as follows:

$$l_r^0 = \begin{cases} 1 & \text{if } M_r > T_v \\ -1 & \text{if } M_r < T_n \\ 0 & \text{otherwise,} \end{cases}$$

$$\theta_r^0 = \begin{cases} 1 & \text{if } M_r \notin [T_u, T_v] \\ 0 & \text{otherwise.} \end{cases}$$

3 Experimental Results

In this section, we compare our results with other methods of the state of the art. In order to evaluate the algorithm presented here, we test the proposal with two databases available online: DRIVE and STARE databases. DRIVE database is available at <http://www.isi.uu.nl/Research/Databases/DRIVE/> and STARE

database at <http://www.ces.clemson.edu/~ahoover/stare/>. Both databases are comprised by 20 fundus images, their corresponding masks (or region of interest) and hand-labelled images by two human experts. The image size in the STARE database is 700×605 pixels and the image size in the DRIVE database is 584×565 pixels. In both cases, the original images are in color.

In order to compare different detection algorithms [11,18,23] we use the following performance measures: True Positive Rate (TPR), False Positive Rate (FPR) and Accuracy (ACC),

$$\begin{aligned} TPR &= \frac{TP}{TP + FN}, \\ FPR &= \frac{FP}{FP + TN}, \\ ACC &= \frac{TP + TN}{TP + FP + FN + TN}, \end{aligned}$$

where TP , FP , FN , TN are the number of True Positive, False Positive, False Negative and True Negative respectively. The TPR or *sensitivity* can be interpreted as the capacity of the algorithm for correctly detecting blood vessels. FPR is related with the *specificity* ($FPR = 1 - \text{specificity}$). The *specificity* can be interpreted as the probability of detecting the non-vessel region whereas the *accuracy* (ACC) is the probability for simultaneously detecting vessel and non-vessel regions.

Table 1. Vessel detection results for the STARE dataset

Method	TPR	FPR	ACC
2nd Human observer	0.8949	0.0610	0.9354
Hoover [5]	0.6751	0.0433	0.9267
Soares [18]	0.7165	0.0252	0.9480
Mendonca [11]	0.6996	0.0270	0.9440
MF-FDOG [23]	0.7177	0.0247	0.9484
Martinez-Perez [10]	0.7506	0.0431	0.9410
MF-MET	0.7380	0.0604	0.9184
MFCA	0.7606	0.0599	0.9212

As it has been noted before in different works [18] and references therein, the green channel of retinal images shows the best vessel/background contrast while the red and blue channels are very noisy and have a lower contrast. Therefore, we use the image green channel in all experiments. Similar to the previous work in Ref. [23] we use as ground truth the hand-labelled image by the first expert.

In the experiments we use the following parameters: the angular resolution is 15° , $\sigma = 1.4$ (scale), $L = 11$ (length), $\mu_v = 1.05$ (threshold factor for the blood vessels) and for the threshold factor for the non-vessel region we use $\mu_n = 0.84$ for the DRIVE database and $\mu_n = 0.92$ for the STARE database, the number of iterations of the Cellular Automata is 10, see Section 2. All these parameters were found by minimizing the segmentation error

over the discrete parameter space defined by $\sigma \in \{1.0, 1.1, 1.2, \dots, 3.0\}$, $L \in \{7.0, 7.1, 7.2, \dots, 15.0\}$, $m \in \{3, 4, 5, 6, 8, 9, 10, 12, 15\}$, $\mu_n \in \{0.8, 0.81, \dots, 0.99\}$ and $\mu_v \in \{1.01, 1.02, \dots, 1.2\}$.

Table 2. Vessel detection results for the DRIVE dataset

Method	TPR	FPR	ACC
2nd Human observer	0.7761	0.0275	0.9473
Soares [18]	0.7283	0.0212	0.9466
Mendonca [11]	0.7344	0.0236	0.9452
MF-FDOG [23]	0.7120	0.0276	0.9382
Martinez-Perez [10]	0.7246	0.0345	0.9344
MF-MET	0.6992	0.0459	0.9199
MFCA	0.7620	0.0592	0.9168

In some works, the authors give no detail about the thresholding method, for example in [23]. However, vessels detection can be very sensitive to the thresholding method. We experiment with several global thresholding techniques: Otsu, entropy, moments, concavity, minimum error, intermeans, intermodes,..., etc [15,4]. For this experiment we use all the thresholding techniques provided by the *HistThresh toolbox for MATLAB* available at <http://www.cs.tut.fi/~ant/histthresh/>. Among them, the *minimum error thresholding* technique, proposed by Kittler and Illingworth [6] is the thresholding method that produces the best vessel detection. Therefore, in the experiments we use the *minimum error thresholding* technique for computing T , Subsection 2.2.

Tables 1-2 show the results obtained with our proposal and other methods of the state of the art. We include the result obtained after applying the *minimum error thresholding* technique, Matched Filter with Minimum Error Thresholding (MF-MET), i.e., the first two steps of Section 2. The purpose of including the results of MF-MET is to prove that the Cellular Automata improves the segmentation results. The experimental results of the algorithms of the state of the art are taken from Zhang et al. [23], recently published, see also Refs. [5,18,11,10,23]. As we can see in the first column of tables 1-2, the method proposed here, MFCA, presents the best vessel detection results. Second and third columns of tables 1-2 show that our proposal yields competitive results compared with methods of the state of the art for detecting vessels and non-vessel regions. Note that the sensitivity of the method by using Cellular Automata has increased the True Positive Rate (TPR) vessel detection percentage; nevertheless, the False Positive Rate (FPR) has increased as well. The accuracy ACC therefore diminishes a little bit. Hence, there is a trade off between TPR and FPR. On the other hand, the consuming time of the proposed method is low. The *running time average* per image in the STARE database by using the MF-MET algorithm is 0.4 seconds and the *running time average* of the MFCA algorithm is 1.0 second. The implementation is in Matlab 7.10 and we run the script in a Core i7 2.2 GHz with 4 GB of RAM.

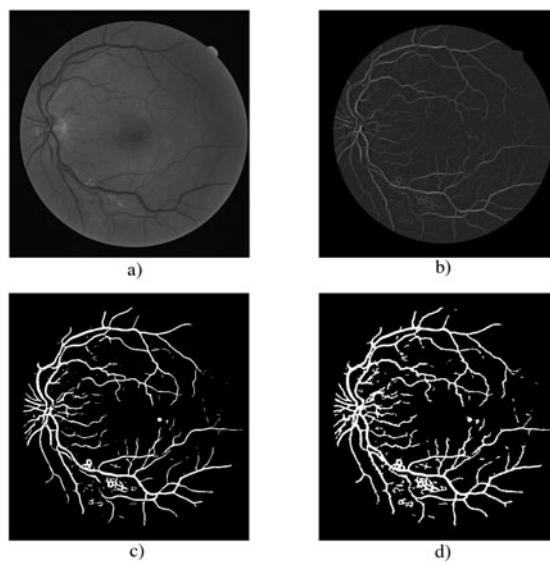


Fig. 1. ‘03-test’ image of the DRIVE database. a) Green channel, b) Maximum response to Matched Filter, c) Segmentation using MF-MET, d) Segmentation using MFCA

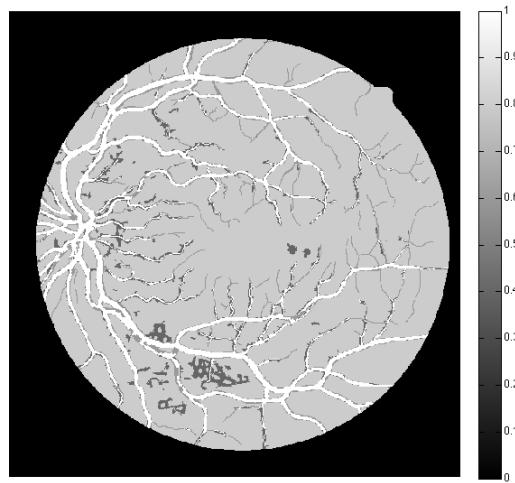


Fig. 2. Confusion image corresponding to ‘03-test’ image of the DRIVE database, see Fig. 1 a). True Positive: gray level image equal to 1.0, True Negative: gray level image equal to 0.8, False Negative: gray level image equal to 0.6 and False Positive: gray level image equal to 0.4.

Figures 1-2 illustrate the result obtained by using the MFCA algorithm to the ‘03-test’ image of the DRIVE database. The ‘03-test’ image was randomly selected from the DRIVE database for illustration purposes. Figure 1 panel a) shows the green channel of the original image, panel b) depicts the maximum response of the Matched Filter Bank, panel c) is the segmentation after applying the *minimum error thresholding* and finally, panel d) shows the blood vessel detection of the MFCA algorithm. Figures 2 depicts the *confusion image*, defined here as the image of TP, TN, FP and FN results. This representation is very important because it allows us to detect the regions where the algorithm obtains good segmentation results and where the algorithm is wrong. The gray levels 1.0 and 0.8 correspond to regions where the MFCA algorithm correctly detects the blood vessels and non-vessels regions respectively, i.e., TP and TN results. The gray levels 0.6 and 0.4 correspond to regions where the MFCA algorithm is wrong, i.e., FP and FN results. From this figure, we can see that the main limitation of our proposal is for detecting thin vessels. A possible solution for this problem is to use different scales in the MF bank, similar to the previous work by Zhang et al. [23]. Another problem is that our algorithm detects some retinal lesions as vessels. This is due to the MF bank has high response in regions with high contrast, i.e., hard exudates, see Fig. 1 panels a) and b). We note that unlike other algorithms, our method does not pre process or post process the image. Therefore, we can improve our results by making a pre-processing of the image in order to eliminate regions with high contrast, because it is well-known that regions with high contrast correspond to *exudates* or the *optic disc*. A simple alternative is to remove the high contrast regions using a thresholding technique, and a more sophisticated alternative could be to design filters that have low response in regions with high contrast.

4 Conclusion

In this paper we present a three-stage algorithm for automatic blood vessel extraction. We show experimentally that our proposal presents the best vessel detection results compared with methods of the state of the art. The proposed algorithm is simple, computationally efficient and easy to implement. The non-vessel region extraction and the simultaneous detection of vessel and non-vessel regions are competitive with the methods of the state of the art. The main problem of our method is for detecting thin vessels. Another difficult is that our algorithm detects some lesions as vessels. One way to solve this problem is to make a pre-processing or a post-processing to eliminate areas of high contrast. Despite of our strategy was proposed in the context of retinal images for detecting blood vessels it is not limited to this kind of images. There are several image processing tasks in which this algorithm can be used, for example, for road detection in satellite imagery and for carbon nanotube detection in order to study properties of nanomaterials.

Acknowledgments. The authors wish to thank Dra. Graciela Farias and Dr. Antonio Costilla for their advices and the constant encouragement for this work. This research was supported in part by CONACYT (Grant 105657).

References

1. Al-Rawi, M., Qutaishat, M., Arrar, M.: An improved matched filter for blood vessel detection of digital retinal images. *Computers in Biology and Medicine* 37, 262–267 (2007)
2. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on Medical Imaging* 8(3), 263–269 (1989)
3. Gang, L., Chutatape, O., Krishnan, S.M.: Detection and measurement of retinal vessels in fundus images using amplitude modified second-order Gaussian filter. *IEEE Transactions on Biomedical Engineering* 49(2), 168–172 (2002)
4. Gorman, L.O., Sammon, M.J., Seul, M.: Practical Algorithms for Image Analysis: Description, Examples, Programs and Projects. Cambridge University, New York (2008)
5. Hoover, A., Kouznetsova, V., Goldbaum, M.: Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging* 19, 203–210 (2000)
6. Kittler, J., Illingworth, J.: Minimum error thresholding. *Pattern Recognition* 19, 41–47 (1986)
7. Konushin, V., Vezhnevets, V.: Interactive image colorization and recoloring based on coupled map lattices. In: Graphicon, pp. 231–234 (2006)
8. Li, Q., Zhang, L., You, J., Zhang, D., Bhattacharya, P.: Dark line detection with line width extraction. In: IEEE International Conference on Image Processing, ICIP 2008, pp. 621–624 (2008)
9. Luo, Z., Liu, Z., Li, J.: Micro-Blood Vessel Detection using K-means Clustering and Morphological Thinning. In: Liu, D., et al. (eds.) ISNN 2011, Part III. LNCS, vol. 6677, pp. 348–354. Springer, Heidelberg (2011)
10. Martinez-Perez, M., Hughes, A., Thom, S., Bharath, A.A., Parker, K.: Segmentation of blood vessels from red-free and fluorescein retinal images. *Medical Image Analysis* 11, 47–61 (2007)
11. Mendonca, A., Campilho, A.: Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *IEEE Transactions on Medical Imaging* 25(9), 1200–1213 (2006)
12. Palomera-Pérez, M.A., Martínez-Pérez, M.E., Benítez-Pérez, H., Ortega-Arjona, J.L.: Parallel multiscale feature extraction and region growing: application in retinal blood vessel detection. *IEEE Transactions on Information Technology in Biomedicine*, 500–506 (2010)
13. Perfetti, R., Ricci, E., Casali, D., Costantini, G.: Cellular Neural Networks with Virtual Template Expansion for Retinal Vessel Segmentation. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 54, 141–145 (2007)
14. Rangayyan, R.M., Ayres, F.J., Oloumi, F., Oloumi, F., Eshghzadeh-Zanjani, P.: Detection of Blood Vessels in the Retina Using Gabor Filters. *J. Electronic Imaging* 17(2), 023018 (2008)

15. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging* 13(1), 146–168 (2004)
16. Shapiro, L.G., Stockman, G.C.: *Computer Vision*. Prentice Hall, Upper Saddle River (2001)
17. Singh, N., Tripathi, R.C.: Automated early detection of diabetic retinopathy using image analysis techniques. *International Journal of Computer Applications* 8(2), 18–23 (2010); published By Foundation of Computer Science
18. Soares, J.V.B., Le, J.J.G., Cesar, R.M., Jelinek, H.F., Cree, M.J., Member, S.: Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification. *IEEE Transactions on Medical Imaging* 25, 1214–1222 (2006)
19. Usher, D., Dumskyj, M., Himaga, M., Williamson, T.H., Nussey, S., Boyce, J.: Automated detection of diabetic retinopathy in digital retinal images: a tool for diabetic retinopathy screening. *Diabetic Medicine a Journal of the British Diabetic Association* 21(1), 84–90 (2004)
20. Vezhnevets, V., Konouchine, V.: “GrowCut” - Interactive Multi-Label N-D Image Segmentation By Cellular Automata. In: *Graphicon 2005*, pp. 150–156. Novosibirsk Akademgorodok, Russia (2005)
21. Zana, F., Claude Klein, J.: Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE Transactions on Image Processing* 10, 1010–1019 (2001)
22. Zhang, B., Li, Q., Zhang, L., You, J., Karray, F.: Retinal vessel centerline extraction using multiscale matched filter and sparse representation-based classifier. In: *Second International Conference on Medical Biometrics, ICMB 2010*, pp. 181–190 (2010)
23. Zhang, B., Zhang, L., Zhang, L., Karray, F.: Retinal vessel extraction by matched filter with first-order derivative of Gaussian. *Computers in Biology and Medicine* 40, 438–445 (2010)

Computer Assisted Diagnosis of Microcalcifications in Mammograms: A Scale-Space Approach

Alberto Pastrana Palma, Juan Francisco Reyes Muñoz,
Luis Rodrigo Valencia Pérez, Juan Manuel Peña Aguilar,
and Alberto Lamadrid Álvarez

Universidad Autónoma de Querétaro, Laboratorio de Gestión Tecnológica
Cerro de las Campanas, Querétaro, México
alberto@apastrana.com, {paleph212,royvalper,juan_manuelp}@hotmail.com,
alberto.lamadrid@gmail.com
<http://www.uaq.mx>

Abstract. Computer Assisted Diagnosis (CAD) is rapidly reaching worldwide acceptance in different fields of medicine. Particularly, CAD has found one of its main applications in breast cancer diagnosis where the detection of microcalcifications in women breasts is typically associated with the presence of cancer. In this paper, a method for automatic breast contour detection is presented as a pre-processing step for microcalcification detection. Then, a combination of scale-space algorithms are used to locate candidate regions of microcalcifications and a significant percentage of false positives are finally discriminated via thresholding. Detected regions using this method have been found to describe 91.6% of microcalcifications from the MIAS database with an average specificity of 97.30%.

Keywords: mammography, microcalcifications, scale-space, sieve.

1 Introduction

Breast Cancer is one of the most common causes of dead among women around the world [10]. Generally, a preliminary stage in the detection of breast cancer is to look for minuscule deposits of calcium that are known as microcalcifications via mammographic analysis [3]. In many cases, microcalcifications are associated with the presence of breast cancer in women at an early stage [3]. Therefore opportunistic diagnosis improves live expectancy according to how early the detection has been made. However it is always difficult to find enough radiologists to supply the increasing demand of mammographies. Computer Assisted Diagnosis (CAD) may be used to provide an alternative opinion to that of a radiologist either confirming a diagnosis or bringing the radiologist attention to specific areas that may have been overlooked.

Here we present a scale-space approach to automatically detect microcalcifications on mammograms. The paper is divided into 3 main sections: the first, describes the process of breast segmentation using a combination of contrast enhancement and scale-space filtering. The second section explains how a scale-space filter known as SIEVE may be used to find candidate regions of microcalcifications. Then, section three describes how a collection of thresholding techniques can effectively reduce more than 97% of the initial number of candidate regions whilst preserving around 92% of the regions marked as microcalcifications. Figure 1 illustrates a general view of the whole process.

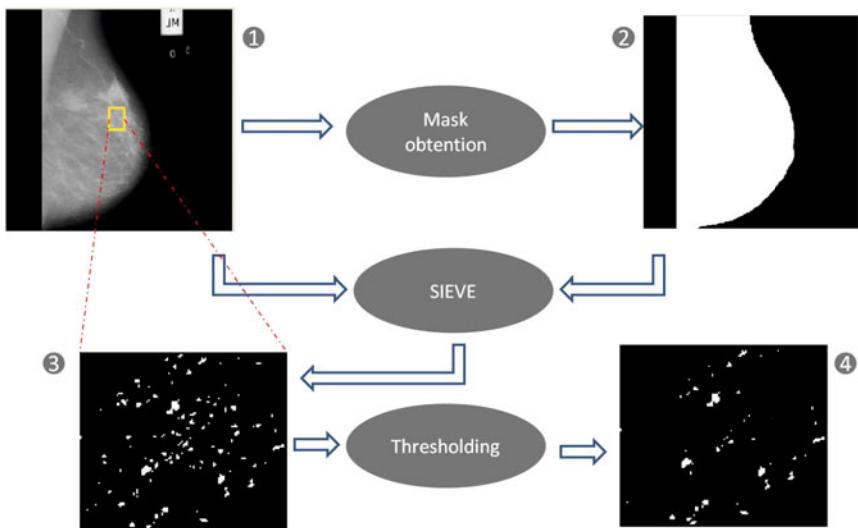


Fig. 1. A general diagram of the process. First an input image is introduced (1) from which a breast segmentation is obtained and used as mask (2). The SIEVE is then used to detect candidate regions (3) and finally, thresholding is applied to obtain regions with more chance to represent microcalcifications (4).

2 Breast Segmentation

An important step towards the detection of microcalcifications is to segment the breast area from the mammogram in order to reduce the search space of candidate regions and also to eliminate particular annotations or labels that may have been introduced by the image acquisition device. Here, a method for breast segmentation using SIEVE morphological scale-space operators together with contrast enhancement is applied to segment the breast area. SIEVES are a family of algorithms which iteratively detect local maxima and/or minima regions at different scales in the image. Our results have shown that the application of such

operators on mammograms help to reduce noise whilst eliminating superimposed figures such as device-introduced labels. Coupled with this, a variety of contrast enhancement techniques are applied to enable easy contour detection that is later carried out by thresholding the intensity values of the mammogram.

Gamma correction factor is probably one of the most common ways to enhance image contrast. It describes a function under which image luminance is transformed in order to provide a desired output image. Gamma correction factors should be chosen according to the problem domain under consideration. Typically, it is defined by the following power-law expression:

$$V_{out} = V_{in}^{\gamma} \quad (1)$$

Where V is the input and output luminance for a given image pixel. The result of varying γ between an interval [low, high] from the luminance domain may be described as follows:

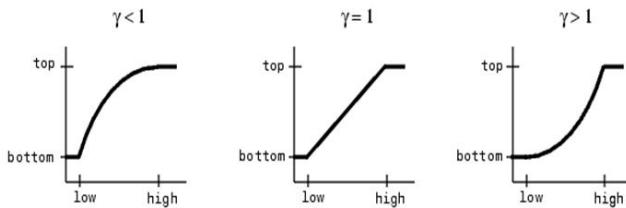


Fig. 2. The effect of different values of γ on image luminance

Mammographies describe the degree to which human tissue is penetrated by radiation. Soft tissues like human skin appear darker than hard tissues such as bones. Therefore image background is extremely black. Bearing this in mind, in order to segment the breast area in the mammogram, it is sensible to choose a gamma factor of nearly zero, thus increasing the luminance of everything but the background. In our experiments we found that a factor of $\gamma=0.01$ for values low=0.055 and high=0.65 are convenient choices. Figure 3 describes the result of contrast enhancement over a mammography

3 Using SIEVE Filtering on Mammograms

Scale-space offers a wide variety of techniques to analyze specific image features. Applying morphological operators over scale-space presents the advantage of preserving original image contours, a property known as 'scale-space causality'. Particularly, the SIEVE algorithm is a robust process that is capable of eliminating noise up to some degree [1]. Also, as opposite to conventional morphological filters, SIEVES do not require a particular structuring element but rather operate over connected-set regions of specific scales, thus avoiding the introduction of

artifacts produced by the structuring element in the resulting image [7]. SIEVE filtering has been used before on breast cancer[2] but only focusing on the detection of abnormal masses at a, perhaps, late stage of the disease rather than microcalcifications at an early one. Lately, wavelets seem to be taken popularity as a method for microcalcifications detection[9][8] but not much has been said of a scale-space approach for this same problem.



Fig. 3. An example of contrast enhancement. The input image is shown on the left. In the middle the same image after only gamma correction is applied. The rightmost image shows the result of using the SIEVE to eliminate regions out of the breast such as the annotations made by the acquisition device.

Figure 4 depicts the SIEVE algorithm. The original signal is repeatedly filtered by locating and merging local extrema. Initially maxima and minima of size 1 is removed, then extrema of size 2 and the process continues until there is nothing else to merge. A hierarchy of granularities is formed by the differences of subsequent stages of the algorithm.

Resulting images of the SIEVE filter may be described as a simplified version of the original where detail has been removed up to some level. However, the focus of the process proposed here is on the differences between resulting images of the algorithm at different stages (the granularities). A detailed description of the SIEVE algorithm is found in [1] and a very complete analysis of SIEVE granularities was reviewed on [5].

Some acquisition devices automatically insert labels on the mammograms which in fact may be seen as noise by the candidate region detection process. Eliminating such regions is relatively simple since, after SIEVE filtering, they appear as local maxima of a size not as large as the whole breast segment. We simply eliminate local maxima whose size is less than 1500 pixels. The result is typically a large unique segment representing the breast area. Figure 3 illustrates an example of the application of SIEVE on breast segmentation.

Typically, microcalcifications appear as very bright small regions in the mammogram. Therefore, SIEVE is quite fitted to the purpose since it is designed to

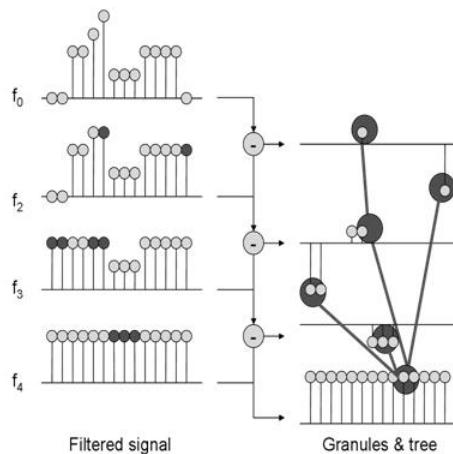


Fig. 4. The SIEVE algorithm and the resultant granularity tree. The input signal f_0 is repeatedly filtered by locating local extrema and merging it to their nearest neighbor in terms of intensity value. The difference between two stages of the algorithm (e.g. f_2 and f_3) produces granularities that can be arranged in a hierarchical manner.

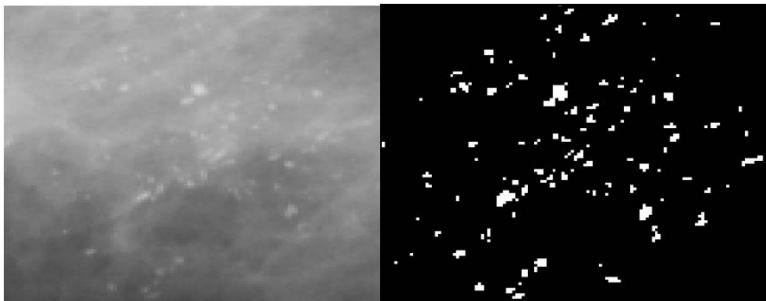


Fig. 5. On the left a mammogram section depicting many microcalcifications. On the right the corresponding mask obtained from the SIEVE granularities. In many cases the extrema indeed represents microcalcifications.

locate and merge local maxima or minima (bright or dark regions) at different increasing size. Obviously, not all the resultant regions may be labeled as microcalcifications and it becomes necessary to reduce the search space by applying criteria based on the definition of microcalcifications [10]. One of such criteria is to only select regions of size less than 2 mm based on the conversion rate pixel-to-millimeter that must be attached to each mammogram. Figure 5 describes the result of applying SIEVE to a mammogram depicting microcalcifications.

4 Thresholding over Candidate Regions

Each candidate region along with their surrounding area is characterized by a set of descriptors that statistically describe gray level. We have also taken into account the surrounding tissue of the region of interest since it has been proved that such information is also valuable on the detection of microcalcifications[6]. In our experiments we have used the following features:

- F_{avg} Gray level average of only the candidate region
- B_{avg} Gray level average of only the area surrounding the candidate region.
- F_{std} Standard deviation of the candidate region
- G_{fb} Gradient (absolut difference) between F_{avg} and B_{avg} .

The surrounding area is a square window of side length= $2s$ where s is the size (in pixels) of the microcalcification.

The regions are then described with a feature set of 4 elements for which once again criteria based on the medical description of microcalcifications is applied. In our experiments all intensity values were normalized to values from 0 to 1 where 1 represents the maximum intensity level of 255. Four empirical rules are employed here to reduce the number of candidate regions:

1. The greater the value of G_{fb} , the more likely this region is to be labeled as microcalcification by a radiologist. However some microcalcifications may not appear as evident and therefore the value of G_{fb} has to be sufficiently small to cover most microcalcifications without including a great deal of false positives. In our experiments this value was set to 0.039
2. The greater the value of F_{std} , the more likely this region is to not represent noise. This threshold was set to 0.01
3. If the value of B_{avg} is high, this indicates that the background intensity is also high and therefore a smaller G_{fb} must be considered as a criteria for the first rule. Figure 6 illustrates this behaviour.

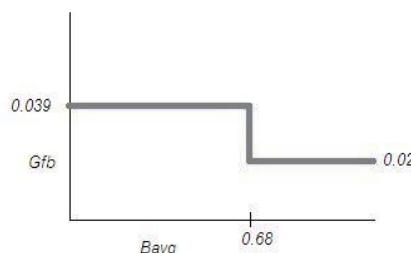


Fig. 6. The gradient function used in our experiments. When $B_{avg}=0.68$ then G_{fb} descends from 0.039 to 0.020.

Table 1. Reduction percentage of candidate regions

Image	Candidate regions after SIEVE	Candidate Regions after thresholding	Reduction Percentage	Percentage of microcalcifications retained
Mdb209	11253	290	2.58%	100.00%
Mdb211	11445	325	2.84%	100.00%
Mdb213	7527	129	1.71%	93.33%
Mdb216	8310	454	5.46%	95.24%
Mdb218	11286	293	2.60%	100.00%
Mdb219	15118	305	2.02%	84.62%
Mdb222	8259	279	3.38%	66.67%
Mdb223	7270	262	3.60%	100.00%
Mdb231	19215	291	1.51%	100.00%
Mdb233	10225	408	3.99%	96.92%
Mdb236	9974	366	3.67%	100.00%
Mdb238	12317	156	1.27%	100.00%
Mdb239	9632	312	3.24%	50.00%
Mdb241	8507	346	4.07%	100.00%
Mdb245	7930	276	3.48%	100.00%
Mdb249	11419	328	2.87%	100.00%
Mdb252	12751	208	1.63%	100.00%
Mdb253	10962	313	2.86%	80.00%
Mdb256	16672	163	0.98%	75.00%
TOTALS			2.83%	91.67%

5 Results

In order to obtain the thershold values described above, three randomly selected images (Mdb226, Mdb227 and Mdb248) from the MIAS database (Mammographic Image Analysis Society) were visually inspected by an expert radiologist[11]. For the assessment of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) we have used the groundtruth provided by Oporto in [4]. Results are described in Table 1 where the main finding is the significant reduction of 97% of the initial number of candidate regions on each image whilst retaining an average of 91.6% of the microcalcifications already diagnosed for such database.

One might like to analyze the results from a different perspective. Table 2 describes the specificity and sensibility of our detection process where the average sensibility is 91.67% and average specificity is 97.30%. A key factor for this performance is the great reduction of False Positives due to the thresholding process. If one looks at the thresholds of the feature set used on these experiments, it becomes apparent that they were adjusted to describe small, bright regions that are quite distinguishable from their background, a model that seems to fit well to describe microcalcifications at least for this image dataset.

Table 2. Sensibility and Specificity

Image	True Negatives (TN)	True Positives (TP)	False Positives (FP)	False Negatives (FN)	Sensibility $TP/(TP+FN)$	Specificity $TN/(TN+FP)$
Mdb209	10963	19	271	0	100.00%	97.59%
Mdb211	11120	2	323	0	100.00%	97.18%
Mdb213	7397	14	114	1	93.33%	98.48%
Mdb216	7855	20	433	1	95.24%	94.78%
Mdb218	10993	3	290	0	100.00%	97.43%
Mdb219	14811	11	292	2	84.62%	98.07%
Mdb222	7979	2	276	1	66.67%	96.66%
Mdb223	7008	6	256	0	100.00%	96.48%
Mdb231	18924	2	289	0	100.00%	98.50%
Mdb233	9815	63	343	2	96.92%	96.62%
Mdb236	9608	4	362	0	100.00%	96.37%
Mdb238	12161	5	151	0	100.00%	98.77%
Mdb239	9319	1	310	1	50.00%	96.78%
Mdb241	8161	13	333	0	100.00%	96.08%
Mdb245	7654	43	233	0	100.00%	97.05%
Mdb249	11091	11	317	0	100.00%	97.22%
Mdb252	12543	7	201	0	100.00%	98.42%
Mdb253	10648	4	308	1	80.00%	97.19%
Mdb256	16508	3	159	1	75.00%	99.05%
TOTALS					91.67%	97.30%

6 Conclusion

This paper has presented two main developments: a method for automatic breast contour detection and a method to locate microcalcifications on mammograms. Breast contour detection was achieved through contrast enhancement and elimination of labels produced by the acquisition devide were eliminated using SIEVE algorithm. In order to initially find candidate regions, the SIEVE algorithm was used to detect local maxima of size less than 2mm. False positives were later discriminated via thresholding of a feature set of 4 elements. Detected regions using this method have been found to describe 91.6% of microcalcifications from the MIAS database with an average specificity of 97.30%. Bearing in mind that the aim here is to be able to focus the attention of the radiologist on certain areas of the mammogram, the results presented in this paper have shown that the set of regions finally obtained as a result of the whole process effectively contain most of the microcalcifications and thus assistance to the radiologist may be provided.

References

1. Bangham, J.A., Chardaire, P., Ling, P., Pye, C.J.: Multiscale nonlinear decomposition: the sieve decomposition theorem. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18, 518–527 (1996)

2. Zwiggelaar, R., Taylor, C.J.: Abnormal Masses in Mammograms: Detection Using Scale-Orientation Signatures. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 570–577. Springer, Heidelberg (1998)
3. Calero Cuerda, F.: Factores de Riesgo en Cáncer de Mama. *Journal de Progresos de Obstetricia y Ginecología* 42(90), 9065–9088 (1999)
4. Oporto-Díaz, S., Hernández-Cisneros, R., Terashima-Marín, H.: Detection of Microcalcification Clusters in Mammograms Using a Difference of Optimized Gaussian Filters. In: Kamel, M., Campilho, A. (eds.) ICIAR 2005. LNCS, vol. 3656, pp. 998–1005. Springer, Heidelberg (2005)
5. Southam, P., Harvey, R.: Texture Granularities. In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 304–311. Springer, Heidelberg (2005)
6. Karahaliou, A., Skiadopoulos, S., Boniatis, Sakellaropoulos, P., Likaki, E., Panayiotakis, G., Costaridou, L.: Texture analysis of tissue surrounding microcalcifications on mammograms for breast cancer diagnosis. *British Journal of Radiology* 80, 648–656 (2007)
7. Pastrana Palma, A., Bangham, J.A.: Correspondence Regions and Structured Images. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 580–589. Springer, Heidelberg (2007)
8. Leena, J.S., Govardhan, A., Baskaran, S.: Classification of Microcalcification in Mammograms using Nonsubsampled Contourlet Transform and Neural Network. *European Journal of Scientific Research* 46(4), 531–539 (2010)
9. Veni, S., Narayananakutty, K.A., Fayas, A.: Employing Hexagonal Wavelets for Detecting Microcalcification in Digital Mammography. *ICGST-GVIP Journal* 11(1) (2011)
10. American Cancer Society: Breast Cancer Screening & Treatment. Position Statement as of February 16 of 2011 (2011)
11. The Mammographic Image Analysis Society (2011),
<http://peipa.essex.ac.uk/info/mias.html> (last visited on May 2011)

Diagnosis in Sonogram of Gall Bladder

Saad Tanveer¹, Omer Jamshaid², Abdul Mannan³, Muhammad Aslam⁴, Ana Maria Martinez-Enriquez⁵, Afraz Zahra Syed⁶, and Gonzalo Escalada-Imaz⁷

¹ School of CS, GIFT University, Gujranwala, Pakistan
saadtanveer@engineer.com

² Department of CE, NFC I. E. T., Multan, Pakistan
umerkhan@engineer.com

³ Department of CE, NFC I. E. T., Multan, Pakistan
mannan.sheikh@gmail.com

⁴ Department of CS & E, UET, Lahore, Pakistan
maslam@uet.edu.pk

⁵ Departmeent of CS, CINVESTAV-IPN, D.F., Mexico
ammartin@cinvestav.mx)

⁶ Department of CS & E, UET, Lahore, Pakistan
afrazsyed@yahoo.com

⁷ Artificial Intelligence Research Institute, IIIA-CSIC, Spain
gonzalo@iiia.csic.es

Abstract. This paper describes the development and testing of a diagnostic system using sonograms. In far flung areas of under developing countries where the availability of specialists is a problem and sometimes not even a possibility, it is highly beneficial to have on site diagnosis computer application to support medical staff. Diagnose of sonograms to identify infected part in offline scenarios is not always easy. Besides, lack of infrastructure does not permit online solutions to be a practical option. We implement a system named Intelligent-Eye (I-Eye) which employs imaging and diagnostic techniques to support in the gallbladder diagnosis, saving time and cost of operating medical procedures. We implemented an algorithm highly capable of being used on different diagnostic ultrasonography machines, generating accurate information reports and diagnosis.

Keywords: Disease diagnosis, image processing. Gall Bladder.

1 Introduction

Health care services has been always a dynamic and challenging frontier for humans, requiring more innovative and ground breaking solutions. The image taken by an ultrasound, generally presents noisy and it is unclear. To read an image and make a proper diagnosis it is necessary to have medical background knowledge, so computer support is always welcome[16]. To have better solution to this problem it is required to replace older machines by advanced ones and to improve the treatment. Our system focuses on the gallbladder disease diagnosis the implemented system is capable of diagnosing major diseases of gall.

Currently, there exist solutions like SonoCalc offered by Sonosite [4], Toshiba's Auto-IMT by Toshiba [5][6] and US PLUSView family of ultrasound products [6]. Some of them offer features like zooming that improves image readability, but they are not good for diagnosis. Such utilities neither identify the malady nor are they able to identify the organ of interest. For instance, SonoCalc offers the diagnostic assistance for thickness carotid artery measurement, patient profile database, and generates reports. Other options include on board features present on ultrasound machines which provide the measurement of objects of interest in the sonogram. However, these functions do not contribute much to increase in accuracy of diagnosis.

Medical services need aid during diagnosis process because no system is integrated with ultrasound equipment or at least a second opinion about certain cases of sonogram under observation is always beneficial. Thus, a diagnostic system is required as suggested in [1]. No ultrasound diagnostic services can be provided in far flung or underdeveloped parts of the world where resources in terms of equipment and expertise are few and are far too stretched out. Both of the above stated problems; presence of medical specialists or expensive machinery or sometime both, require huge investments in time and money. Coupled with lack of computer aid for doctors in diagnosis and report formatting with data basing utilities are area which need improvements. This requires third party software for data basing or reporting. I-Eye is one stop solution for all. The consequence of this is medical resources being scarce in underdeveloped countries deprives the people of these areas from reliable medical diagnosis. Furthermore doctors and emergency staff should have all the help they can get while making a decision about diseases.

To overcome this problem of gallbladder diagnosis using sonograms, we designed, implemented, validated, and tested a knowledge based system, which we named I-Eye. The inputs of I-Eye are sonograms, and then a report is generated with the diagnosis. Permanent integration of I-Eye module is also possible into ultrasound equipment taking automated diagnosis to a whole new dimension. As automation in 2-dimensional ultrasound diagnosis has not been explored much, there is a lot of room for innovative and practical ideas to be implemented making smarter solutions and more reliable diagnosis using computer aided decision making and image processing.

Gall bladder secretes bile juice, and also aids the concentration of juices produced in liver. The organ is found in close proximity of liver. In vertebrates the gallbladder (cholecyst, gall bladder) is small organ that aids mainly in fat digestion and concentrates bile produced by the liver [12][13]. In adults, approximately gallbladder measures 8 centimeters (3.1in) in length and 4 centimeters (1.6 in) in diameter when fully distended [14][15]. It is divided into three sections: fundus, body and neck.

I-Eye uses image processing filters to reduce noise, improving the image quality. Pattern detection techniques and energy distribution are used to determine major gallbladder maladies which can later be upgraded on other human organs.

The rest of the paper is organized as follows: Section 2 describes the architecture and functionality of I-Eye, describing maladies diagnosed along with the algorithm proposed. Section 3 discusses positive outcomes of deployment of I-Eye, as medical diagnostic application with future prospects of the development. Also Section 3 sheds some light on related works elaborating the existing systems, their pros and cons. Also, emphasis is made on the problems that lead to I-Eye innovation. Section 4 concludes and Section 5 explains future prospects.

2 I-Eye Architecture

The ultrasonic imaging is not only used in health care and industry but also in nature by bats and dolphins for navigation and imaging of the environment. Similarly this is used for imaging by ultrasound machines.

I-Eye is a diagnostic utility based on Matlab Framework using Image and Signal processing toolboxes. Sonogram is taken by receiving the ultrasound signal reflected by the tissue and bones that are originated from the transducer which in effect is a transceiver. These are then input into the computer for processing and resultant sonogram is displayed on the screen. Radiologists then diagnose on the basis of their experience. The I-Eye architecture is shown in Figure 1, we can observe the block level abstract model of the above states processes.

The user takes a sonogram of the organ and the sonogram is the input of our system as shown in block diagram in Figure 1, at this stage the Data base is also updated for the patient, so the history can be maintained for later reports. This is performed by using user friendly interface. The image is filtered to remove unwanted noise and image features not required for diagnoses are removed. This includes also the adjustment of the brightness and contrast of the image to further improve the visibility of the objects. Then, in next block the image is subject to processing for detection of organ which is based on energy absorption characteristic of both the malady and organ. This is done in two steps first the organ is isolated from the image and rest of the image is assumed as noise for the system and is discarded.

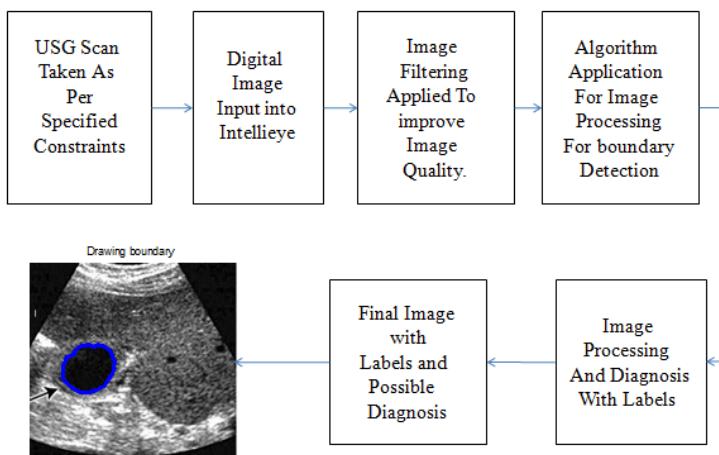


Fig. 1. Block diagram of I-Eye

Once the organ is isolated, the presence of gallstones is checked. Then processing moves to next block that performs combination of image processing which under a set of rules lead to diagnosis of malady, including the measurement of shape of the organ and the size of the gall stones as well as count if gall bladder duplication occurs. The malady is marked on the image and is highlighted. Figure 1 shows the processing

moves to the next block and the marked malady and organ is superimposed on the enhanced image. The possible diagnosis is also given in the final image this concludes the processing. I-Eye can diagnose the following three maladies of gallbladder:

- 1) Normal Gallbladder.** This is as shown in Figure 2(a). The organ is pear shaped with range of length 6-12 [cm] and width of 3-4 [cm]. Wall thickness of a healthy organ is somewhere between 1 to 2 mm. A normal gallbladder shows as a dark area of uniform gray scale value. The volume of this organ should be around 40 to 80 cm³. This information helps I-Eye to detect and notify that no malady in the organ is found.
- 2) Calculi (Gall stones).** Calculi is the occurrence of light objects inside a gallbladder which cast a distinct acoustic shadow this particular condition is commonly known as stones. Two types can arrive, one as shown in Figure 3(b) is calcareous one, Cholesterol calculi is the presence of light objects inside a gallbladder which does not cast acoustic shadow unlike the calcareous version (see Figure 4 (a)).

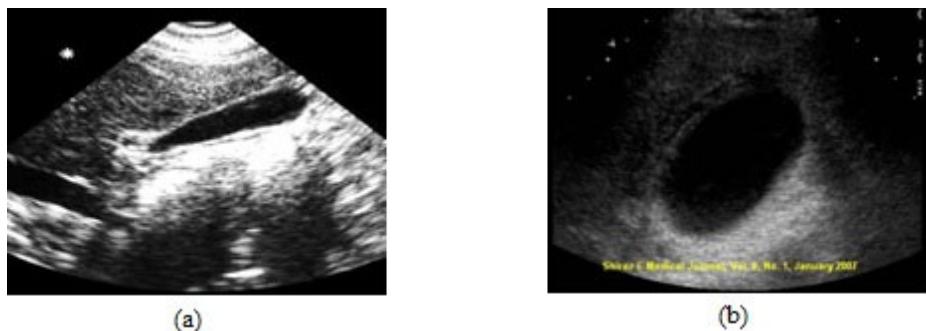


Fig. 2. Sonogram of normal gall bladder (a) and one ailed with hydrocele (b)

- 3) Hydrocele.** Enlargement of a gallbladder is beyond regular size, caused by excessive accumulation of bile. This is the case when the gall bladder loses its original pear shape and most of the cases become a round ball like body. In Figure 2(b) the round body is ailed with hydrocele.
- 4) Gall Bladder Duplication.** It is a rare anomaly in which the patient may have more than one gallbladder, although it is not a malady, but in some cases it is possible that these twins overlap or twist with each other causing severe pain. It is easily observable in a sonogram, showing multiple bodies of gall bladder (see Figure 3 (a)).

Algorithm with working example: The input image of I-Eye goes through the following stages.

- A) The image is fed into the system through the user interface. To adjust the Image lighting this includes the contrast, brightness etc. When image does not have 1% of data saturated in the higher and lower intensities, it is making within the range. The resulting image is then, ready for further processing. The image is converted into gray scale, since gray scale is the true nature of a sonogram. This is done by the “Adjusting intensity of the input image” rule.

```

STARTRULE "Adjusting intensity of the input image"
IF Input(Image) = SonogramImage      /* input the image */
Intensity_1_high And Intensity_1_low (Image) = 1%
THEN Image <=ReDistributeIntensities(Image)
                                /*redistributes the intensities in the image */
ENDRULE

```

The I-Eye engine uses a Rule Base (RB) to dynamically process and analyze sonograms. The RB is independent of the IE code. Thus, it is possible to adapt the RB content by the addition of more rules, removing others that are no longer applicable, and organizing them in different abstract levels. Each rule may be modified/updated by changing its condition part and/or adding new actions. The kernel of the inference engine remains totally independent of the rule base and more precisely of the different modifications applied to the rules (addition, deletion, and update). The inference rules are written in the traditional first order predicate logic (without functions symbols).

B) The filtering noise known as wiener2 [3] was applied to remove inherent noise to sonograms. 2-D adaptive noise-removal filtering is performed. Wiener2 low pass-filters a gray scale image that has been degraded by constant power additive noise. Wiener2 uses a pixel wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel.



(a)



(b)

Fig. 3. The case of duplication is shown (a) and an organ with a gall stone input for the IE for testing (b)

Equations 1, 2, 3 [17] are used to calculate $N \times M$ which is the neighborhood. That is if the pixel and filter is created using the approximations based on above values. In equation (3) v^2 is the noise variance. Wiener2 low pass-filters a gray scale image that has been degraded by constant power additive noise [7]. Wiener2 uses a pixel wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel. C) Then, the object detection starts up at the different threshold [10] levels of object detection. That is based on the object size to be registered. The size is increased by loop control till the registration is started .Till the point no object is registered in the image as the size of object is increased.

This is an algorithm based on slow start and constant increment of threshold level, so when the object is detected the first time to the time it is lost from acquisition creating a set of images.

That is object is in allowable range size of organ then select image from set. Objects detected of sizes smaller than object remove value (noise or reflections) and bigger object (resonance of liver in back ground) are ignored and the objects connected to boundary are removed from the image to further ease the operation and implementation. This is done by rule “Detect organ”.

```

STARTRULE "Detect organ"
IF      InImage ← ObjectDetected (image)
        /* Detect if there is an organ detected*/
THEN   SaveThreshold ← CurrentThreshold(image)
        /* save the current value of threshold */
        CurrentThreshold <= CurrentThreshold + increment
        Until Inimage ← ObjectDetected (image) = FALSE
ENDRULE

```

Now, take select the best of set to get the sharpest boundary and organ outline this is done by the rule “Select Threshold With Minimal Loss”.

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2) \quad (1)$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2 \quad (2)$$

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} (a(n_1, n_2) - \mu) \quad (3)$$

```

STARTRULE "Select Threshold With Minimal Loss"
IF      ImageConstraints = Sharpboundary.CurrentThreshold(Image)
        /* To detect image from set and select sharpest image*/
THEN   SaveThreshold <= CurrentThreshold
        Image <= ApplyThreshold(SaveThreshold, Image)
        /* To save the image in original image with the selected threshold */
ENDRULE

```

This leads to intelligent detection of organ.

D) The isolated image is then subjected to conversion into a binary image, and is then inverted to show the object of interest [8]. If there are small holes in the image and the small holes are filled in the skeleton picture. For this rule “Removing Noise and Larger Objects” comes into play.

```

STARTRULE "Removing Noise and Larger Objects"
IF      (Image.ObjectsDetected < ThresholdObjectSize )
THEN   Image <= FillSmallObjects(image)
        /* Fill holes in image which may be resultant of speckles or noise*/
ENDRULE

```

E) Idem for the larger background organs and are ignored by using the following rule “Removing Larger Objects”. This is done to remove background organs like liver etc.

```

STARTRULE "Removing Larger Objects"
IF      (Image.ObjectsDetected > ThresholdObjectSize)
THEN   Image <=IgnoreLargeObjects(Image)
           /*Remove the object that larger than the objects of interest*/
ImageIsolatedOrgan <=IsolateOrgan(image)
           /*Isolate the organ and save it into the image*/
ENDRULE

```

F) After that the boundary is extracted from this image and is plotted on the actual image. In other word, the object detected is superimposed on the original image. This gives the organ separated from the image.

G) Plotting of the boundary on the original image takes place after they have been extracted from the previous image shown above this is also done for the image for reader to identify the organ detected is correct.. This results on the image shown in Fig 4 (b). At this stage, we also measure the wall thickness of the organ by the rule “Detect Wall Malady”.

```

STARTRULE "Detect Wall Malady"
IF      (OrganWallDetected(ImageIsolatedOrgan) >
          HealthyThresholdOrganWallSize)
THEN   Report <=Add.WallResultsToReport(ImageIsolatedOrgan)
ENDRULE

```



(a)



(b)

Fig. 4. The sonogram with wiener filter applied (a) and the boundary superimposed on the enhanced image (b)

In above rule if the wall thickness is within healthy organ range else measured value is saved and compared with KB. The value is later added to the report.

H) Identification of objects inside the region of interest and applies label and metrics to measure the organ shape, by using the formula equation g (4) as given by [17]:

$$\text{Metric} = 4 * \pi * \text{area}/\text{perimeter}^2 \quad (4)$$

Measurements are then calculated showing the extent of abnormality and displayed on the target image by rule “Detect Organ Shape”.

```

STARTRULE "Detect Organ Shape"
IF      (OrganShapeDetected > 0.62)
        /* Detect shape by formula of area divided perimeter */
THEN   Report <= Add.hydroceleResultsToReport(ImageIsolatedOrgan)
        /* Add data to report that normal shape */
ELSE   Report <= Add.NormalShapeResultsToReport(ImageIsolatedOrgan)
        /* Add data to report that abnormal shape or hydrocele*/
ENDRULE

```

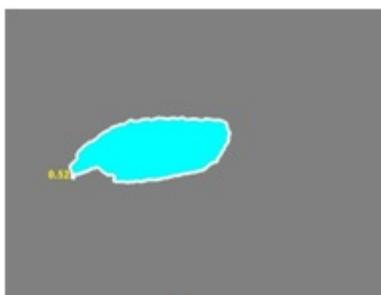
If object with measurement metric values less than 0.62 then it is registered normal i.e. not sufferings from hydrocele else register hydrocele and add to report. This value is selected after testing the results on 32 images of data set of sonograms from different sources. If measured area is healthy compared to Age group this is register healthy else it is registered abnormality and recorded size is added to report. To do this task the rule “Detect Organ Area” is used.

```

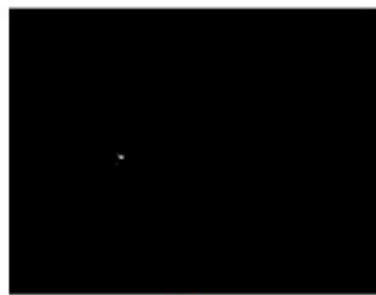
STARTRULE "Detect Organ Area"
IF      (OrganAreaDetected > AgeGroup)
THEN   Report <= Add.MeasureAndargerArea(ImageIsolatedOrgan)
        /* Add data to report that area is larger than age group */
IF      (OrganAreaDetected < AgeGroup)
THEN   Report <= Add.MeasureAndSmallerArea(ImageIsolatedOrgan)
        /* Add data to report that area is smaller than age group */
IF      (OrganAreaDetected = AgeGroup)
THEN   Report <= Add.MeasureAndnormalArea(ImageIsolatedOrgan)
        /* Add data to report that area is normal*/
ENDRULE

```

The abnormality is displayed on the isolated image of the gallbladder.



(a)



(b)

Fig. 5. Metric test to get shape and size characteristics of the organ (a), Stone present inside the organ (b)

- I) If organ registered is more than one then count organs and add to report for duplication or twist by using the rule “Detect Organ Duplication”.

```

STARTRULE "Detect Organ Duplication"
IF (OrganDetectedCount >1)
    /* If there are more organ being registered by the system */
THEN
Report <= Add.CountOrgansAndDuplication(ImageIsolatedOrgan)
    /* Add data to report*/
ENDRULE

```

J) After that subtraction of the two images organ is isolated. Then stone is found using the same techniques as used in above step but with different parameters for boundary and size, If it is inside the boundary of the object [9] register then, Measure the area and add to report. These two images are in which the object was detected and the boundary is drawn on the images and the other was the isolated image which was converted into binary by the rule “Detect Stone”.

```

STARTRULE "Detect Stone"
IF      InImage ←StoneDetected (ImageIsolatedOrgan)
        /* Detect stone in isolated image*/
THEN  Imagestone <=PlotStone(ImageIsolatedOrgan)
        /* superimpose the stone on original image */
        StoneSize <=MeasureStone(Imagestone) /* Measure the stone size */
        Report <= Add. StoneSize
        /* Add data to report giving stone size in pixels*/
ENDRULE

```

K) The resultant image shows the stone inside the area of interest. The report is generated on basis of the facts collected by above stated rules.

Amongst other features is the data basing system with complete patient record and database is available for future reference. A friendly user interface is also provided to further assist the use of our system. This user interface provides simple and easy access to target file on computer. I-Eye provides side by side view of the input image and the enhanced image with diagnosis for an easier comparison. Besides I-Eye produces a formatted patient report and the interactions are recorded into the Data Base for future reference. A detailed integrated assistant is also part of the infrastructure, so further any ambiguity in operations is removed. The system enables medical service to add comments about the case under observation. This concludes the processing with the report enabling user to add their comments or prescription on the last page, with the option to add I-Eye diagnosis. Furthermore, I-Eye can be fine tuned for a specific ultrasound machine this greatly improves the accuracy of the algorithm. The machines with convex probes are suggested for this as the image results are wider and have more useful data in them.

3 Related Work

Specialized open source solution does not exist for the diagnosis of gallbladder diseases. Due to the advent of 3-dimensional technology and the henceforth not much

work has been done towards the 2-dimensional ultrasonography diagnosis[2]. A solution is to use a sonogram at remote location and sent it via Internet to a radiologist specialist, who will analyze the sonogram and later will give the diagnosis about the organ under inspection. Three problems may surge with this approach:

1. Some remote areas exist, where this communication is not possible.
2. This approach may be effective where timing is not of essence but in emergency situations when time is critical, this solution is not effective.
3. Availability of specialists must be maintained all time.

Currently available system for aiding on the decision making process like SonoCalc IMT [4] are available which are specifically designed for aid measurement of thickness of intima-media layer of carotid artery. SonoCalc IMT also generates a report at the end of process. The patient is also registered in the Data Base for future reference. It is able to have a zoom of certain areas of interest after. Although it is not possible to distinguish, locate the organ of interest, or the malady itself. The integrated solutions provided with some diagnostic ultrasound machines provide features like measurement of objects like stones etc but this has to be done manually as the radiologist marks the boundary ends of the objects. Making point to point measurements based on pixels connected by the line which are then scaled back into centimeters or some other metric unit. The selection of points taken from the image is at sole discretion of the radiologist and obvious miscalculations are made when object is not of uniform shape.

Toshiba's Auto-IMT [6] determines the thickness of near and far arterial walls from three segments of the carotid artery: at an optimal angle of incidence and two complementary planes. Auto-IMT traces two complementary planes automatically and calculates the distance between them [6]. Medical specialist is able to determine patient's risk for cardiovascular malady. Nevertheless, the system aids in the diagnosis but active diagnosis is not given yet.

The US PLUSView family of ultrasound products is context Vision's newest and most technologically advanced product for the image enhancement of ultrasound products. There are three members of the family [6]:

- US PLUSView designed for image enhancement, has good speckle reduction and edge and contrast enhancement. Excellent for general radiology uses.
- US PLUSViewADV, advanced version designed to even further reduces speckles and to better harmonize tissue contrast while still presenting very good edge enhancement. This product features adaptive temporal filter with advanced motion compensation.
- US PLUSViewPRE, premium version designed to not only continue to reduce speckle even more but with its directed, focused, enhancement capability. it produces better image of obese patients with enhancement of selected features and depths.

US PLUSView family provides fast, real time performance, and can be run on any quad core i5,7 processor using either Windows XP 7 or Linux operating systems. USPLUSView family provides multi-threaded, multi-core support, supporting maximum performance with extended image refinement capabilities [11].

US PLUSView family is versatile, helps with added accuracy in terms of domain specialization. But it requires high end hardware to meet requirements for operation, and diagnosis problem still persists.

I-Eye stands out as it does not only enhances the image to improve clarity of the image without losing critical information in the sonogram, resulting in an image with lesser noise making it easier for doctors as it takes unwanted additional information and noise off the image. To further boost the accuracy of the estimation of the size of objects detected inside the organ. I-Eye can find more details and measurable metrics like size in terms of length and breadth, as well as shape and area. This information assists not only in the diagnosis, but also for selecting surgical alternatives.

The contrast and brightness of sonogram dependent on different factors like penetration energy, frequency of used signal, and organ under observation. On specific features case like fat present in the area under consideration, once the image is frozen on a machine it cannot be tweaked further to increase readability of image.

On the other hand, I-Eye optimizes the gallbladder diagnosis, such that any image from data set, when input in to the system is intelligently optimized in brightness and contrast to make more information available for the user. This is dynamic and further enhances the diagnosing ability If the user.

There is no provision in most of the machines for sonography to maintain records of a specific or a number of patients. Whereas, I-Eye also maintains a database of the patient and hence comparisons can be made periodically with older observations which provide periodic improvement or degradation in patient's condition.

4 Testing I-Eye

The system was tested on image set of 32 items from different sources varying in patients under consideration and also types of machines with only one constraint that the sonogram should be taken from a convex probe.

Table 1. Mean Authentication of diagnosis from experts

Condition	Number of sonograms considered	Authenticated Diagnostics by Experts Matched With I-Eye
Normal Gall bladder	7	91%
Gall Stone	13	84%
Hydrocele	8	87%
Duplication	4	79%

Table 1 gives the opinion and diagnosis of I-Eye compared with the diagnosis of four radiologists. The average of their opinion is given in favor of the diagnosis are given below, suggesting the high reliability of I-Eye.

For testing the reliability of the system developed, four experts were given the same images one by one. It was requested to give their diagnosis on the sonograms. Then the experts were shown the diagnosis results of I-Eye and their individual opinion for each sonogram compared with I-Eye. The results were gauged numerically in terms of this is

where we are discussing the percentage and then their mean was taken. This result is shown in Table 1. This system is not only versatile and dependable as it achieved a high agreement from radiologists. Also it can be modified for other organs.

So, it is a viable solution for areas which do not have adequate infrastructure or trained medical personnel for diagnosis and also can provide aid to doctors in diagnosis and maintaining the database of patients. Duplication led to least accuracy in diagnosis the cause of which was mainly the orientation of probe not being optimal and adding the image of cyst that was false registered as duplication due to the fact that the energy absorption being so high its registered on sonogram. This problem can easily be tackled with proper handling of the imaging probe. The sonograms should be taken at a particular direction and orientation of probe. Also it cannot handle images with cropped organs as it inhibits it from registering the true shape of organ as one of its limitations. That is the I-Eye needs a sonogram in which the organ is complete. For better results the sonograms of gall bladder are taken with the person lying on their back so the whole organ can be seen. Also a sonogram with side view of gall bladder will cause problems leading to inaccurate measurement of organ size and possibility of missing some information like the number of stones and size of gall bladder itself. These limitations are also applicable for a radiologist if given a side view of the organ would face same problems.

5 Conclusion

I-Eye shows ability to tackle with the inherent noise of sonograms. To improve visibility, I-Eye enhances the contrast and brightness of images, diagnoses maladies like hydrocele, folds, turns or duplication presence of gallstones and also when an organ is healthy. I-Eye maintains a Data Base of patient history, providing standard reporting system for generation of reports with the provided utility. The system is reliable and can be further custom tuned for specific architecture.

I-Eye will be enhanced to diagnose malady like gallstones in further detail identifying the type of stones. I-Eye can also be integrated into machines to perform the same functionality removing the constraint of using a computer too. We also intend to make I-Eye 3-d ready. Some experimentation on slices of 3-d ultrasounds yielded good results especially in gall stone registration. This trend toward programmable ultrasound machines and software will continue.

References

1. Slawomir, B.: Automated Detecting Symptoms of Selected Gallbladder Illness based on Static Ultrasound Image Analysis. In: Bio-algorithms and Med-systems, pp. 35–44 (2006)
2. Mazumdar, B., Mediratta, A., Bhattacharyya, J., Banerjee, S.: A Real Time Speckle Noise Cleaning Filter for Ultrasound Images. In: 19th Int. Symp. on Comp. based Med. Systems, pp. 341–346 (2006)
3. Juang, P.-A., Wu, M.-N.: Ultrasound Speckle Image Process Using Wiener Pseudo-inverse Filtering. In: 23rd IEEE Conf. on Industrial Electronics Society, pp. 2446–2449 (2007)

4. <http://www.sonosite.com/products/sonocalc-IMT/product-tour/>
5. <http://medical.toshiba.com/news/press-releases/2011/02/16/883/>
6. <http://www.contextvision.com/?id=2841>
7. <http://www.mathworks.com/help/toolbox/images/ref/wiener2.html>
8. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice Hall (2001)
9. Singh, E.J., Kaur, N.: Automated segmentation of gallstones in ultrasound images. In: IEEE Int. Conference on Systems, Man, and Cybernetics, vol. 4, p. 2855 (October 1996)
10. Semmlow, J.L.: Biosignal and Biomedical Image Processing MATLAB(2004)
11. York, G., Kim, Y.: Ultrasound Processing and Computing: Review and Future Directions. Annual Review of Biomedical Engg. 1, 559–588 (1999)
12. <http://en.wikipedia.org/wiki/Gallbladder>
13. <http://www.buzzle.com/articles/where-is-the-gallbladder-located-in-the-body.html>
14. Meilstrup, J.W.: Imaging Atlas of the Normal Gallbladder and Its Variants, p. 4. CRC Press, Boca Raton (1994)
15. Myers, R.P., Shaffer, E.A., Beck, P.L.: Gallbladder polyps: epidemiology, natural history and management. Can J. Gastroenterol. 16(3), 187–194 (2002); PMID 11930198
16. Lim, J.S.: Two-Dimensional Signal and Image Processing, p. 548. Prentice Hall, Englewood Cliffs (1990); equations 9.44 – 9.46
17. <http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ipexroundness.html>

Genetic Selection of Fuzzy Model for Acute Leukemia Classification

Alejandro Rosales-Pérez, Carlos A. Reyes-García, Pilar Gómez-Gil,
Jesus A. Gonzalez, and Leopoldo Altamirano

National Institute of Astrophysics, Optics and Electronics (INAOE)
Computer Science Department
Luis E. Erro No.1 Tonantzintla,
Puebla, Mexico
{arosales,kargaxxi,pgomez,jagonzalez,robles}@ccc.inaoep.mx

Abstract. Leukemia is a disease characterized by an abnormal increase of white blood cells. This disease is divided into two types: lymphoblastic and myeloid, each of which is divided in subtypes. Differentiating the type and subtype of acute leukemia is important in order to determine the correct type of treatment to be assigned by the affected person. Diagnostic tests available today, such as those based on cell morphology, have a high error rate. Others, as those based on cytometry or microarray, are expensive. In order to avoid those drawbacks this paper proposes the automatic selection of a fuzzy model for accurate classification of types and subtypes of acute leukemia based on cell morphology. Our experimental results reach up to 93.52% in classification of acute leukemia types, 87.36% in lymphoblastic subtypes and 94.42% in myeloid subtypes. Our results show a significant improvement compared with classifiers which parameters were manually tuned using the same data set. Details of the proposed method, as well as experiments and results are shown.

Keywords: Leukemia classification, selection of fuzzy model, genetic algorithms.

1 Introduction

Leukemia is a disease that arises due to an overproduction of immature white blood cells, which proliferate in the circulatory system and replace normal blood cells [3]. This causes that the body is exposed to potential diseases, due to the lack of defenses.

According to the Leukemia & Lymphoma Society, leukemia is divided into two types: myeloid and lymphoblastic, and each of them can be either acute or chronic [13]. Also, according to FAB¹, acute leukemia is divided in subtypes: L1, L2 and L3 for Acute Lymphoblastic Leukemia (ALL) and M0 to M7 for Acute Myeloid Leukemia (AML) [2].

¹ French-American-British (FAB) Co-operative Group.

Leukemia can be curable if it is diagnosed and treated at the early stages of the disease. The detection of the disease starts with the presence of signs and symptoms and blood count analysis. If the results of the blood counts are abnormal, a study with a microscope is required for identifying abnormalities in white blood cells, this study is based on morphological appearance of the cells. However, the study by optical microscopy shows an error rate of between 30% and 40% (according to information provided by our domain experts), depending on the experience of the person performing the analysis. Other techniques, such as flux cytometer or microarray analysis have good results, but have the drawback of being expensive and can be carried out by very few laboratories [14].

Due to high error rates, incurred by the human expert, and the high costs of other techniques it is necessary to have new accurate and inexpensive techniques. In that sense, few solutions have been proposed. Galindo [5] and Reta [16] classify acute leukemia in types and subtypes based on morphological appearance. However, these works focused mainly on the segmentation and feature extraction tasks. Even though they obtained promising classification results, we believe that these can be improved. In this work, we are interested in acute leukemia recognition, especially in the accurate classification of types and subtypes of acute leukemia based on morphological appearance.

In our work, we propose the automatic selection of a fuzzy classification model for recognizing acute leukemia. However, in recognition of acute leukemia based on morphological appearance of the cell, certain subtypes of leukemia have similar morphological features. The fuzzy model allows knowing the strength with which each cell belongs to each type and subtype of acute leukemia. This additional information is useful to the physician, especially in difficult cases such as those that can belong to more than one subtype of acute leukemia (biphenotypic acute leukemia). Results obtained by our proposed method are compared with results reported in [15] and [16].

The rest of the paper is organized as follows: in section 2 the related works are presented. In section 3 the proposed method is described. Next, in section 4 the experiments and results are shown. Finally, in section 5 the conclusions and future work are presented.

2 Related Work

In order to have a more accurate diagnosis, it is necessary to have more reliable techniques for the recognition of acute leukemia, for this reason many efforts for that task have been performed. In this section, we describe some works related to leukemia classification.

Golub et al. [7] propose leukemia classification based on gene expression using DNA microarrays. The data set has 38 bone marrow samples (27 for ALL and 11 for AML) for training and 34 samples (20 for ALL and 14 for AML) for testing. This work describes cancer classification as two challenges: class discovery and class prediction; they use Self Organizing Maps to discriminate between ALL and AML.

The works [18,20,9,11] are inspired in the work of Golub et al. using the same data set. Su et al. [18] perform several experiments using the original data set which is transformed by means of the magnitude and phase of Fourier transform. They also employ modular neural networks to classify two types of acute leukemia and the best reported result is 75% of correct classification. Xu et al. [20] adopted the ellipsoid ARTMAP to classify between ALL and AML; they generated seven subsets of genes (3000, 1000, 500, 100, 50, 10, 5) and the best result is 97.1% of accuracy. Huang et al. [9] perform a series of experiments using different feature selection methods and a probabilistic neural network is used as a classifier; 95.4% of accuracy is the best reported result. Kanth and Giridhar [11] performed a gene selection through a correlation analysis and Wilcoxon Rank-Sum Test. After that, they proposed a modification of a fuzzy hypersphere neural network for the classification of acute leukemia. They reported a classification accuracy of 100%.

Adjouadi et al. [1] use data collected with a flux cytometer and a neural network is implemented for the recognition of acute leukemia, which is designed to recognize between normal cells or abnormal cells (ALL and AML). The data set has 220 samples (160 are normal cells and 60 are abnormal cells). Reported results reached up to 98.46% for ALL and 97.27% for AML.

In another approach, in Reta et al. [15] types and subtypes of acute leukemia recognition is performed based on morphological appearances. Their data set has 633 samples for types (295 for ALL and 338 for AML) and 435 for subtypes of acute leukemia. They propose to use morphological appearance to recognize acute leukemia types and subtypes. They also propose a segmentation (and feature extraction) technique that differentiates the nucleus and the cytoplasm of the cells. Their reported results reach up to 92.20% accuracy on acute leukemia types classification.

In the works described it is important to note that although the microarray analysis and flux cytometry techniques are very accurate, they have the inconvenience of being very expensive, which is a problem in developing countries such as Mexico. Moreover, and more important, they do not consider subtype classification. On the other hand, work based on the morphological appearance perform a hard classification. In addition to this, the works described above have been manually designed. This motivates us to explore the recognition of acute leukemia based on morphological appearance. We propose the automatic selection of a fuzzy model. The fuzzy model allows us to have additional information of the strength of membership of each cell to types and subtypes of acute leukemia.

3 Genetic Selection of a Fuzzy Model

Genetic algorithms [6] are part of the evolutionary computation area. Evolutionary computation is inspired in the evolutionary theory and it tries to solve problems using computational models of evolutionary processes, such as

selection, reproduction, mutation, survival of the fittest, etc. Generally, a genetic algorithm has five basic components: an encoding scheme that represents the potential solutions for the problem in the form of chromosomes or individuals, a form to create potential initial solutions, a fitness function to measure how close a chromosome is to the desired solution, operators for selection and operators for reproduction [4].

The process of genetic selection of a fuzzy model is shown in Figure 1. Data set is the input data represented in an array of $M \times N$, where M is the number of samples and N is the number of features plus an array of classes. First, the most relevant features in data set are selected using a feature selection method. Then, the subset of selected features are converted to fuzzy values, this is the fuzzification step. Next, the parameters of fuzzy membership are fitted to reduce the overlap degree. Finally, with the fuzzy features a fuzzy classifier is built. Additionally, a genetic algorithm is used to find the combination of each block that minimized the classification error. We explain how each of these components were designed in this system, in order to select a fuzzy model for the classification task.

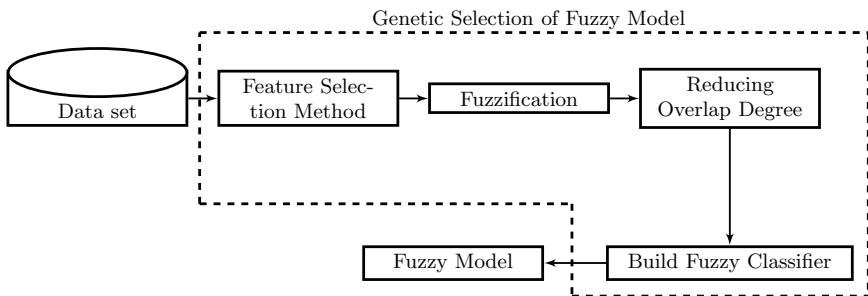


Fig. 1. Genetic selection process of a fuzzy model

3.1 Coding Scheme

In this work, we use a binary representation of the chromosome for our fuzzy model. This representation fits well the needs of our problem because according to Holland [8], it allows us to have more schemes than other representations.

Given a pool of feature selection methods, learning algorithms, and fuzzy membership functions, the problem consists of finding the combination of them that minimizes the classification error. The coding scheme adopted is described in detail next. The feature selection method is represented by two bits: a value of $[0, 0]$ means information gain; $[0, 1]$ represents feature selection based on correlation, a value of $[1, 0]$ means χ^2 statistic and $[1, 1]$ represents Relief.

The type of fuzzy membership function is represented by two bits: $[0, 0]$ means triangular, $[0, 1]$ trapezoidal, $[1, 0]$ gaussian and $[1, 1]$ bell. The number of

linguistic properties is represented by two bits: [0, 0] represents three linguistic properties (low, medium and high), [0, 1] five linguistic properties (very low, low, medium, high and very high) and [1, 0] seven linguistic properties (very low, low, more or less low, medium, more or less high, high, very high).

Finally, the learning algorithms are represented by a variable number of bits, according to the parameters required by each of them. This means that the proposed genetic algorithm works with individuals of different size. The learning algorithms considered in this work are: Fuzzy Decision Tree (FDT) [19], Fuzzy Decision Forest (FDF) [10], Fuzzy Relational Neural Network (FRNN) [17] and Fuzzy K-Nearest Neighbor (FKNN) [12]. The coding of the parameters are shown in Tables 1 and 2. Table 1 shows the parameters encoded with two bits and table 2 shows the parameters encoded with three bits associated with each learning algorithm.

Table 1. Two-bit encoding for the parameters of the learning algorithms

Parameter	Encoding bits			
	[0, 0]	[0, 1]	[1, 0]	[1, 1]
FRNN				
Relational product	Max-min	Square	Sub-triangle	Super-triangle
FKNN				
Similarity measure	City-block	Euclidean	Chord	Correlation

Table 2. Three-bit encoding for the parameters of the learning algorithms

Parameter	Encoding bits							
	[0, 0, 0]	[0, 0, 1]	[0, 1, 0]	[0, 1, 1]	[1, 0, 0]	[1, 0, 1]	[1, 1, 0]	[1, 1, 1]
FDT								
Confidence	1.00	0.95	0.90	0.85	0.80	0.78	0.75	0.70
FDF								
Confidence	1.00	0.95	0.90	0.85	0.80	0.78	0.75	0.70
Trees	3	5	7	8	9	10	12	15
FRNN								
Epochs	2	5	10	15	20	25	30	50
Learning rate	0.10	0.15	0.20	0.25	0.30	0.33	0.40	0.50
Bias	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.50
Weights	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80
FKNN								
Neighbors	1	2	3	4	5	7	8	10

3.2 Reducing Overlap

In order to deal with uncertainty in the classification we propose to use fuzzy logic to transform input data to fuzzy values using membership functions, which

will be determined by the chromosome described in section 3.1. Additionally, the parameters of fuzzy membership functions are adjusted to fit the problem. A second genetic algorithm is used for that task.

In this case, each membership function is represented by four parameters (a, b, c, d) and every parameter is represented by four bits and its codification is shown in Table 3. For that reason, the updating parameters for the membership functions are represented by a chromosome of $16 \times numLP \times n$ bits, where $numLP$ is the number of linguistic properties and n the number of selected features. In Table 3 each value of the gene is associated with a value between 0.50 and 1.50 which represents a scaling value (SV) to increase or decrease appropriately the parameters of fuzzy membership functions.

Table 3. Binary codification for optimizing parameters of the membership function

Gene	SV	Gene	SV	Gene	SV	Gene	SV
[0, 0, 0, 0]	0.50	[0, 1, 0, 0]	0.75	[1, 0, 0, 0]	1.20	[1, 1, 0, 0]	1.40
[0, 0, 0, 1]	0.55	[0, 1, 0, 1]	0.80	[1, 0, 0, 1]	1.25	[1, 1, 0, 1]	1.45
[0, 0, 1, 0]	0.60	[0, 1, 1, 0]	0.90	[1, 0, 1, 0]	1.30	[1, 1, 1, 0]	1.50
[0, 0, 1, 1]	0.70	[0, 1, 1, 1]	1.10	[1, 0, 1, 1]	1.35	[1, 1, 1, 1]	1.00

As a fitness function for this genetic algorithm we use a metric of separability of classes, expressed as:

$$overlap = \frac{\sum_{i=1}^n \delta(x_1, N_1(x_i))}{\sum_{i=1}^n \delta(x_1, N_2(x_i))} \quad (1)$$

where $N_1(x_i)$ and $N_2(x_i)$ are, respectively, the intra-class and inter-class nearestneighbor of x_i , δ is a distance function and in this case is the Euclidean distance.

3.3 Genetic Operations

The selection operation is done by tournament; more information on this operator is detailed in [4]. The selection operator chooses two individuals (chromosomes) for the crossover operation, but in order to ensure diversity in the population, the Hamming distance between two individuals is computed and if it is greater than a threshold β the individuals are selected, otherwise the process is repeated.

Crossover is performed as follows: the selection operator chooses two individuals to be crossed, subsequently, the bits that are different in both individuals are extracted and the cross is performed by one-point crossover over them. A one-point crossover operator randomly selects a crossover point in the two previously selected parents, and the bitstrings after that point are swapped between the two of them.

In addition, we use a mutation operator with the aim to introduce new genetic material into an existing individual. In mutation, the bits that are equal are considered and uniform mutation is applied on them.

3.4 Avoiding Extinction

The goal of genetic operations is to create new individuals from existing ones. The selection operation, which is one of the genetic operations, selects the parents to be crossed, giving preference to the fittest individuals. However, individuals who represent a type of learning algorithm may be less selected, due to its low fitness value, than others causing the number of these individuals to become smaller and disappear in future generations.

When the size of the population of individuals that represent a type of learning algorithm is lower than a threshold α , the surviving individuals for this type are copied, mutated and subsequently added to the population. This step is performed in order to avoid the extinction of these individuals, as well as to maintain the diversity of the different models. In order to ensure that the population size remains constant, bad performing individuals of other types are removed.

3.5 Fitness Function

Fitness function is an important part for a genetic algorithm because it is in charge of evaluating the potential solutions.

To estimate the error we use the balanced error rate (BER). This metric is a useful error measure, mainly on imbalanced data sets. The expression for BER is:

$$BER = \frac{1}{l} \sum_{i=1}^l e_i \quad (2)$$

where BER is the balanced error rate, l is the number of classes in the problem and e_i is the missclassification rates for the i^{th} class.

The fitness function is defined as the average of BER in 2-fold cross validation.

4 Experiments and Results

For experiments we used the leukemia data set from the Laboratory of Specialties at the IMSS Puebla. This data set has 633 examples of images of bone marrow cells and is described in Table 4. The images were collected by [14] using a digital camera connected to a Carl Zeiss optical microscope with a 100x objective. All images have a resolution of 800x600 pixels with 24 bits. Images of bone marrow were segmented and features from the whole cell as well as the nucleus and the cytoplasm (N&Ci) were extracted from them. Geometrical, statistical and texture as well as principal component features were extracted from both whole cell and the nucellus and cytoplasm by Reta [16]. The data sets were divided, on one side, with features extracted from the whole cell and on the other

Table 4. Description of the leukemia data set

ALL		AML	
Subtype	Number	Subtype	Number
L1	102	M2	95
L2	135	M3	47
		M5	56
Other	58	Other	140
Total	295	Total	338

with features extracted from the N&Ci. For each of these three different feature sets we used: first, geometric, statistical and texture features (A); second, principal components features (B); third, geometric, statistical, texture, and principal component features (C).

We performed several experiments considering binary classification and multiclass classification on the acute leukemia classification data set. In each experiment we considered different classification tasks according to [16]. Firstly, we consider acute leukemia type classification; next, subtype leukemia classification tasks for type ALL (L1 and L2) and AML (M2, M3 and M5) as a binary problems. Finally, the multiclass problem is treated in two different ways: first, we classify all AML subtypes (M2 vs M3 vs M5); second, all acute leukemia subtypes (L1 vs L2 vs M2 vs M3 vs M5).

The genetic algorithm parameters used are: population size is 24, initially 6 individuals for each learning algorithm, the threshold α is set to 3, threshold β is set to 4, the number of generations is fixed to 50, we used a crossover rate of 0.8 and 0.05 as the mutation rate.

The evaluation was done using 10 fold cross validation. 10 fold cross validation divides the data set into 10 disjoint subsets, and in each fold a subset is used for testing and the remaining subsets for training. This process is repeated until all subsets have been used for testing and training. As evaluation metrics we are using accuracy (ACC) and area under the ROC² curve (AUC) for all experiments; additionally, in binary problems sensibility (TPR) and specificity (TNR) are used.

Table 5 shows the resulting models for each case and it describes the feature selection method (FSM), type of membership function (TMF), number of linguistic properties (NLP) and the fuzzy classifier (FC) chosen for each specific data set by the genetic algorithm. Previous results obtained for each experiment were reported in [15,16] and those obtained by our proposed method are presented in table 6. In this table we can see that the results are improved for most data sets, especially in data sets with features extracted from the whole cell.

² Receiver Operating Characteristic.

Table 5. Resulting classification model for each data set. See text for explanation of nomenclature

ID	FSM	TMF	NLP	FC
1	information gain	gaussian	7	FDT
2	correlation	bell	5	FRNN
3	relief	triangle	5	FKNN
4	relief	trapezoid	3	FKNN
5	χ^2	gaussian	5	FDT
6	relief	triangle	3	FKNN
7	correlation	bell	5	FDF
8	relief	gaussian	7	FDT
9	correlation	trapezoid	5	FKNN
10	relief	bell	7	FDF
11	correlation	gaussian	5	FRNN
12	χ^2	gaussian	5	FDF
13	correlation	trapezoid	3	FKNN
14	information gain	gaussian	7	FDT
15	correlation	trapezoid	5	FKNN
16	relief	bell	7	FDF
17	information gain	gaussian	3	FDT
18	χ^2	gaussian	5	FDF
19	information gain	gaussian	3	FDT
20	information gain	gaussian	7	FRNN
21	relief	triangle	5	FKNN
22	correlation	bell	3	FDF
23	information gain	bell	7	FDT
24	information gain	gaussian	5	FDF
25	relief	triangle	3	FKNN
26	information gain	gaussian	7	FDT
27	relief	bell	5	FDF
28	χ^2	gaussian	5	FDF
29	information gain	gaussian	7	FDF
30	correlation	bell	3	FDF
31	correlation	trapezoid	5	FKNN
32	correlation	gaussian	7	FDF
33	relief	triangle	3	FKNN
34	correlation	gaussian	5	FDT

Table 6. Classification results for each experiment. Results are the average using 10 fold cross validation for each metric for the different types and subtypes of acute leukemia. The best result of each case is shown in bold.

ID	Region	Data set	Accuracy		TPR		TNR		AUC	
			Features	[15,16]	Proposal	[15,16]	Proposal	[15,16]	Proposal	[15,16]
ALL vs AML										
1		A	82.68	88.78	81.70	89.88	83.89	86.75	89.26	93.90
2	Cell	B	73.87	76.62	78.88	78.31	67.68	74.15	79.06	83.06
3		C	81.32	88.78	82.17	84.75	80.32	92.31	89.92	91.68
4		A	81.32	90.52	82.17	92.31	81.43	88.47	91.68	95.38
5	N&Ci	B	74.95	78.99	77.97	77.97	71.26	79.89	81.27	83.27
6		C	92.20	93.52	91.97	91.52	92.37	95.27	92.17	96.13
L1vsL2										
7		A	76.78	87.36	66.64	82.45	84.45	91.26	81.40	91.89
8	Cell	B	72.59	75.20	65.64	65.73	77.80	82.36	76.08	76.69
9		C	75.92	84.84	65.55	84.27	83.79	85.27	82.25	88.34
10		A	84.40	86.89	83.45	85.27	85.27	87.97	90.69	91.92
11	N&Ci	B	77.63	79.30	66.82	80.36	85.88	78.52	83.67	81.51
12		C	84.04	87.00	78.55	85.27	88.41	88.35	88.61	91.03
M2vsM3M5										
13		A	74.24	92.41	70.56	92.67	77.55	92.18	80.45	96.22
14	Cell	B	64.21	73.80	56.00	71.22	71.82	76.00	71.06	78.89
15		C	75.32	90.43	71.56	90.44	78.73	90.36	84.12	92.46
16		A	92.45	89.90	88.33	88.22	96.18	91.36	95.90	92.50
17	N&Ci	B	73.13	74.50	79.11	69.00	68.09	79.82	78.93	79.37
18		C	91.95	91.56	90.33	88.67	93.18	94.27	94.68	94.83
M3vsM2M5										
19		A	80.79	93.47	39.00	85.00	94.00	96.00	78.82	96.66
20	Cell	B	81.37	81.38	39.00	51.00	94.79	90.71	78.67	70.79
21		C	77.09	90.02	23.30	78.00	93.77	94.00	71.01	89.03
22		A	91.89	94.42	80.45	85.50	95.50	97.33	87.97	91.42
23	N&Ci	B	79.21	80.83	48.00	46.50	88.79	92.00	73.91	74.46
24		C	89.36	87.63	64.35	68.50	97.15	93.33	89.85	91.90
M5vsM2M3										
25		A	84.37	92.41	73.00	91.00	88.95	93.00	86.64	95.40
26	Cell	B	76.79	71.18	38.67	44.33	91.71	81.67	73.14	66.02
27		C	83.34	93.47	49.67	93.33	96.62	93.67	84.98	93.50
28		A	91.89	94.37	87.00	89.00	93.76	96.43	95.52	96.71
29	N&Ci	B	77.82	72.84	28.67	54.33	97.14	80.29	73.32	70.60
30		C	90.89	92.41	74.33	90.67	97.24	92.95	93.54	95.52
M2vsM3vsM5										
31	Cell	C	66.13	81.65	NA	NA	NA	NA	78.66	90.92
32	N&Ci	C	84.87	86.65	NA	NA	NA	NA	92.80	93.23
L1vsL2vsM2vsM3vsM5										
33	Cell	C	55.86	66.56	NA	NA	NA	NA	84.03	85.05
34	N&Ci	C	77.48	67.62	NA	NA	NA	NA	92.33	83.50
Average			79.95	84.63	67.43	79.18	86.58	88.53	84.31	87.76

5 Conclusion and Future Work

In order to properly recognize the type and subtype of acute leukemia for a given cell we have proposed the use of genetic algorithms for the selection of a complete fuzzy classification model. The proposed method can automatically select a model to fit the problem. Besides, the use of fuzzy set theory allows the physician to know the strength with which each cell belongs to each type and subtype of leukemia. This information is useful to them in order to identify biphenotypic acute leukemias (especially in cells that have features of different leukemia subtypes) and to determine the type of treatment of the patient.

The results obtained with this method show an improvement, in average, of 4.68% accuracy in classification. In addition, our method has the advantage of automatically selecting the fuzzy model for classifying samples of acute leukemia and it allows too to set the right parameters for an appropriate classification model for each case.

Our future work will focus on testing other alternatives to improve performance, such as exploring ensembles of classifiers. Other strategies such as selection of the fitness function and hierarchical classification will be explored. Additionally, we are going to test our proposed method with data sets from other domains.

Acknowledgements. Alejandro Rosales-Pérez (student grant No. 335690) thanks to CONACyT for the financial support given to his postgraduate studies.

References

1. Adjouadi, M., Ayala, M., Cabrerizo, M., Zong, N., Lizarraga, G., Rossman, M.: Classification of leukemia blood samples using neural networks. *Annals of Biomedical Engineering* 38, 1473–1482 (2010), <http://dx.doi.org/10.1007/s10439-009-9866-z>, doi:10.1007/s10439-009-9866-z
2. Bennett, J.M., Catovsky, D., Daniel, M.T., Flandrin, G., Galton, D.A.G., Gralnick, H.R., Sultan, C.: Proposals for the classification of the acute leukaemias french-american-british (fab) co-operative group. *British Journal of Haematology* 33(4), 451–458 (1976), <http://dx.doi.org/10.1111/j.1365-2141.1976.tb03563.x>
3. Bozzone, D.M.: The biology of cancer: Leukemia. Chelsea House Pub. (2009)
4. Engelbrecht, A.: Computational intelligence: an introduction, 2nd edn. Wiley (2007)
5. Galindo, M.C.: Obtención de características de subtipos de leucemia en imágenes digitales de células sanguíneas para su clasificación. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica (2008)
6. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning, 1st edn. Addison-Wesley Professional (January 1989), <http://www.worldcat.org/isbn/0201157675>

7. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286(5439), 531–537 (1999),
<http://www.sciencemag.org/content/286/5439/531.abstract>
8. Holland, J.: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor (1975)
9. Huang, C., Liao, W.: A comparative study of feature selection methods for probabilistic neural networks in cancer classification (2003)
10. Janikow, C., Faifer, M.: Fuzzy decision forest. In: 19th International Conference of the North American Fuzzy Information Processing Society, NAFIPS 2000, pp. 218–221. IEEE (2000)
11. Kanth, B., Giridhar, B.: Gene expression based acute leukemia cancer classification: A neuro-fuzzy approach. *International Journal of Biometrics and Bioinformatics (IJBB)* 4(4), 136 (2010)
12. Keller, J.M., Gray, M.R., Givens Jr., J.A.: A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics* 15(4), 580–585 (1985)
13. Leukemia & Lymphoma Society:
<http://www.lls.org/#/diseaseinformation/leukemia/>
14. Morales, B.A.: Extracción de características en imágenes de células de médula ósea para la clasificación de leucemias agudas. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica (2007)
15. Reta, C., Altamirano, L., Gonzalez, J., Diaz, R., Guichard, J.: Segmentation of bone marrow cell images for morphological classification of acute leukemia. In: Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference, FLAIRS 2010 (2010)
16. Reta, C.: Segmentación y clasificación de células con leucemia a partir de información contextual en imágenes digitales. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica (2009)
17. Reyes, C.: On the design of a fuzzy relational neural network for automatic speech recognition. Ph.D. thesis, Doctoral Dissertation, The Florida State University, Tallahassee, Fl (1994)
18. Su, M., Basu, M., Toure, A.: Multi-domain gating network for classification of cancer cells using gene expression data. In: Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN 2002, vol. 1, pp. 286–289. IEEE (2002)
19. Wang, H.: Intelligent Data Analysis: Developing New Methodologies Through Pattern Discovery and Recovery. IGI Global (2008)
20. Xu, R., Anagnostopoulos, G., Wunsch, D., et al.: Tissue classification through analysis of gene expression data using a new family of art architectures. In: Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN 2002, vol. 1, pp. 300–304. IEEE (2002)

An Ontology for Computer-Based Decision Support in Rehabilitation

Laia Subirats and Luigi Ceccaroni

Barcelona Digital Technology Centre
Roc Boronat 177, 5th floor, 08018 Barcelona, Spain
lsubirats,lceccaroni}@bdigital.org

Abstract. Although functionality and disease classifications are available thanks to initiatives such as the “international classification of functioning, disability and health”, the “systematized nomenclature of medicine - clinical terms” and the “international classification of diseases”, a formal model of rehabilitation interventions has not been defined yet. This model can have a fundamental role in the design of computer-based decision support in rehabilitation. Some initiatives such as the “international classification of health interventions” are in development, but their scope is overly general to cope with the specificities that characterize rehabilitation. The aim of this work is to represent knowledge in order to carry out diagnosis and personalization of activities in cases of people with functional diversity. To define the diagnosis and activity personalization, a methodology has been developed to extract standardized concepts from clinical scales and the literature.

Keywords: knowledge representation, rehabilitation, functional diversity, personalized medicine, evidence-based medicine.

1 Introduction

Until recently, progress in health and rehabilitation was hit-or-miss. We would find something without having a good understanding of how it worked. *Oh, here is an activity that improves quality of life. We have no idea why it works.* We would discover activities to perform desirable functions, often with many severe side effects, but we lacked the means to *design* medical interventions for a carefully targeted purpose. Such a random and irregular approach to medical discovery is typical. But now this situation is changing, and very rapidly. We have moved from the old paradigm, in which the progress in health and rehabilitation has been unpredictable, to a new era in which healthcare has now become an information technology.

Unfortunately, most healthcare practitioners actuate according to the old paradigm and still do not practice rehabilitation as an information technology; do not make maximal use of the latest rehabilitation knowledge that is already available today; and do not take full advantage of the available information and of simulation capabilities. We have the means of simulating biology, physiology and interventions on computers so that we can try out new rehabilitation interventions and drugs on simulators, a process dramatically faster than human testing. The point is that health and

rehabilitation are now information technologies, and that represents a new frontier. As a result, our health technologies are subject to what is sometimes called the law of accelerating returns, an exponential improvement in the ability to understand, model, simulate and reprogram the information processes underlying disease and functioning.

Yet, the way rehabilitation is currently being done is not entirely satisfactory. Therapies are based on therapists' experience and not necessarily on unbiased knowledge accumulated through exhaustive studies of a wide range of cases. This leads to frequent errors, which slow down recuperation, harm people and increase costs.

Research has been carried out to study the evolution of people undergoing neuropsychological rehabilitation. Nevertheless, in spite of their importance, psychological, social and environmental factors have been scarcely taken into account and, furthermore, a deep study of how interventions influence prognosis has not been done. The purpose of the work proposed here is to contribute to fill this gap. In particular, the objective of this work is to represent knowledge used in clinical decision support systems and study how interventions as well as the psychological, environmental and social issues influence the prognosis and the quality of life of people with functional diversity. This is done using several *artificial intelligence* (AI) techniques, with which knowledge will be used to build a decision support system in the field of rehabilitation and tools for prognosis assessment.

1.1 Computer-Based Decision Support

Computer-based clinical decision support is an area between *health informatics* and AI. *Clinical decision support systems* (CDSSs) have a fundamental role in improving people safety and healthcare quality and efficiency (and their design and theoretical foundations are object of much research) when clinicians:

- deal with complex cases;
- are prone to making errors;
- cannot keep up with the ever increasing medical knowledge;
- deal with large numbers of routine decisions.

CDSSs are computer systems designed to impact clinician decision making (e.g., prognosis) about individual persons before, during or after (but ideally at the point in time in which) these decisions are made [1]. With the increased focus on the prevention of medical errors, *computer-based physician order entry* (CPOE) systems and CDSSs have been proposed as a key element in improving people safety [2].

There is a variety of systems that can potentially support clinical decisions. Decision support systems have been incorporated in healthcare information systems for a long time, but these systems usually have supported retrospective analyses of financial and administrative data. Recently, sophisticated data mining approaches have been proposed for similar retrospective analyses of both administrative and clinical data [3]. Although these retrospective approaches can be used to develop guidelines, critical pathways or protocols to guide decision making at the point of care, such retrospective analyses are not usually considered to be CDSSs. Perreault and Metzger [4] have described CDSSs using several dimensions. According to their framework, CDSSs differ in:

- the timing at which they provide support (before, during or after the clinical decision is made);
- how active or passive the support is, that is, whether the CDSS actively provides alerts and knowledge or passively responds to physician input or patient-specific information;
- how easy they are for clinicians to access.

Also, in principle, there are two types of clinical decisions:

- related to the diagnosis, in which computers may assist in diagnosing a disease on the basis of available patient data;
- related to the therapy, in which the best next test or therapy is determined on the basis of evidence or other knowledge.

Once large collections of patient data (e.g., patient history data, laboratory data, drug data, and patient outcomes) are available, new relations among data may be found. This will give rise to new insight and new decision rules, to be implemented in CDSSs. Most often, however, the rules to be implemented in a CDSS are derived from clinical evidence, i.e., from the medical literature and clinical experience.

CDSSs generally use one of the following paradigms to provide support: workflow-driven, production-rule-based, and predictive analytics. Some predictive-analytics systems use case-based reasoning (CBR) [5] to provide support in several stages of medical professionals' interventions [6]: diagnosis, treatment procedure, daily life management for people with chronic or degenerative conditions [7], [8], prognosis to reduce possible risks [9], and people's classification. This paper focuses on issues in clinical vocabularies, user modeling and reasoning.

1.2 Role of Ontologies in Clinical Decision Support System

In rehabilitation and related medical domains there is a lack of formalization and decisions are usually based on therapists' experience and not necessarily on evidence. Furthermore, there is a lack of interoperability among knowledge-management systems. The aim of the introduction of ontologies is to automatically share and reuse knowledge when building CDSSs. This will facilitate the adaptation of knowledge to provide evidence-based and personalized therapies, and the evaluation of the effectiveness of a treatment.

Interoperability is very important to be able to share and reuse knowledge. To provide interoperability, both the syntax of message exchange and the semantics of concepts (e.g., person's profile, scientific evidence, diagnosis, treatment, disease progression, and natural history) should be standardized. In the domain of healthcare services, to identify clinical activities and concepts, nomenclatures exist, such as: the *systematized nomenclature of medicine - clinical terms* (SNOMED CT), the *international classification of functioning, disability and health* (ICF), the *international classification of diseases* (ICD), the *international classification of health interventions* (ICHI), or the *diagnosis-related groups* (DRG) that allow classifying persons using few variables. As shown in Figure 1, mapping *electronic health records* (EHR), or *patient health records* (PHR), to terminologies and classifications helps to share and reuse knowledge in the domains of *population health, clinical environment, administration and report presentation*.

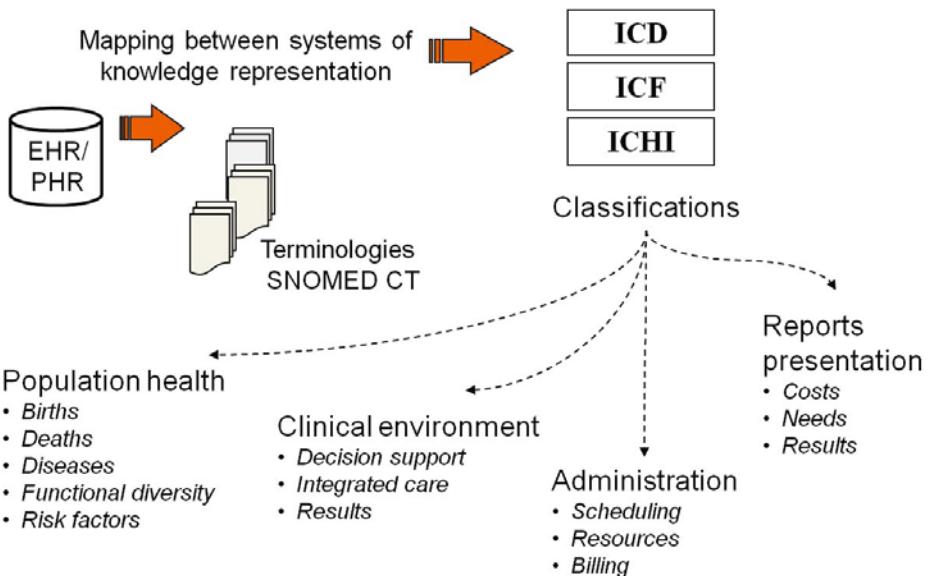


Fig. 1. Role of classifications and terminologies in healthcare services

If ontologies are used for knowledge representation, the relationships in them (see sections 2.2, 3.2 and 3.3) can be exploited to automatically infer implicit knowledge. This kind of inference can be performed by reasoners such as Pellet [10]. To evaluate the consistency and check the validity of data, other ontologies can be used, such as the ICF [11], which specifies constraints for the values of concepts (classes, properties and instances). Queries and assignments in ontologies can be performed by means of *SPARQL protocol and RDF query language* (SPARQL) and Jena [12].

2 Methodologies to Represent Knowledge

Methodologies to encode to international standards, reference models, guidelines and patterns have been designed to represent knowledge. This work is based on Beale and Heard, who provide the reference model for the management of health information called *openEHR* [13]; and also on Cieza et al. [14], who provide a methodology to encode to the ICF standard. Regarding methodologies, Cimino [15] provides some guidelines for the design and use of medical ontologies, and differentiation between concept and context; while Sowa [16] offers best practices for building bio-ontologies and provides ontology design patterns (ODP).

To represent the status of a person in rehabilitation, indicators can be used that characterize the person's body, the rehabilitation processes and as many pathologies as possible. An *indicator* is a parameter or descriptor used to measure or compare *processes*, *results* (obtained in the execution of a rehabilitation task or activity), *body functions*, *body structures*, *activities and participation* and *environment factors*. Body functions, body structures, activities and participation and environmental factors are ICF's branches. A *process indicator* is used to assess whether a task is being performed correctly; and a *result indicator* is used to assess the performance in carrying out an activity or whether

the objectives of the activity have been achieved. These indicators can be extracted from clinical scales and then encoded into international standards. If it is necessary to combine indicators and their values, a methodology might be needed to carry out this combination.

Indicators can be grouped into *core sets* to facilitate daily practice. Core sets can be formed according to functionality, pathology or rehabilitation process. (We consider four classes of rehabilitation processes: cognitive [228553007], functional [229594008], respiratory [108228000] and cardiac [313395003]). Core sets are useful because, in daily practice, clinicians and other professionals need only a fraction of the categories found in ontologies such as ICF and SNOMED CT. Several core sets already exist of different pathologies, such as multiple sclerosis, spinal cord injury or traumatic brain injury [17], however, finding the core categories for rehabilitation processes and moving from a pathology-based approach to one based on functionality and rehabilitation is needed.

A methodology to extract standard-based indicators from existing scales and parameters would include at least the following elements: a search for scales and parameters of the mentioned rehabilitation types (cognitive, functional, respiratory and cardiac rehabilitation); a prioritization and selection of scales and parameters based on literature, and coverage of rehabilitation processes and indicator types; an aggregation of indicators according to the process taxonomy.

2.1 Methodology to Encode Indicators into International Standards

To encode indicators into international standards, ICF is considered first because its domain is closer to the one of rehabilitation and, if no category is found to define a concept, SNOMED CT is considered, which is less specific and includes top-level categories. The methodology to encode into ICF can be found in Cieza et al. [14], [18], while the methodology to translate to SNOMED CT is as follows:

1. Using any search-capable SNOMED CT interface (e.g., [19]), search for the concept that you want to encode.

Example: “Infiltration of local anesthetic and corticoid”.

2. If there is no exact match, search for a synonym.

Example: “Infiltration of local anesthetic and corticosteroid”.

3. If there are no synonyms, use a combination of hypernyms and hyponyms to find concepts that are modeled in SNOMED CT.

Examples: “Skin infiltration of local anesthetic and steroid”, “Intramuscular infiltration of local anesthetic and steroid”, “Infiltration of local anesthetic and steroid to subcutaneous tissue”.

4. Check if the type of the concepts found in SNOMED CT properly models the concept to be encoded. There are 19 types of concepts in SNOMED CT, such as clinical finding, physical object, social context, physical force, substance or procedure. In the previous examples, the type should be “procedure” in all cases.

2.2 Methodology to Combine Indicators and Their Values

Each indicator has a type and a value. When encoding parameters into indicators and combining several parameters into one indicator the following methodology is used:

1. Type of value of the indicator: If the indicator is encoded as ICF, its values are the ones specified by the ICF standard (five qualitative, ordered values, plus *not specified* and *not applicable*: 0, 1, 2, 3, 4, 8, 9). If the indicator is not encoded as ICF and there is only one parameter the indicator is derived from, the type of value is preserved. If the indicator is not encoded as ICF and there is more than one parameter the indicator is derived from and the type of these parameters is different, the type of these parameters is previously translated into five qualitative, ordered values, plus *not specified* and *not applicable*.
2. Values of the indicator: Depending on the specific indicator and on the type of its value, the function to aggregate values of several parameters can be the average, the maximum, the minimum or the median. For example, for *blood pressure* (b420)¹ the maximum or the minimum are used as aggregation functions, while for *dressing* (d540) average or median aggregation functions are used.

3 Representation of the Health Status of a Person

For the representation of the health status of a person in rehabilitation, an ontology based on international standards is proposed. The person's information is based on Beale and Heard [20], who provide the reference model for the management of health information called *openEHR* [13]. External ontologies are used to define metadata [21], interventions [22], functionality [11] and diseases [23]. The proposed ontology, encoded in the OWL format [24], is summarized in the following subsections.

3.1 Summary of the Health Information Ontology

The ontology is composed of 77 classes. There are 2 classes in the first level of the hierarchy, 7 classes in the second, 12 classes in the third and 9 in the forth. The maximum depth is 6 and the maximum number of siblings is 9. Furthermore, there are 5 classes with a single subclass, and 8 properties.

3.2 Classes

The classes of the ontology are summarized in Figure 2. *Health information* has three subclasses: *clinical record* (referring to the past of the person), *opinion* (present) and *therapeutic process* (future). The class **clinical record** includes concepts such as:

Demographic_data (*Demographic_history_detail* [302147001])

Observation (*Personal_health_status* [405157008])

- **Health_condition**:
 - **Indicator**: *Body_structures* (s), *Body_functions* (b), *Activities_and_participation* (d), *Process* [415178003] and *Result*
 - **Assessment_scales** [273249006] and **Observation_parameter** [252116004]

¹ ICF and SNOMED CT codes are written in round and square brackets respectively.

- Contextual_factors:
 - Environmental_factors (*e*):
 - Natural_environment_and_human-made_changes_to_environment (*e2*): Air_quality (*e260*), Pollen_concentration [256259004], Humidity (*e2251*) and Temperature (*e2250*)
 - Residential_environment [272497004]: Ambulatory_care_site [35971002]
 - Hospital_AND/OR_institution [108343000]: Hospital [22232009]
 - Products_or_substances_of_personal_consumption (*e110*): Drug_Aerosol [52262001], Drug_dose [398232005], Nebulizer [334947002] and Drugs (*e1101*)
 - Personal_factors: History_of_present_illness_section [422625006] and Traumatic_AND/OR_non-traumatic_injury [417163006] with subclasses Non-traumatic and Traumatic_abnormality [19130008]
- Therapeutic_process Past_history_of_procedure [416940007]: Activity [257733005] and Prescription_of_therapeutic_regimen [55053003]

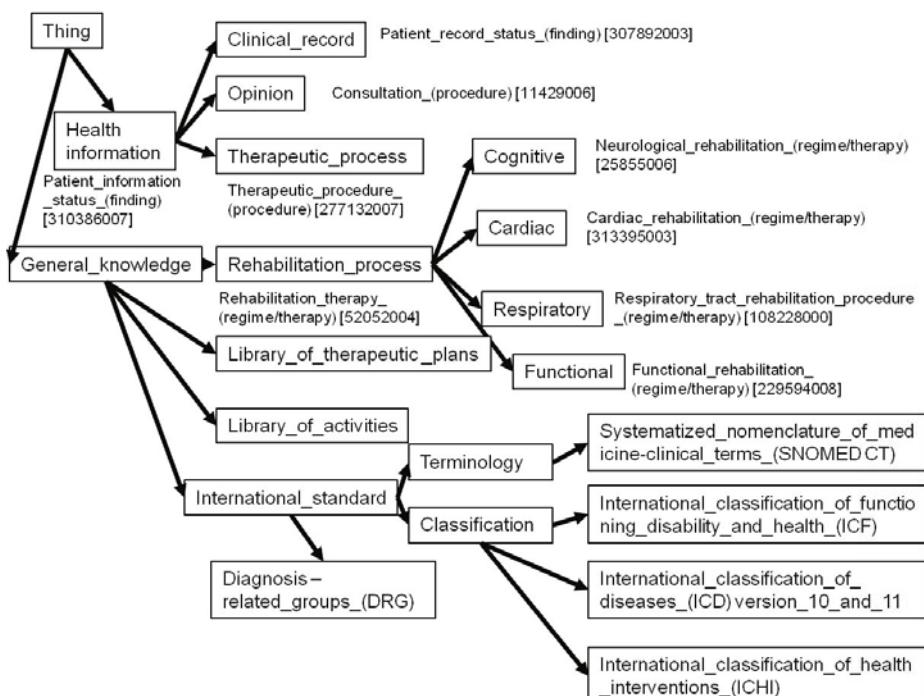


Fig. 2. Class diagram of *health information*

The class **opinion** (*Consultation* [11429006]) includes concepts such as:

Assessment (*Assessment_section* [424836000])

- *Diagnosis* [439401001]: *Indicator*
- *Prognosis/outlook* [170967006]: *Indicator*
- *Risk_factor* [80943009]: *Family risk* (*Familial_risk_factor* [102486008], *Family_history_of_disorder* [281666001] and *Family_history_section* [422432008]) and *Indicator*

Proposal (*Plan_section* [423134005]): *Goal_context* [410518001] and *Recommendation_to* [420227002] (that has *Prescription_of_therapeutic_regimen* [55053003])

Finally, the class **therapeutic process** includes concepts such as *Plan_section* [423134005] and *Prescription* [16076005].

3.3 Properties

Object properties represent relationships between two individuals (classes or instances of classes). Data properties describe relationships between an individual and data values. In the ontology, some properties are semantic relations based on the current proposal of the ICD version 11 [25], for instance:

- Has_disease;
- Has_localization: e.g., an *observation* has as localization a *Body_structure*;
- Is_manifestation_of [417318003]: e.g., a *contextual_factor* has as manifestation a *Health_condition*; a *Health_condition* is manifestation of a *contextual_factor*.

Some object properties are related to activities:

- Has_recipient;
- Has_manager;
- Has_technology;
- Has_process_scale: *Assessment_scales* [273249006] and *Process* [415178003];
- Has_result_scale: *Assessment_scales* [273249006] and *Result_comments* [281296001].

For the main classes, the following properties are defined:

Activity_ [257733005]: *Medical_contraindication_(finding)* [397745006], Recipient, Manager, *Instrument_device_(physical_object)* [57134006], Identifier, Indications, Tasks, Title, Protocol, *Scientific_evidence*, *Procedure_milestone* [397788003], Indicators (equivalent to *Indicator* that is subclass of *opinion*), *Goal_context_(qualifier_value)* [410518001].

Goal_context has the following sub-properties:

- *Activity_of_daily_living* [129025006] (d6)
- Participation: *Finding_related_to_ability_to_perform_community_living_activities_(finding)* [365341008] (d7) (d8)

- Therapeutic: Rehabilitation of *Body_functions* (*b*) and *Body_structures* (*s*)

Demographic_history_detail [302147001]: Surname_ [397678008],
City_of_residence_ [433178008], Carer's_details_ [184140000], Date_of_birth_
[184099003], Patient_sex_[184100006], Patient_name_ [371484003],
Social_security_number_ [398093005], Occupation_(occupation) [14679004] and
County_of_residence_ [432407003].

Event (event) [272379006]: *Cause_of_accident_type_(qualifier_value)* [278443006] and *Origin* and *Type(attribute)* [410657003]

Therapeutic plan (Prescription_of_therapeutic_regimen_(procedure) [55053003]):
Scene, stimulus (image, text and audio), critic point, screen and trajectory

Temporal_observable [364713004]: *Date_of_diagnosis* [432213005], *Date_of_onset* [298059007], *Date_of_report* [399651003] and Frequency. Frequency has the subclasses of Occurrence, *Date_of_onset* [298059007], *Date_of_report* [399651003] and *Time_of_onset* [263501003].

Data is introduced in the ontology by means of instances of classes. In Figure 3, a person is shown with user id 1, who suffers from a traumatic brain injury (TBI) and follows therapeutic plan 1. This therapeutic plan is composed of the *eating* and *dressing* activities of daily living. In particular, the *eating* activity is managed by an occupational therapist, and its result is evaluated by the *functional independence measure* (FIM).

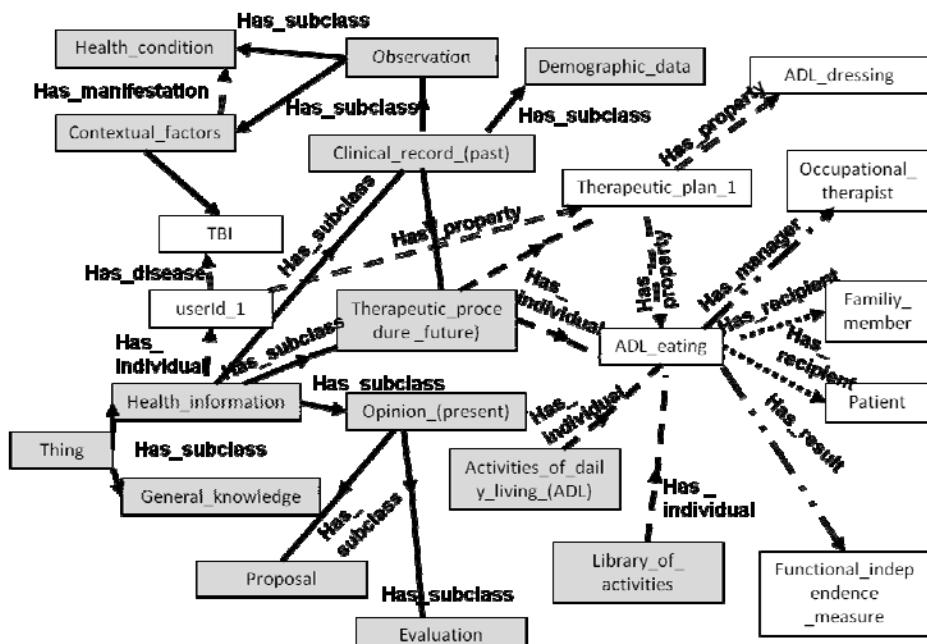


Fig. 3. Summary of relationships of classes and instances

4 Conclusions and Future Work

The use of ontologies and international standards allows to automatically share and reuse knowledge and to build more robust clinical decision support systems, which, in turn, can provide personalized therapies and evaluate the effectiveness of a treatment. Usually, in rehabilitation, these systems have, as objectives, diagnosis and personalization of therapeutic plans.

To represent a person's medical information, a methodology has been used, according to which concepts are encoded into international standards, and indicators and their values can be combined. The representation of a person's medical information is based on the reference model called *openEHR*. Furthermore, standard ontologies are used, such as ICF, ICD and SNOMED CT.

As future work, the ontology will be extended to other types of rehabilitation and functional diversities. Furthermore, a decision support system using the ontology will be implemented and evaluated.

Acknowledgements. The research described in this paper arises from a Spanish research project called Rehabilita (Disruptive technologies for the rehabilitation of the future), which is funded by CDTI, under the CENIT program, in the framework of the Spanish government's INGENIO 2010 initiative. This work is also partly supported by the Catalonia Competitiveness Agency (ACC1Ó). The opinions expressed in this paper are those of the authors and are not necessarily those of Rehabilita project's partners, CDTI or ACC1Ó.

References

- [1] Berner, E.S., La Lande, T.J.: Overview of Clinical Decision Support Systems. In: Comp. in Health Care, pp. 3–22. Springer, New York (2007)
- [2] Bates, D.W., et al.: Effect of Computerized Physician Order Entry and a Team Intervention on Prevention of Serious Medication Errors. American Med. Association 280(15), 1311–1316 (1998)
- [3] DeGruy, K.B.: Healthcare applications of knowledge discovery in databases. Health Inf. Manag. 14(2), 59–69 (2000)
- [4] Perreault, L.E., Metzger, J.B.: A pragmatic framework for understanding clinical decision support. Healthcare Inf. Manag. 13(2), 5–21 (1999)
- [5] Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun. 7(1), 39–59 (1994)
- [6] Holt, A., Bichindaritz, I., Schmidt, R., Perner, P.: Medical applications in case-based reasoning. Knowl. Eng. Review 20(3), 289–292 (2005)
- [7] Marling, C.R., Whitehouse, P.J.: Case-Based Reasoning in The Care of Alzheimer's Disease Patients. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 702–715. Springer, Heidelberg (2001)
- [8] Montani, S., Bellazzi, R., Portinale, L., Riva, A., Stefanelli, M.: Multi-modal Reasoning in Diabetic Patient Management. In: Horn, W., Shahar, Y., Lindberg, G., Andreassen, S., Wyatt, J.C. (eds.) AIMDM 1999. LNCS (LNAI), vol. 1620, pp. 113–123. Springer, Heidelberg (1999)

- [9] Palaudàries, A., Armengol, E., Plaza, E.: Individual prognosis of diabetes long-term risks: A CBR approach. *Methods of Inf. in Med.* (2011)
- [10] Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. *J. of Web Sem.* 5(2), 51–53 (2007)
- [11] World Health Organization, "International Classification of Functioning, Disability and Health (ICF)" (2011),
<http://apps.who.int/classifications/icfbrowser/>,
<http://bioportal.bioontology.org/ontologies/1411>
(accessed: June 15, 2011)
- [12] HPLabs, Jena, A Semantic Web Framework for Java (2010),
<http://openjena.org/index.html> (accessed: July 04, 2011)
- [13] OpenEHR Foundation, "The openEHR Reference Model," (2008),
http://www.openehr.org/releases/1.0.2/architecture/rm/ehr_im.pdf (accessed: July 04, 2011)
- [14] Cieza, A., Geyh, S., Chatterji, S., Konstanjsek, N.: ICF linking rules: an update based on lessons learned. *J. Rehabil. Med.* 37(4), 212–218 (2005)
- [15] Cimino, J.J.: Desiderata for Controlled Medical Vocabularies. In: Presented at the Methods Inf. Med., vol. 37, pp. 394–403 (1998)
- [16] Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing, Pacific Grove (2000)
- [17] Research Branch ICF Core Sets Projects, <http://www.icf-research-branch.org/icf-core-sets-projects.html> (Accessed: July 04, 2011)
- [18] Cieza, A., Brockow, T., Ewert, T., Amman, E.: Linking health-status measurements to the international classification of functioning, disability and health. *J. Rehabil. Med.* 34(5), 205–210 (2002)
- [19] College of American Pathologists y National Health Service, "Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT)",
<http://bioportal.bioontology.org/ontologies/42157?p=terms&c=conceptid=362981000#details> (accessed: June 15, 2011)
- [20] Beale, T., Heard, S.: An Ontology-based Model of Clinical Information. Presented at the MedInfo, Brisbane, Australia, pp. 760–764 (2007)
- [21] Dublin Core, Dublin Core Metadata Element Set, Version 1.1,
<http://protege.stanford.edu/plugins/owl/dc/protege-dc.owl>
(accessed: June 15, 2011)
- [22] National Centre for Classification in Health, "Condensed Classification of Health Development" (2004),
http://sydney.edu.au/health_sciences/ncch/condensed_classifications.shtml (accessed: July 04, 2011)
- [23] World Health Organization, "International Classification of Diseases (ICD-10)",
<http://apps.who.int/classifications/apps/icd/icd10online>,
http://dkm.fbk.eu/index.php/ICD-10_Ontology
(accessed: June 15 (2008))
- [24] Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. *J. of Web Sem.: Science, Services and Agents on the WWW* 4, 309–322 (2008)
- [25] Rodrigues, J.M., Kumar, A., Bousquet, C., Trombert, B.: Using the CEN/ISO Standard for Categorical Structure to Harmonise the Development of WHO International Terminologies. Presented at the Med. Inf. in a United and Healthy Europe, Sarajevo, Bosnia and Herzegovina (2009)

Heuristic Search of Cut-Off Points for Clinical Parameters: Defining the Limits of Obesity

Miguel Murguía-Romero^{1,*}, Rafael Villalobos-Molina¹,
René Méndez-Cruz², and Rafael Jiménez-Flores²

¹ Unidad de Biomedicina

² Carrera de Médico Cirujano

Facultad de Estudios Superiores Iztacala

Universidad Nacional Autónoma de México

Ave. de los Barrios 1, Los Reyes Iztacala, Tlalnepantla 54090, México

*miguelmurguia@gmail.com

Abstract. We studied the variability of obesity in a sample of 4,164 young Mexicans (17-24 years old) measured through the waist circumference. According to the American Heart Association, obesity is one of the five clinical alterations to define the Metabolic Syndrome (MS); the other four are low levels of HDL cholesterol, and high values of triglycerides, glucose, and blood pressure. It has been proposed a cut-off point of 80 cm for women and 90 cm for men to define a normal or altered value of waist circumference for Mexicans. We assume that the waist circumference in healthy population has a normal distribution, so a monolithic cut-off point is only an upper limit for normal values. The objective of this work is to estimate the subjacent normal distribution of the waist circumference of healthy people, involving in this analysis the other four components of the MS, and approaching the problem as a combinatory one. We defined a combination of cut-off points for the other four components of the MS; if considering a set of 50 cut-off points candidates for each of the five parameters, then results in a searching space of 50^5 (more than 300 millions of combinations). Each particular combination of cut-off points (excluding waist circumference) sets a subpopulation in which parameter values fall into normal ranges so defined; then for each subpopulation we calculated the histogram of the waist circumference values. Using a heuristic function involving the symmetry value of the histogram (skewness), we applied a ‘best first search’ on the combination of cut-off points. We found a combination of cut-off point values that generates the more symmetrical histogram, so we propose it as a useful criterion to set cut-off points of MS parameters for Mexicans. Finally, the obtained histogram is proposed as the normal distribution of healthy population, and represents the variability of the waist circumference of non-obese young Mexicans.

Keywords: Heuristic search, best first search, obesity, metabolic syndrome, normal distribution.

1 Introduction

1.1 Obesity and Metabolic Syndrome

Obesity is an abnormal or excessive fat accumulation that may impair health [1]. Abdominal obesity is a public health problem worldwide because its elevated prevalence [2,3], and is a risk factor for developing diabetes mellitus 2 (DM2) and cardiovascular disease (CVD) [4].

It is known that more than half of Mexicans are overweight or obese [5]. Waist circumference is one of the most used measurements to evaluate obesity; Mexican health authorities set a cut-off point of ≤ 85 cm in women, and ≤ 95 cm in men for waist circumference in Mexicans [6], while higher values represent an altered condition; nevertheless, the widely accepted cut-off points for waist circumference in Mexicans are <80 cm in women, and <90 cm in men [7,8]. Then the following questions arise: Are such cut-off points adequate for all age ranges? Are those the most adequate cut-off points for Mexicans? What does it mean that a person be obese? What implication for her/his health has becoming obese? How is the frequency distribution of the waist circumference values of healthy people?

Obesity has been included as a component of the Metabolic Syndrome (MS) [9,8]. Metabolic Syndrome is a group of risk factors to develop CVD and DM2 that involves a variety of disorders of diverse etiology, including obesity, dyslipidemia, high blood glucose, and high blood pressure, among others [9,8]. Thus, the MS is an adequate conceptual frame to study obesity; to address the questions previously raised, we propose to study obesity in the context of the other metabolic alterations included in the MS (Table 1).

Table 1. Reference values of clinical and anthropometric parameters according to the American Heart Association (AHA) metabolic syndrome definition [8,9]. AHA criterion indicates that metabolic syndrome is present when three or more parameters are altered, i.e. exceed the cut point.

Parameter	Categorical cut-off point
HDL Cholesterol	<50 mg/dL in women <40 mg/dL in men
Waist circumference	≥ 80 cm in women ≥ 90 cm in men
Triglycerides	≥ 150 mg/dL
Blood pressure	≥ 130 mmHg systolic ≥ 85 mmHg diastolic
Fasting glucose	≥ 100 mg/dL

1.2 Categorical Cut-off Points

Physicians use categorical references values as a threshold, i.e. if the patient has a HDL value lower than the reference value then he/she is diagnosed to have an

altered value. From the clinical point of view, that threshold approach is practical, because allows physicians to communicate in a congruent way. However, it is accepted that clinical parameters of healthy people show a normal distribution (i.e. Gaussian) as the very used method to set cut-off points based on cumulated centiles [10,11] is supported on this view.

Because the prevalence of obesity is very high among Mexicans, its estimation is very sensible to little variations of the cut-off point for waist circumference. Cut-off point for waist circumference could be best set if the underlying Gaussian distribution of healthy young is well known.

The objective of this work is to estimate the normal distribution of waist circumference of healthy young Mexicans (17-24 years old), searching in a combinatory space of cut-off points values of the other parameters related to MS (HDL-cholesterol, triglycerides, blood pressure, and glucose); and also, using the frequency distribution so found, to confirm or adjust the cut-off points of 80 cm for women and 90 cm for men.

2 Method

2.1 The Sample Data

As a part of a study from the research group GMISARA —stands for Multidisciplinary Research Group on Health and Academic Performance— to investigate the health of undergraduate students, a set of data including anthropometric parameters and blood samples were obtained from students from two public universities of Mexico City's metropolitan area. All students signed an informed consent to participate in the study. The blood pressure and anthropometric data collection —including waist circumference— were supervised or taken by two internist physicians of our research group; the blood samples were taken and analyzed by CARPERMOR S.A. de C.V., an international reference laboratory, including HDL cholesterol, triglycerides, and glucose, among others. A total of 4,164 students —2,872 women and 1,292 men— aged from 17-24 years old were included in this analysis.

2.2 Combinatory of Cut-off Points

The objective is to estimate the normal distribution of the waist circumference of healthy young, so a criterion was established to hypothesize that a person is healthy, i.e. that all parameters of MS match the criteria of 'normal' value. For each student, the waist circumference is included in the histogram if all the corresponding values of HDL-cholesterol, triglycerides, diastolic and systolic blood pressure, and glucose, fall into the normality or 'not altered' range, i.e. that none of all five parameters were altered. If the search space is composed of five parameters, each varying 50 units, then the total possible nodes are $50^5 = 312,500,000$.

The Best First Search algorithm (BFS) searches for a solution node in a tree [12], guided by a heuristic function, and expanding only a small sub tree

of all the combinatory [13]. The search space begins in a specific initial node, and expands the tree generating all the possible child nodes, these nodes are evaluated with a heuristic function that estimates how far the solution node is from such node. The heuristic function is static, i.e. evaluates the node, and not the position of the node in the tree. The BFS requires a mechanism to generate all the successors of a node, here denominated as **rules**. The initial tree is composed by only one node, the root, representing the initial conditions of the search.

We programed the best search algorithm in Prolog language, based on the code included in [14]. The nodes of the tree have the structure:

```
node(CP_HDL, CP_TG, CP_DBP, CP_SBP, CP_GLU).
```

The **CP_** prefix stands for ‘cut-off point’. Suffixes **HDL**, **TG**, **DBP**, **SBP**, and **GLU** stand for HDL-cholesterol, triglycerides, diastolic and systolic blood pressure, and glucose, respectively. A set of rules were defined to expand the node into its successors (Figure 1) that basically add and subtract a unit to each parameter. So, the successors of `node(50,150,80,120,100)` are:

```
node(51,150,80,120,100).
node(50,151,80,120,100).
node(50,150,81,120,100).
node(50,150,80,121,100).
node(50,150,80,120,101).
node(49,150,80,120,100).
node(50,149,80,120,100).
node(50,150,79,120,100).
node(50,150,80,119,100).
node(50,150,80,120,99).
```

To avoid local maximum, a set of ten additional rules were added, that adding and subtract five units instead the unit, adding ten more successor. So, the next nodes are added to the previous list:

```
node(55,150,80,120,100).
node(50,155,80,120,100).
node(50,150,85,120,100).
node(50,150,80,125,100).
node(50,150,80,120,105).
node(45,150,80,120,100).
node(50,145,80,120,100).
node(50,150,75,120,100).
node(50,150,80,115,100).
node(50,150,80,120,95).
```

Because it is accepted that the waist circumference is higher in men than in women, two searches were run independently, one considered only data for women and other for men.

Each node implies a subset of waist circumference values, since the node arguments represent the values of cut-off points that a person need to match. The next Prolog clause implements that specific meaning:

```
histo(node(CP_HDL, CP_TG, CP_DBP, CP_SBP, CP_GLU, L) :-
    !,
    findall(WC, (data(Id, Age, Sex, HDL, WC, TRI, PAD, PAS, GLU),
                  HDL>=CP_HDL, TRI<CP_TRI, PAD<CP_PAD, PAS<CP_PAS, GLU<CP_GLU),
            L).
```

The clause collects a list L with waist circumference values (WC) for those persons which parameter values satisfy:

- HDL-cholesterol greater or equal than CP_HDL,
- Triglycerides less than CP_TG,
- Diastolic blood pressure less than CP_DBP,
- Systolic blood pressure less than CP_SBP, and
- Glucose less than CP_GLU.

The root nodes for women and men, respectively, were:

```
node(50,150,80,120,100).
node(40,150,80,120,100).
```

Parting of the initial node, the BFS expands the tree using the rules (Figure 1) that generated the child nodes. In turn, the same rules are used to generate its child nodes.

2.3 The Normal Distribution Assumption

To identify the healthy population, we applied the notion that their waist circumference values are normally distributed (Gaussian). We hypothesised that such normal distribution will biases when also ‘not healthy’ is included: the more ‘healthy’ proportion, the more close the histogram shape to the bell curve.

2.4 The Heuristic Function: Symmetrical Normal Curve

As a bias measure of the normal curve of waist circumference values, we considered the symmetry of the histogram. To evaluate the symmetry of a histogram we used the skewness measure [15] using a statistics based on the second and third moments of the mean [16]. For a sample of n values the sample skewness s , is

$$s = \frac{m_3}{m_2^{3/2}} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^{3/2}}$$

where:

\bar{x} is the sample mean,
 x_i are the sample values,
 n is the sample size,
 m_2 is the second moment of the mean, and
 m_3 is the third moment of the mean.

2.5 Criteria for the Solution Node

The criteria for a solution node is to minimize the skewness, so the solution is defined as a node that has a heuristic function less than a threshold, initially set to a skewness value of 0.1. Other mandatory criterion for the solution was that it should include 20% of individuals in the original sample.

```
% Prolog rules
% rule(rule_name, input_node, output_node).

rule(-hdl, node(HDL,TG,DBP,SBP,GLU), node(HDL2,TG,DBP,SBP,GLU)) :-
    HDL2 is HDL - 1.
rule(-tg, node(HDL,TG,DBP,SBP,GLU), node(HDL,TG2,DBP,SBP,GLU)) :-
    TG2 is TG - 1.
rule(-dbp, node(HDL,TG,DBP,SBP,GLU), node(HDL,TG,DBP2,SBP,GLU)) :-
    DBP2 is DBP - 1.
rule(-sbp, node(HDL,TG,DBP,SBP,GLU), node(HDL,TG,DBP,SBP2,GLU)) :-
    SBP2 is SBP - 1.
rule(-glu, node(HDL,TG,DBP,SBP,GLU), node(HDL,TG,DBP,SBP,GLU2)) :-
    GLU2 is GLU - 1.
rule(+hdl, node(HDL,TG,DBP,SBP,GLU), node(HDL2,TG,DBP,SBP,GLU)) :-
    HDL2 is HDL + 1.
rule(+tg, node(HDL,TG,DBP,SBP,GLU), node(HDL,TG2,DBP,SBP,GLU)) :-
    TG2 is TG + 1.
rule(+dbp, node(HDL,TG,DBP,SBP,GLU), node(HDL,TG,DBP2,SBP,GLU)) :-
    DBP2 is DBP + 1.
rule(+sbp, node(HDL,TG,DBP,SBP,GLU), node(HDL,TG,DBP,SBP2,GLU)) :-
    SBP2 is SBP + 1.
rule(+glu, node(HDL,TG,DBP,SBP,GLU), node(HDL,TG,DBP,SBP,GLU2)) :-
    GLU2 is GLU + 1.
```

Fig. 1. Prolog rules used to generate the combinatory of cut-off points

3 Results

3.1 The Solution Nodes

The total nodes expanded to find a solution were 8,955 for women and 266,725 for men (Figure 2), where the minimum skewness values were <0.01 (0.0000895)

```
% solcut(>=HDL, <TG, <DBP, <SBP, <GLU, Skewness, n).

solcut(50, 150, 80, 120, 100, 1.7555765989530134, 767).
solcut(45, 150, 80, 120, 100, 1.3545800430171506, 1125).
solcut(50, 145, 80, 120, 100, 1.7854489937063036, 762).
solcut(50, 150, 75, 120, 100, 1.7913219788986792, 738).
solcut(50, 150, 80, 115, 100, 0.2553960978651566, 754).

:
solcut(49, 159, 79, 109, 95, -0.0028772172752248087, 579).
solcut(44, 159, 79, 109, 95, 0.23957306012137308, 812).
solcut(49, 154, 79, 109, 95, 0.005865458883888189, 576).
solcut(49, 159, 74, 109, 95, -0.02206887041142564, 554).
solcut(49, 159, 79, 104, 95, -0.04549356054360512, 551).
solcut(49, 159, 79, 109, 90, 0.001162552037332858, 475).
solcut(49, 164, 79, 109, 95, 0.0004392861867400391, 581).
solcut(49, 159, 84, 109, 95, 0.05977230005098815, 648).
solcut(49, 159, 79, 114, 95, 0.18942324572428104, 769).
solcut(49, 159, 79, 109, 100, 0.046997440642322384, 618).
solcut(48, 159, 79, 109, 95, 0.0959776657773151, 624).
solcut(49, 158, 79, 109, 95, -1.0895590268743898e-5, 578).
```

Fig. 2. Some expanded nodes by the ‘Best First Search’ algorithm. The first five parameters represent thresholds criteria to include the waist circumference value of a person in the histogram to evaluate; the sixth parameter is the skewness of the corresponding histogram; the last value is the number of persons matching the criteria specified by the first five parameters. Data are from young women.

for women, and 0.58 for men; the respective nodes were taken as the solutions. The solution nodes (Table 2) include a total of 578 young women (20.1% of the total sample), and 559 young men (43.3%). The cut-off points of the solution for women were $HDL \geq 49$, $TG < 158$, $DBP < 79$, $SBP < 109$, and $GLU < 95$; while $HDL \geq 41$, $TG < 176$, $DBP < 81$, $SBP < 125$, and $GLU < 108$ for men.

3.2 Biomedical Attributes of the Solutions

Analyzing the sample data filtered by the cut-off points of the solution, the relative frequency in women has a mode in the range 70-75 cm, with the 24.7%; whereas in men the mode is in the range 75-80 cm, with the 24.3% (Table 3; Figures 3 and 4).

Table 2. Statistics of solution histograms (BFS) compared to *All* and *AHA* histograms. *All*: histogram of entire sample; *AHA*: histogram including only young that do have none parameters altered, according to the American Heart Association definition of metabolic syndrome; *BFS*: histogram of the ‘Best First Search’ solution. Cut-off point columns show the values used to generate the respective histograms.

Histogram	Waist circumference (cm)					Cut-off points				
	Mean	SD	95-centile	Skewness	n	HDL \geq	TG<	DBP<	SBP<	GLU<
Women										
All	80.8	10.86	101	1.08	2,872	-	-	-	-	-
AHA	77.3	9.22	93	1.35	1,023	50	150	85	130	100
BFS	75.8	8.18	89	<0.01	578	49	158	79	109	95
Men										
All	84.2	11.67	106	0.91	1,292	-	-	-	-	-
AHA	79.4	8.91	96	0.70	581	40	150	85	130	100
BFS	79.8	8.73	96	0.58	559	41	176	81	125	108

Table 3. Relative frequencies of solution histograms (BFS) compared to *All* and *AHA* histograms. *All*: histogram of entire sample; *AHA*: histogram including only young that have none parameters altered, according to the American Heart Association definition of metabolic syndrome; *BFS*: histogram of the ‘Best First Search’ solution.

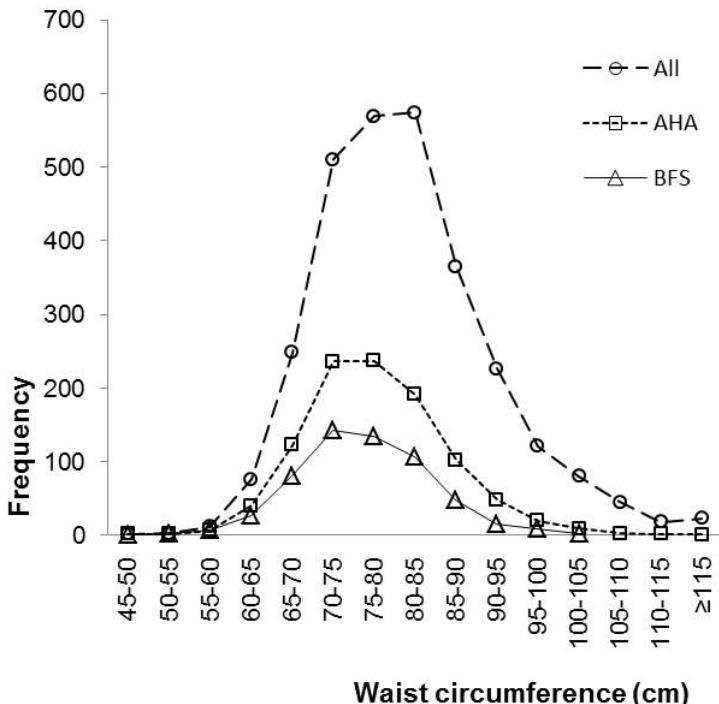


Fig. 3. ‘Best First Search’ (BFS) waist circumference histogram solution (*triangles*) for young women

The mean of waist circumference of the solution is 75.8 cm (SD=8.18) for women and 79.8 cm (SD=8.73) for men. The standard deviations were lower in the solutions compared with the entire sample. The centile 95 were 89 cm and 96 cm, for women and men, respectively, and also, were lower compared with the entire sample.

Graphically, the curves of the histograms for the entire sample look with a skew to the right. The curves of the solutions look bell shaped (Figures 3 and 4).

3.3 Prevalences of Obesity and Metabolic Syndrome

Using the widely accepted cut-off points for waist circumference of 80 cm (women) and 90 cm (men) the prevalence of obesity was 43.4% (50.6% women, 27.4% men), and the prevalence of metabolic syndrome was 14.2% (14.1% women, 14.3% men); while the prevalence of obesity calculated with cut-off points for waist circumference of the 95-percentile of the histogram solutions of 89 cm (women) and 96 cm (men) was 19.0% (20.0% women, 16.7% men), and the prevalence of metabolic syndrome was 9.4% (8.6% women, 11.2% men).

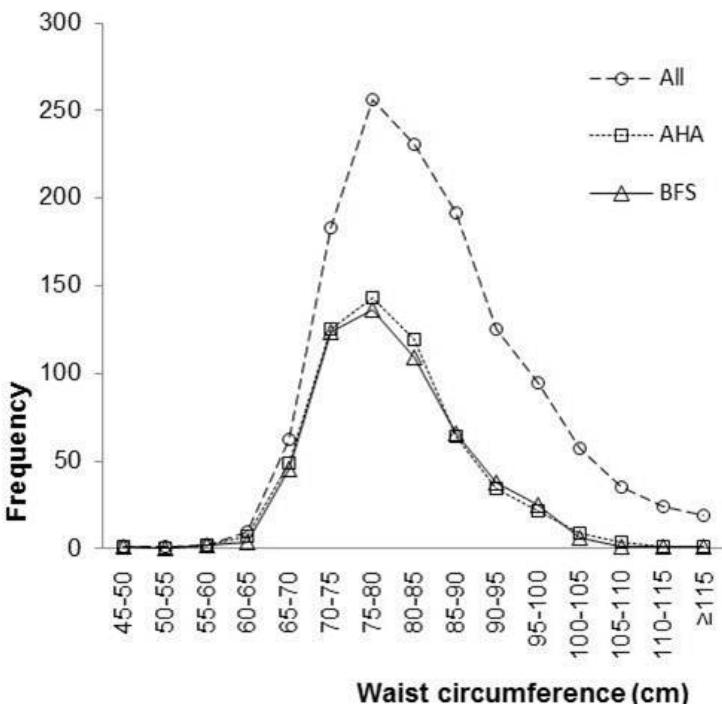


Fig. 4. ‘Best First Search’ (BFS) waist circumference histogram solution (*triangles*) for young men

4 Discussion

The solution node suggests that the waist circumference of healthy young has a normal distribution, so when analyzing the obesity of a population, the cut-off point needs to be applied carefully.

The 89 cm value of the centile 95 of the solution for women, compared with the widely accepted cut-off point of 80 cm for Mexican women, suggest that the current cut-off point to evaluate obesity needs to be revised.

Considering that the solution for men has a lower skewness than AHA histogram but graphically are very similar, suggests that skewness—as the main component of the heuristic function, and as solution criterion—is not sufficient to bring realistic solutions, and it could be complemented with a statistical test for homoscedasticity.

The MS prevalence for Mexicans is high, then the accuracy of the MS diagnoses has strong implications, thus a slight variation on the reference values may imply high variations on estimated number of persons with MS. At individual level, to establish whether or not a person has altered values of waist circumference is essential for the final identification of the MS.

5 Conclusions

Although the heuristic search is an AI technique proposed some decades ago, it remains as a very useful tool when problems are structured as a combinatory of states. Best First Search is a practical tool in biomedicine since it allows to investigate relationships among clinical parameters involved in complex diseases. The BFS algorithm brings an additional frame of communication between biomedical researchers, mainly because the combinatory approach complements other views, and the heuristic function allows an explicit form to find solutions; also, the initial conditions (root), and the goal node, or its conditions, are alternatives to represent biomedical information and knowledge.

Acknowledgments. We thank the staff of CARPERMOR S.A. de C.V. that took and analyzed the blood samples. This study was supported by grants PAPCA 95 2007-2008 FES Iztacala, PAPIIT IN226708 from U.N.A.M., and PICDS08-69 from ICyT- GDF, México.

References

1. World-Health-Organization: Obesity and overweight, Fact Sheet No. 311. WHO (2011), <http://www.who.int/mediacentre/factsheets/fs311/en/>
2. World-Health-Organization: Obesity: preventing and managing the global epidemic, Report of a WHO Consultation. World Health Organ. Tech. Rep. Ser. 894 (2000)
3. Popkin, B., Gordon-Larsen, P.: The nutrition transition: worldwide obesity dynamics and their determinants. *Int. J. Obes.* 28, S2–S9(2004)
4. Gautier, A., Roussel, R., Ducluzeau, P., Lange, C., Vol, S., Balkau, B., Bonnet, F., Group, D.S.: Increases in waist circumference and weight as predictors of type 2 diabetes in individuals with impaired fasting glucose: influence of baseline BMI: data from the DESIR study. *Diabetes Care* 33, 1850–1852 (2010)
5. Barquera, S., Campos-Nomato, I., Hernández-Barrera, L., Flores, M., Durazo-Arvizu, R., Kanter, R., Rivera, J.: Obesity and central adiposity in mexican adults: results from the Mexican National Health and Nutrition Survey 2006. *Salud Pública Méx.* 51(suppl. 4), S595–S603 (2009)
6. Secretaría-De-Salud: Manual de procedimientos. Toma de medidas clínicas y antropométricas en el adulto mayor, Secretaría de Salud. Mexico Federal Government (2002), <http://www.salud.gob.mx/unidades/cdi/documentos/D0CSAL7518.pdf>
7. Shamah-Levy, T., Cuevas-Nasu, L., Mundo-Rosas, V., Morales-Ruán, C., Cervantes-Turrubiates, L., Villalpando-Hernández, S.: Estado de salud y nutrición de los adultos mayores en México: resultados de una encuesta probabilística nacional. *Salud Pública Méx.* 50, 383–389 (2008)
8. Grundy, S., Cleeman, J., Daniels, S., Donato, K., Eckel, R., Franklin, B., Gordon, D., Krauss, R., Savage, P., Smith, S.J., Spertus, J., Costa, F.: Diagnosis and management of the metabolic syndrome. an American Heart Association/National Heart, Lung, and Blood Institute Scientific Statement. *Circulation* 112, 2735–2752 (2005)

9. Alberti, K., Eckel, R., Grundy, S., Zimmet, P., Cleeman, J., Donato, K., Fruchart, J.C., James, W., Loria, C., Smith, S.J.: International-Diabetes-Federation-Task-Force-On-Epidemiology-And-Prevention, National-Heart-Lung-And-Blood-Institute, American-Heart-Association, World-Heart-Federation, International-Atherosclerosis-Society, International-Association-For-The-Study-Of-Obesity: Harmonizing the metabolic syndrome: a joint interim statement of the International Diabetes Federation Task Force on Epidemiology and Prevention; National Heart, Lung, and Blood Institute; American Heart Association; World Heart Federation; International Atherosclerosis Society; and International Association for the Study of Obesity. *Circulation* 120, 1640–1645 (2009)
10. World-Health-Organization: Use and interpretation of anthropometric indicators of nutritional status. *Bull. World Health Organ.* 64, 929–941 (1986)
11. de Onis, M., Habicht, J.: Anthropometric reference data for international use: recommendations from a World Health Organization expert committee. *Am. J. Clin. Nutr.* 64, 650–658 (1996)
12. Pearl, J.: *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley (1984)
13. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* 4, 100–107 (1968)
14. Bratko, I.: *Prolog Programming for Artificial Intelligence*, 3rd edn. Addison Wesley (2000)
15. Groeneveld, R., Meeden, G.: Measuring skewness and kurtosis. *The Statistician* 33, 391–399 (1984)
16. D'Agostino, R., Belanger, A., D'Agostino, J.R.: A suggestion for using powerful and informative tests of normality. *The American Statistician* 44, 316–321 (1990)

Development of a System of Electrodes for Reading Consents-Activity of an Amputated Leg (above the knee) and Its Prosthesis Application

Emilio Soto¹, Jorge Antonio Ascencio², Manuel Gonzalez³,
and Jorge Arturo Hernandez¹

¹ Technological Institute of Puebla, Division of Graduate Studies and Research
Technological Avenue, Puebla, Pue. 420. cp72220
sotobungee@hotmail.com

INBIO, Biomedical Engineering S.R. M.I. de C.V. www.inbio.com.mx

² National Autonomous University of Mexico, Institute of Physics
Circuit of the Scientific Research in University City, CP 04510 México, D.F
ascencio@fis.unam.mx

³ Autonomous Popular University, Puebla State, Graduate of Biomedical Engineering
21 sur 1103 Col. Santiago C.P. 72410, Puebla, Pue. Mexico
manuel.gonzalez@upaep.mx

Abstract. It is reported the design of electrodes which was standardized and positioned based on the anatomy study and motor units, identified by the nerve branches of an amputated leg (above the knee)[1], obtaining the myoelectric signals of maximum amplitude of $20\mu V$ [2]. The initial identification of the optimal position of the electrode was characterized with myoelectrography in order to determine the movements that the patient makes consciously. The myoelectrical signal was accomplish by taking into account electrochemical schema of the cellular membrane [3] based on the fields and frequencies, recognized by the circuit structural materials that provide sufficient resistivity to the spurious frequencies [4], isolating each motor unit [5]. The results show that it is possible to have a cleaned signal which describes the movement that the patient desires consciously and the application in a prosthesis.

Keywords: Parameterizations myoelectric, structured material, Myoelectrical.

1 Introduction

This research is located in the area of the characterization of myoelectric signals in order to make a sketch of the patient's wishes to move his leg, his knee and his foot.

These movements are essential for the people's life. Through this research, they are detected myoelectric signals generated by the patient's stump to determine the cerebral impulse to move the knee.

As an example in Mexico there are 23,000 patients with this problem; nevertheless, only some people have the possibility of acquiring a prosthesis who are in a range from 10 years to 60 years of age.

On the first stage focuses on the detection of all myoelectric signals that move the knee.

2 Identification of Motor Units

The prominent signals that belong to the motion of the knee, are located in the femoral biceps and in the quadriceps. For an amputated patient in this section, it is required he has a good sensitivity, and can move these motor units without any hassle. For this reason, the number of electrodes used was three, two terminals along of the propagation zone of a subtraction point in a motor unit, and other electrode used as a reference. The three electrodes formed a triangulation of the signals generated by the patient, located as shown below.



Fig. 1. The research and the application

If the patient has a good muscular contraction can be sufficient to determine the movements of the extension or flexion of the knee.

3 Myoelectric Signals

Certain characteristics were looked for the myoelectric signal wave to capture an electrical impulse from a motor unit of a muscle in a patient. What is important is to grab a hold of the contraction that match to the desired movement. After capturing the signal with the aid of electrodes, the myoelectrical signal was filtered and adequate.

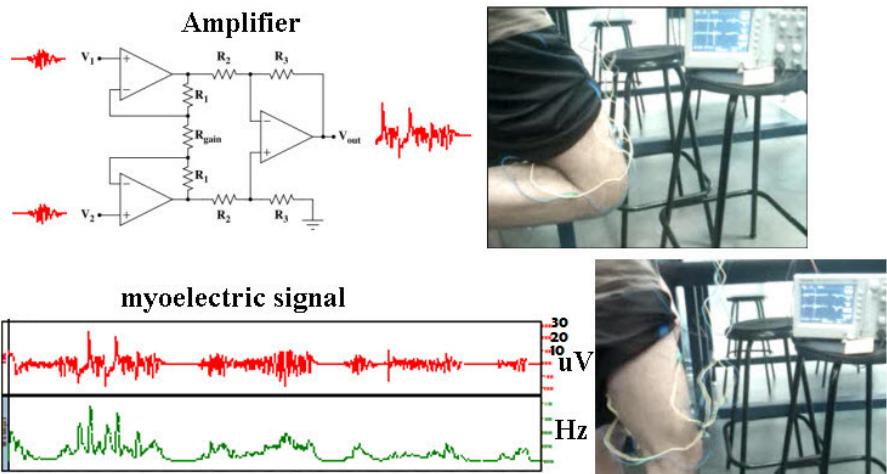


Fig. 2. Generation of the myoelectric signal

As shown in the figure 2 [3], the *RMS* value of surface *EMG* is a nonlinear function of muscle strength, even when relations of strength are obtained by a muscle in the leg, the velocity of the flexion-extension, or the change of the strength in the muscle, influence in the *RMS* value.

It is important to obtain a rectified pulse, with good identification of the wave produced by a specific motor unit; although, this is interfered by other muscles that are sensing at the same time in their contractions. That is why, it must analyze how the wave is generated along the cellular membrane.

4 Electrochemical Processes

The wave that produces has its origin in the sodium-potassium pump. The next figure has electrical relations which can be approximated to the behavior of the signal generated by a electrical circuit, determined by:

- V_m Membrane voltage (Volts)
- C_m Membrane capacitance (Farads)

- R_m Membrane resistance (Ohms)
- t_m Time constant of the membrane (seg)
- E_m Leakage reversal potential (Volts)

The figure of the cellular membrane is:

With the V_m , which is negative voltage with regard to grounding and the list of similarities in the sodium-potassium pump, the concept is to replicate with an electric circuit along all the cellular membrane that estimates the behavior of the myoelectrical signal wave.

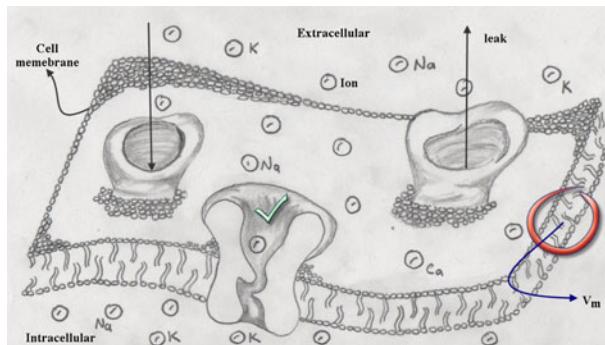


Fig. 3. Sodium and potassium pump

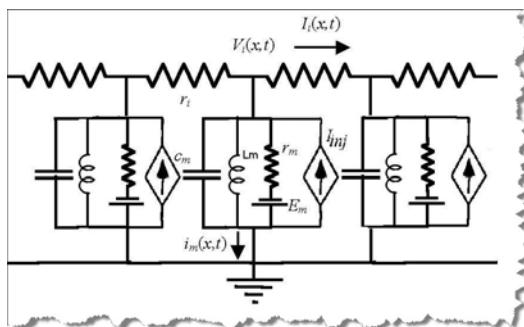


Fig. 4. Circuit with a complete resonator that simulates the cell membrane

Where it is: $i_m + i_{iny} = i_{m_c} + i_{m_L} + i_{m_r}$

Substituting equations, in the inductor: $i_{m_L} = \frac{1}{L} \int V_m dt$

In the capacitor: $C = \frac{q}{V}$; $VC = q$; $C_m \frac{\partial V_m}{\partial t} = i_{m_c}$

In the resistor: $V_m = R_m i_{m_R} + E_m$; $i_{m_R} = \frac{V_m - E_m}{R_m}$

As the current in membrane is: $i_m = \frac{1}{r_i + r_o} \frac{\partial^2 V_m}{\partial^2 x}$

Solving: $i_m = i_{m_c} + i_{m_L} + i_{m_r} - i_{iny}$ Using the cable's equation [6], the circuit of the cellular membrane is:

$$\frac{1}{r_i + r_o} \frac{\partial^2 V_m}{\partial^2 x} = C_m \frac{\partial V_m}{\partial t} + \frac{1}{L} \int V_m dt + \frac{V_m - E_m}{R_m} - I_{iny} \quad (1)$$

$$\lambda^2 \frac{\partial^2 V_m}{\partial^2 x} = \tau_C \frac{\partial V_m}{\partial t} + \tau_L \int V_m dt + (V_m - E_m) - I_{iny} \quad (2)$$

A simulation of this wave (equation 2), is shown in the figure 5, this signal was tried to obtain at the end of this research as showed in the figure 10, what was possible with the help of carbon fiber material.

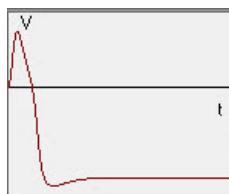


Fig. 5. Simulation wave

5 Insulation Materials

Because of the propagation characteristics that describe the wave signal within the motor units it is necessary to involve materials with certain characteristics. These materials are able to isolate locally the electromagnetic signals and reduce the propagation of these signals in the other muscle areas.

The signals in the muscles are a group of excitations for different innervations of axons of motor neurons, which make the contraction-relaxation movement, the force, the muscle tone and the response.

One element that can absorb the electrical signals by its mechanical-electrical and conductivity properties, is the carbon fiber, which it is designed as a circuit that could be a Faraday cage which isolates the myoelectric signal of the spurious signals on the motor unit, and it does not permit to enter noise that could mix with the frequencies generated by all ionic pulses of nerve innervations on the motor unit of the muscle.

The carbon fibers used are industrial type, produced by dissolution or the casting of carbon and stabilized by pre-oxidation or thermal stabilization, and finally carbonized in an inert atmosphere.

It is built a compoundable structure with form of continuous fiber sheets, linked to a polymer matrix, forming a sandwich like of the agglomeration of carbon fiber with the polymer, the matrix is insulating with epoxy and covers the fibers completely, the volume of the electrical resistivity can be infinite in the perpendicular direction to the fibers, and in the parallel direction to the fibers of the sandwich should have a very good electrical conduction [7].

Where the resistivity of the composite depends on the increase of the temperature when it is created and depends on the direction of the carbon fiber, and the thickness of the connections among the sheets of carbon fiber. So that the combination of electric flow among the sheets of carbon fiber, provides a circuit of current which can behave as a self-induction circuit, which means that the laminate composite behaves as an inductor.

This inductor made by a carbon structure, can capture a lot of invasive external signals.

In the internal circuit of the carbon structure is captured the desire myoelectric signal, the versatility of the structure made by the carbon materials, is referred to insulated of the invasive electronic fields to the capture myoelectrical signal [4].

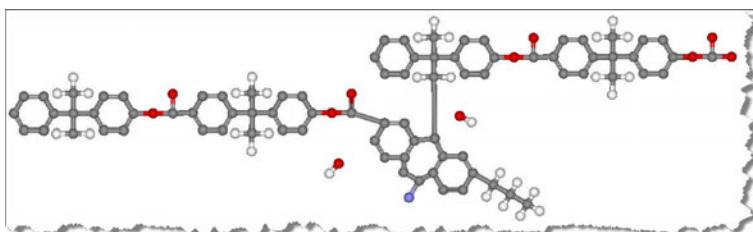


Fig. 6. Epoxy polymer with two molecules of OH

The structural integrated circuit is shown:

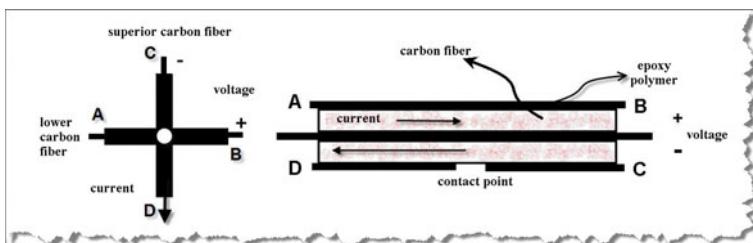


Fig. 7. Circuit structured composite

6 Instrumentation Results

As shown, the area of study focuses on the femoral biceps and triceps, with the help of an instrumentation amplifier; it is captured the myoelectric signal in the muscle, that is electrically neutral with the rest of the muscles, so that detects nerve activity during contraction or relaxation which is originated in the nerve impulse wave. This impulse is detected in the cellular membrane potential that can range from $50\mu V$ and $20\mu V$. The myoelectric signal using the instrumentation amplifier with filters is:

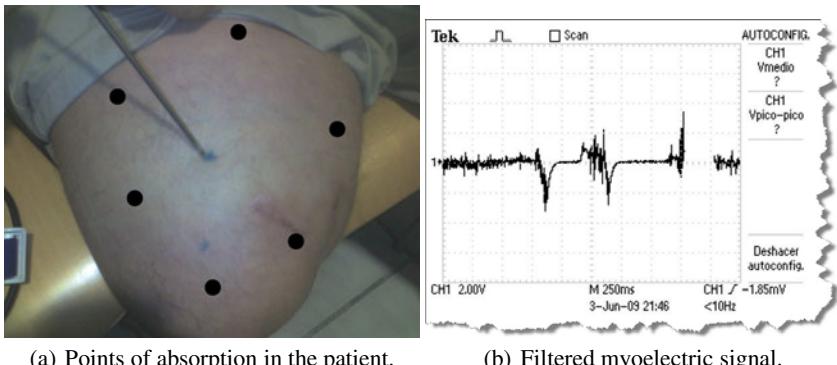


Fig. 8. The myoelectric signals obtained from the stump with only the instrumentation amplifier

It connects to the instrumentation amplifier with the following circuit made of carbon fiber.

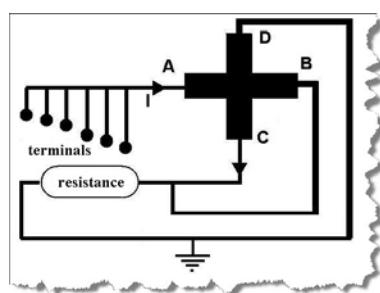


Fig. 9. Circuit basis of the composite insulation

And the myoelectric signal using the circuit-basis of the composite with carbon fiber are shown:

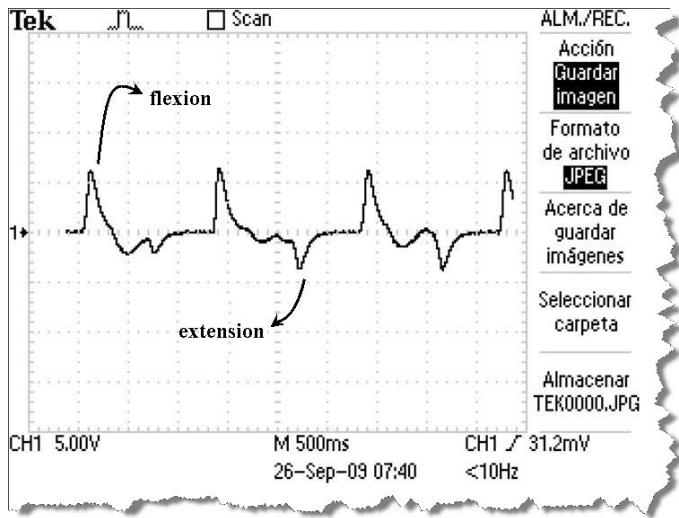


Fig. 10. Myoelectric signal obtained with Circuit structured composite

For a better insulation of myoelectric signal's acquisition, the patient's socket was wrapped with carbon fiber, as show in the figure 11.



Fig. 11. Socket in prosthesis

7 Application

One application of the circuit composite of carbon fiber, was the capture of the movements of the knee of a amputated patient, who moves automatically a prosthesis of leg.

The mechanical part was designed with the company INBIO, Biomedical Engineering in Aguascalientes City, which proposed a mechanical design based on a liner actuator that makes the knee's movement. This movement is shown in the figure 12.

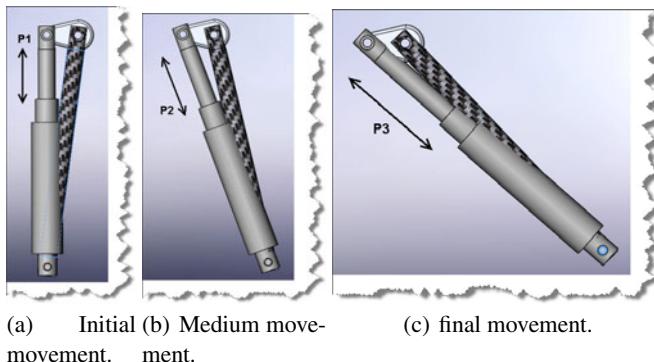


Fig. 12. The movement of the prosthesis

The mechanical design is shown below:

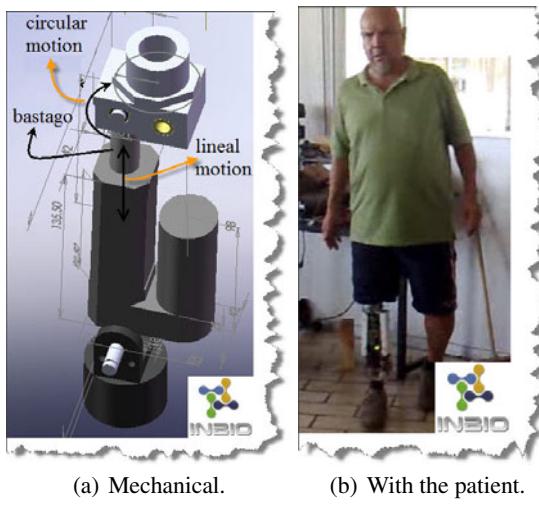
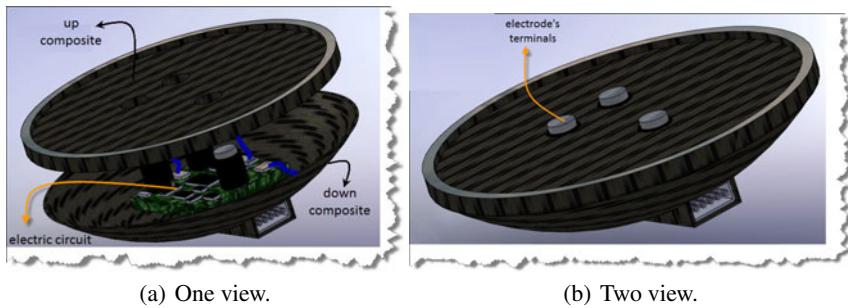


Fig. 13. The mechanics and coupler with the patient

For future applications a circuit design as a shell shape is proposed to integrate the electrode elements with the composite circuit of carbon fiber, is show in the next figure.

**Fig. 14.** Myoelectronic sensor

And the result is.

**Fig. 15.** Myoelectronic prosthesis

References

1. Drake, R.L., Vogl, W., Mitchell, A.W.M.: View points fitting of prostheses. Elsevier, Madrid (2007)
2. Malmivuo, J.: Understanding the myoelectric signal. Oxford University Press, New York (1995)
3. Rosch, P.J., Markov, M.S.: Understanding the myoelectric signal. Marcel Dekker, New York (2004)
4. Wang, S., Chung, D.L.L.: Electrical behavior of carbon fiber polymer-matrix composites in the through-thickness direction. Journal of Materials Science 25, 91–100 (2000)

5. de Pablo Gomez Navarro, P.J., Herrero, J.G., Biel, B., Vidal, J.G., Rubio, A., Flores, F.: Controlling the conductance of singlewalled carbon nanotubes. Departamento de Física de Materia Condensada, Universidad Autónoma de Madrid 4, 103–108 (2000)
6. Rosch, P.J., Markov, M.S.: Bioelectromagnetic medicine. Computers and the Humanities 34, 103–108 (2000)
7. Schrader, L., Jakob, H.D., Müller, H.P., Niggemann, J.: Mezclas de resinas de reacción termoendurecibles y su uso. Traducción De Patente Europea T3, Patente. 2 121 252, Solicitud: 95106227.2, k86 Número de publicación: 0 682 048, k87 (1995) Patente. 2 121 252

Predicting the Behavior of the Interaction of Acetylthiocholine, pH and Temperature of an Acetylcholinesterase Sensor

Edwin R. García*, Larysa Burtseva, Margarita Stoytcheva, and Félix F. González

Engineering Institute of Autonomous University of Baja California,

Calle de la Normal S/N, 21280, Mexicali, B.C., México

edwin.garcia@uabc.edu.mx*

{1pb, margarita}@iing.mxl.uabc.mx, ffgn2001@yahoo.com

Abstract. The steady-state current response of an acetylcholinesterase electrochemical sensor of second generation, which results from the interaction of substrate concentration, pH and temperature, was evaluated to improve biosensor's analytical characteristics using computational learning models. Artificial Neural Network and Support Vector Machine models demonstrated excellent results, despite of the limited number of samples. The predictions provided by both models were compared in order to determine which of them possesses a better approximation of the response generated by the sensor signal.

Keywords: biosensor, acetylcholinesterase, model, Artificial Neural Network, Support Vector Machine.

1 Introduction

Acetylcholinesterase AChE (EC 3.1.1.7) is an enzyme that belongs to the hydrolase class [1]. This substance is involved in the transmission of the nerve impulses, catalyzing the hydrolysis of the acetylcholine into choline inside insects, animals, and humans. Nevertheless, neurotoxic substances such as organophosphorus and carbamate pesticides, as well as warfare agents, inhibit the acetylcholinesterase action [2-4], which in turn affects the transduction of the nerve impulses and induces decreased intraocular pressure, bradycardia, hypotension, hypersecretion, bronchoconstriction, prolonged muscle contraction, up to the death. For this reason, the organophosphorus and carbamate compounds are effective for pest control. They are among the most widely used pesticides in agriculture [5], meanwhile they pose risks to damage the non-targeted organisms and the environment.

Many efforts have been made to evaluate pesticide concentration in environmental samples with the purpose of reduce potential health and ecological effects. The traditional methods are based on gas or high performance liquid chromatography [6, 7], requiring time-consuming extraction, pre-concentration, clean-up procedures, skilled personnel and expensive laboratory equipment. Their alternatives are the emerging

biosensors-based techniques providing fast, sensitive, selective, and accurate response without or with a minimum sample preparation using affordable, portable, and miniaturized instrumentation, appropriate for “in field” measurements [8].

According to the International Union of Pure and Applied Chemistry (IUPAC) definition, the biosensors are integrated receptor-transducer devices, which are capable of providing selective quantitative or semi-quantitative analytical information using a biological recognition element [9]. Most of the biosensors for organophosphorus pesticide analysis employ the acetylcholinesterase as biological specificity-conferring element. In the second-generation of electrochemical acetylcholinesterase sensors, the enzyme activity change is determined electrochemically through monitoring the thiocholine formed by enzymatic hydrolysis of acetylthiocholine, registering the current of its oxidation at a potential of +0.80 V/Ag, AgCl [10]. The organophosphorus pesticide level determination is based on the quantification of the acetylcholinesterase inhibition they provoke.

The steady-state current response of the acetylcholinesterase sensor is strongly dependent on the substrate concentration, the value of pH and temperature. The obtained overlapped information cannot be calibrated and modeled by linear expressions, resulting in the necessity of applying other methods such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN), coming to substitute the commonly used in analytical chemistry empirical approaches for signal improvement. The performance of ANNs for chemical applications was demonstrated in recent works for modeling and quantification purposes [11, 12]. The SVMs were developed in the reverse order to the ANNs: the SVMs have been evolved from the theory to the implementation and experiments; meanwhile the ANNs follow a more heuristic path, from applications and extensive experimentation to the theory. Today, the SVMs show the better or comparable performance to ANNs and other statistical models [13].

In this paper, the ANN and SVM regression models were chosen and applied as powerful techniques for estimating the analytical signal, resulting from the interaction of the substrate concentration, the pH factor, and the temperature, in an acetylcholinesterase electrochemical sensor of second generation.

2 Material and Equipment

The following reagents were used in the reactions [14]:

acetylcholinesterase (AChE), 2.5 mg/cm³;

solutions of acetylthiocholine (ACh) with different concentrations;

Britton-Robinson buffer (5 ≤ pH ≤ 9).

The electrochemical experiments were carried out in a conventional electrochemical cell. The biosensor was prepared through immobilization of the acetylcholinesterase enzyme by chemical bonds, on the surface of a graphite electrode (Ringsdorf Werke, Germany). The auxiliary electrode was made from a glassy carbon, and the reference electrode from saturated calomel.

The experimental data were obtained amperometrically. The rotation speed of the electrode (1000 rpm), and the enzyme concentration (2.5 mg/cm³) were kept constant.

The ACh concentration varied from 0.2 to 1.0 $\mu\text{mol/L}$ (0.2, 0.4, 0.6, 0.8, 1.0 $\mu\text{mol/L}$). The pH was kept equal to 5, 6, 7, 8, and 9. The temperature was changed in 6 cases (25, 30, 40, 50, 60, 70°C). The title of factors and output parameter, their notations, measuring units and the levels are shown in Table 1.

Table 1. Experiment parameters

Parameter	Notation	Measuring unit	Levels
Substrate concentration (ACh)	C_s	$\mu\text{mol/L}$	0.2, 0.4, 0.6, 0.8, 1.0
Acidity	pH	-	5, 6, 7, 8, 9
Temperature	t	°C	25, 30, 40, 50, 60, 70
Output current	I	μA	-

3 Suitable Computer Methods and Applications

Two appropriate computing methods for the regression modeling are considered below.

3.1 ANN Approach

An ANN consists of a large class of different architectures [15]. The most useful neural networks for function approximation are the Multilayer Layer Perceptron (MLP) that generalizes almost any function, and the Radial Basis Function (RBF) networks known owing to providing good universal approximations. Here we concentrate on the MLP networks.

One of the most used forms of MLP is the Error Back Propagation that is a powerful method for network training owing to differentiable activation functions. It minimizes the network error function improving its computational efficiency.

3.1.1 ANN Regression Model

After comparing various configurations to train the ANN¹ through Grid Search where different quantities of layers, neurons and trainings are used, it was concluded that the presented configuration gave the best result, consuming relatively small amounts of time. Other networks showed a very significant percentage of improvement consuming considerable time and resources. The chosen network was programmed with 30 neurons in each one of the 5 layers. The Levenberg-Marquadt algorithm was used for parameter optimization. It was applied an assessing technique 5k-fold Cross-Validation 5×5 [16] to improve the evaluation of the network that it had due to limited amount of collected data.

The proposed Neural Network Regression configuration is as follows:

¹ The neural network toolbox of the MATLAB was employed for the experiment.

Instance dimension	3×150 ;
Layers	5;
Neurons	$30 \times$ layer;
Assessing technique	5k-fold Cross-Validation;
Training	Levenberg-Marquadt algorithm;
Performance	Mean Square Error (MSE).

3.2 Support Vector Machines

The SVM is a supervised learning method that generates input-output mapping functions from a set of labeled training data [17-19]. In addition to its solid mathematical foundation in statistical learning theory, the SVM has demonstrated highly competitive performance in numerous real world applications, such as bioinformatics, text mining, face recognition, and image processing, which have positioned the SVM as one of the state-of-the-art tools for machine learning and data mining [20, 21]. The SVM method has been extended successfully to regression and density estimation problems [22]. This type of SVM is known as Support Vector Regression (SVR). The goal of a SVR is to produce a model based on the training data with the purpose of predict the target test data values given through only the test data attributes.

3.2.1 SVR Model

After the pre-processing of data, a Grid Search process with an exponential increase of the parameters, as it is suggested in [23], was employed to find the most suitable parameters of the SVR². Only three parameters are needed: the penalty parameter of the error term (C), the insensitive loss function for the noise tolerance control (ε), and the kernel parameter (α).

Several nonlinear kernels (radial, polynomial and sigmoid) were tested; each search result was taken for the reduction of the parameter range to improve the search itself, applying at the same time a 5k-fold cross validation. In this way, the best possible parameter values were ensured. Then, the best kernel results were compared to select the one with the smallest MSE. Once found the parameters, the model was trained using the best configuration to simulate the test data. In this way, the function predicted by the SVR was compared against the test data.

The proposed SVR configuration is as follows:

Instance dimension	3×150 ;
C	$2^9 = 512$;
α	$2^{2.21} = 4.6268$;
ε	0.075;
Kernel	Radial Basis Function;
Assessing technique	5k-fold Cross-Validation;
Performance	MSE.

² The MATLAB implementation *libsvm* [24] for the SVR experimentation was used.

4 Results

The predicted output parameter values were compared to evaluate the computer models described in this paper. Both of them were simulated with the same test and training data.

4.1 ANN Modeling the Influence of Parameters on the Biosensor Output

A simulation was realized for the parameters (C_S , pH and t) behavior prediction. The previously trained ANN described in Section 3.1 was compared against the original data. In this manner, the mutual influence of the model parameters was investigated. Between the two developed models, the ANN was most affected by the limited amount of the training data; and therefore it would be most beneficial in case of interpolated data. Figure 1 shows the comparison of the original data against the ANN prediction.

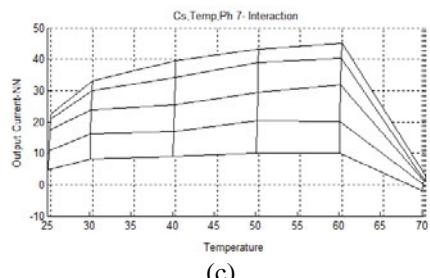
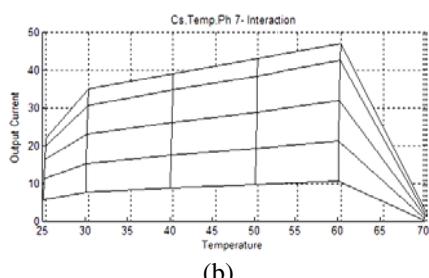
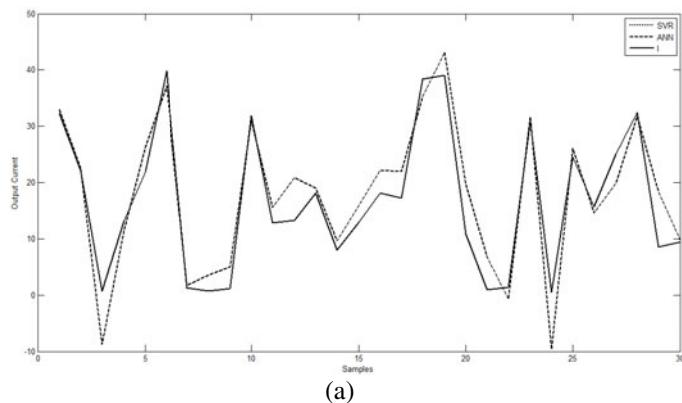


Fig. 1. Comparative graphics of the ANN approximation and experimental data:(a) Output current against one predicted by the ANN model; (b) Plot surface of experimental data; (c) Plot surface of ANN predicted data.

4.2 SVR Modeling the Influence of Parameters on the Biosensor Output

For the training of the SVR based on the configuration proposed in Section 3.2. The model was trained and tested employing the same data partition that the one for the ANN. The comparison of the SVR against the output parameter of the original data is shown on Figure 2a. A simulation of the original data and the one predicted by the SVR is shown on Figures 2b and 2c, respectively, with pH = 7, since that level is where the output current take the highest value.

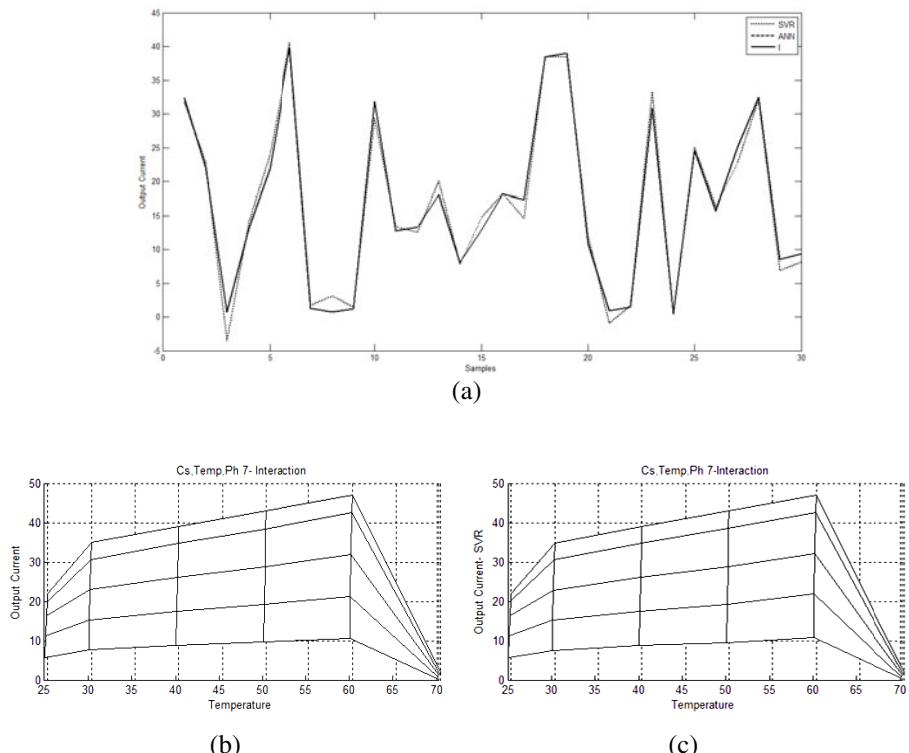


Fig. 2. Comparative graphics of the SVR approximation and experimental data: (a) Output current of test data in contrast with SVR predicted data; (b) Plot surface of experimental data; (c) Plot surface of SVR predicted data.

4.3 Comparative Analysis of Computational Models

The comparative quantitatively and qualitatively analysis was realized observing the test chart of the results thrown by the models, as well as comparing the MSEs. Figure 3 shows the charts of the original data and the data produced by the ANN and SVR.

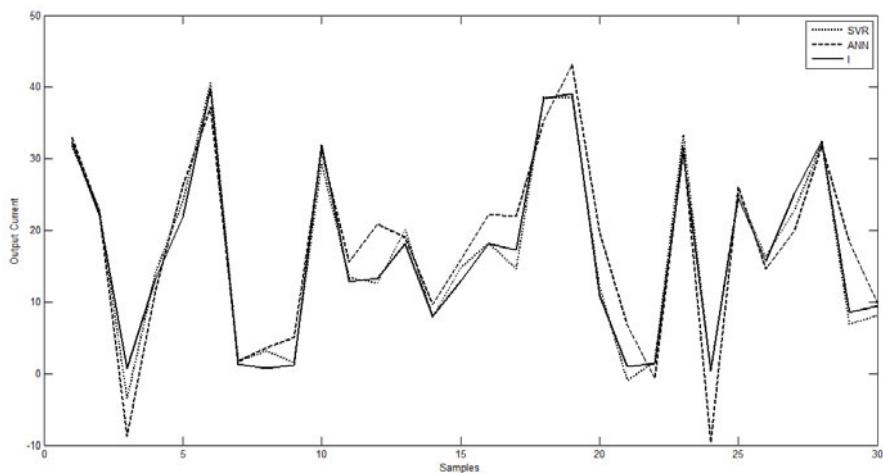


Fig. 3. The comparison of the experimental data vs. the ANN and SVR approximations

Table 2 shows the quantitative comparison between models and test data, based on the MSE of every model; three random data samples were taken. As the Figure 3 and Table 2 prove, both models shown good approximations but the SVR model behavior is more exact.

Table 2. Quantitative comparison of three random samples of experimental data, ANN and SVR approximations

No	Test Data			ANN		SVR		
	C_s , μmol	pH	$t, ^\circ\text{C}$	$I, \mu\text{A}$	$I, \mu\text{A}$	Error, %	$I, \mu\text{A}$	Error, %
1	0.2	6	30	8.03	9.68	1.65	7.84	0.19
2	0.8	7	50	18.15	22.21	4.06	18.26	0.09
3	0.4	7	40	38.41	35.27	3.14	38.56	0.15
Performance (MSE), %						2.95	0.143	

5 Conclusions

This paper describes the performance of two computational methods applied to model the variable-response in the outflow of an acetylcholinesterase sensor, taking into account the influence of the substance concentration, pH and temperature. The superiority of SVR in respect to the ANN model is clear, in both

accuracy and speed, considering the time a not very important parameter, given the conditions for use of the biosensor.

It is considered that in order to improve the results applying any of these models is required the interpolation procedure to obtain new data. The final benefit of this work is the optimization of the parameters that interact in this biosensor.

Acknowledgements. We thank CONACYT for the scholarship awarded, from which this work is derived.

References

1. Schomburg, D., Schomburg, I., Chang, A.: Springer handbook of enzymes, vol. 9. Springer, Berlin (2003)
2. Gupta, R.C. (ed.): Toxicology of organophosphate & carbamate compounds. Elsevier Academic Press, London (2005)
3. Fukuto, R.: Mechanism of action of organophosphorus and carbamate insecticides. Environmental Health Perspectives 87, 245–254 (1990)
4. Matsumura, F.: Toxicology of insecticides. Plenum Press, New York (1980)
5. FAOSTAT, <http://faostat.fao.org/site/424/default.aspx#ancor> (accessed on January 09, 2011)
6. Jeannot, R., Dagnac, T.: In: Nollet, L. (ed.) Chromatographic Analysis of the Environment, pp. 841–889. CRC Press, Boca Raton (2006)
7. Schlecht, P.C., O'Connor, P.F. (eds.): NIOSH manual of analytical methods, 4th edn. DHHS (NIOSH) Publication 94-113 (1994)
8. Rodriguez-Mozaz, S., Marco, M.-P., Lopez de Alda, M.J., Barcela, D.: Biosensors for environmental applications. Future development Trends. Pure and Applied Chemistry 76(4), 723–752 (2004)
9. Thévenot, D.R., Tóth, K., Durst, R.A., Wilson, G.S.: Electrochemical biosensors: recommended definitions and classification. Pure and Applied Chemistry 71(12), 2333–2348 (1999)
10. Stoytcheva, M., Zlatev, R., Velkova, Z., Valdez, B.: Organophosphorus Pesticides Determination by Electrochemical Biosensors. In: Stoytcheva, M. (ed.) Pesticides-Strategies for Pesticides Analysis, InTech, Croatia, pp. 359–372 (2011)
11. Rangelova, V., Tsankova, D., Dimcheva, N.: Soft Computing Techniques in Modeling the pH and Temperature on Dopamine Biosensor. Intelligent and Biosensors, 386 (2010)
12. Alonso, G., Istamboulie, G., Ramirez-Garcia, A., Noguer, T., Marty, J., Muñoz, R.: Artificial neural network implementation in single low-cost chip for the detection of insecticides by modeling of screen-printed enzymatic sensor response. Computers and Electronics in Agriculture 74, 223–229 (2010)
13. Betancourt, G.: Las Máquinas de Soporte Vectorial. Scientia et Technica Año XI (27) (April 2005)
14. Britton, H.T.K., Robinson, R.A.: CXCVIII-Universal buffer solutions and the dissociation constant of veronal. Journal of the Chemical Society, 1456–1462 (1931)
15. Bishop, C.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1995)
16. Refaeilzadeh, P., Tang, L., Liu, H.: Cross-Validation: Encyclopedia of Database Systems. Springer, Heidelberg (2008)
17. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)

18. Smola, A.J., Scholkopf, B.: A tutorial on Support Vector Regression (September 2003)
19. Hamel, L.: Knowledge Discovery with Support Vector Machines. University of Rhode Island (2009)
20. Pedroza, G.: Aplicación de las Maquinas de Soporte Vectorial a Reconocimiento de Hablantes. Universidad Autónoma Metropolitana, México (2007)
21. Gunn, S.R.: Support Vector Machines for classification and regression. University of Southampton, Mayo (1998)
22. Cortes, C., Vapnik, V.: Support-vector network. Machine Learning (1995)
23. Hsu, C.-W., Chang, C.-C., Lin, C.-J.: A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University (2003)
24. Chang, C.C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (2011)

Author Index

- Acosta Mesa, Héctor-Gabriel II-225
Aguilar, Juan Manuel Peña I-515
Aguilar, Omar Alejandro I-490
Aguirre L., Marco A. I-277
Agussurja, Lucas I-415
Alarcon, Teresa I-504
Albores-Velasco, Francisco Javier II-443
Alcaraz, Jorge Alberto Soria II-177
Aldana-Franco, Fernando I-466
Altamirano, Leopoldo I-537
Álvarez, Alberto Lamadrid I-515
Álvarez-Olguin, Gabriela II-249
Alvez, Carlos II-285
Argotte, Liliana I-222
Arroyo, Gustavo I-222
Ascencio, Jorge Antonio I-572
Aslam, Muhammad I-382, I-524
Ayala-Ramirez, Victor II-452
- Baltazar Flores, Rosario II-177
Bandyopadhyay, Sivaji I-267
Barman, Utsab I-267
Barreda, Dennis I-338
Barták, Roman I-162
Batyrshev, Ildar II-70
Beigi, Akram II-153
Bello, Rafael II-82
Bello Pérez, Rafael II-351
Benotti, Luciana I-345
Bergamo, Yannick Plaino I-454
Bertoá, Nicolás I-345
Brena, Ramón F. II-201
Burtseva, Larysa I-583
- Cadena, Pedro I-113
Caeiros, Alfredo Victor Mantilla II-319
Calderon, Felix II-398
Camiña, Benito I-174
Carballo Hernández, José Juan II-522
Carballido, José Luis I-16
Cardoso, Janette II-70
Carrizales-Turrubiates, Oscar I-198
Castañón-Puga, Manuel I-65
Castillo, Norberto II-119
- Castillo, Oscar II-1, II-58, II-131, II-331, II-363
Castro, Juan R. II-46
Ceberio, Martine II-13, II-70
Ceccaroni, Luigi I-549
Cervantes, Jair I-187, II-261
Cervantes, Leticia II-1
Chanona-Hernandez, Liliana I-328
Cheng, T.C.E. I-76
Chovanec, Andrej I-162
Colores, Juan M. II-499
Conant-Pablos, Santiago Enrique I-125
Cortes-Berrueco, Luis Enrique II-235
Costa, Anna Helena Reali I-454
Cozman, Fábio Gagliardi I-28, II-343
Cruz, Laura II-119
Cruz, Zenaida I-16
Cruz-Barbosa, Raúl II-249
Cruz-Cortés, Nareli II-189
Cruz-Ramírez, Nicandro II-225
Cruz Reyes, Laura I-137
Cruz Reyes, Rafael I-305
Cruz-Santiago, Rene II-474
Cuadros, Ernesto I-338
Cueva, Diego R. II-343
Cunha, Iria da I-316
- Dalmau, Oscar I-504
Delgadillo, Miguel A. II-95
de Lope, Javier I-443
Dey, Debangana I-357
Diegel, Olaf I-407
Domínguez, Martín Ariel I-244
Domínguez Mayorga, Carlos Roberto II-374
Dubois, Didier II-70
- Elalouf, Amir I-76
Escalada-Imaz, Gonzalo I-524
Escalante, Hugo Jair I-232, I-357
Escamilla Hernández, Enrique II-374
Espejel Rivera, María Angélica II-37, II-374

- Fabian, Aldo I-258
 Faghihi, Usef I-478
 Faili, Heshaam I-394
 Figueroa, Fernando David Ramirez II-319
 Filiberto Cabrera, Yaima II-351
 Flores, Dora-Luz I-65
 Flores-Pulido, Leticia II-443
 Fournier-Viger, Philippe I-478
 Frost, Richard A. I-291
- Galeano, July II-431
 Galicia-Haro, Sofía N. I-328, I-370
 Gallegos-Funes, Francisco Javier II-474
 García, Edwin R. I-583
 García, Farid I-187
 García, María M. II-82
 García, Uriel A. II-95
 García Hernández, René I-305
 García-Vázquez, Mireya II-499
 García-Vega, Angélica I-88
 Garrido, Leonardo I-113
 Garza Villarreal, Sara Elena II-201
 Gaxiola, Fernando II-363
 Gaxiola-Pacheco, Carelia I-65
 Gelbukh, Alexander I-267, I-305, I-370, II-213
 Gómez, Claudia II-119
 Gómez-Gil, Pilar I-537
 Gonçalves, Rafael A.M. II-343
 González, Aram B. I-101
 González B., Juan J. I-277
 González Mendoza, Miguel II-510
 González, Félix F. I-583
 Gonzalez, Jesus A. I-537
 Gonzalez, Manuel I-572
 Gonzalez-Hernandez, Loreto I-210
 Guillén Galván, Carlos II-463
- Hafiz, Rahmatullah I-291
 Hayet, Jean-Bernard II-407
 Hernández Aguirre, Arturo II-165
 Hernandez, Jorge Arturo I-572
 Hernandez, Manuel I-258
 Hernández Torres, Pablo II-37
 Hernández, Paula II-119
 Hernandez-Belmonte, Uriel H. II-452
 Hernández Cuamatzi, Patrick II-443
 Hernandez-Morales, Cindy G. II-107
 Herrera Alcántara, Oscar II-510
- Herrera-de-la-Cruz, Juve Andrea I-328
 Hoder, Kryštof I-1
 Hofestädt, Ralf II-143
 Hossain, Md. Zulfikar I-407
 Huegel, Joel Carlos I-490
 Huhns, Michael I-88
- Ibargüengoytia, Pablo H. II-95
 Ibarra, Jesús Eduardo Carrillo I-137
 Infante-Lopez, Gabriel I-244
- Jacob, Christelle II-70
 Jamshaid, Omer I-524
 Jimenez, Gonzalo Ramos II-351
 Jiménez-Flores, Rafael I-560
 Jiménez Vargas, Sergio II-213
 Jolivot, Romuald II-431
 Jose-Garcia, Adan II-107
 Júnez-Ferreira, Carlos A. II-398
- Kido, Hiroyuki I-52
 Kovács, Laura I-1
 Kreinovich, Vladik II-13, II-70
 Kurekin, Andrey II-487
 Kuri-Morales, Angel II-235
- Lau, Hoong Chuin I-415
 Ledeneva, Yulia I-305
 León, Maikel II-82
 Levner, Eugene I-76
 Li, Xiaouo II-261, II-273
 Licea, Guillermo II-46
 Locés, Mario César López I-137
 López, Asdrúbal I-187
 López Chau, Asdrúbal II-261
 López, Roque Enrique I-338
 Lopez-Arevalo, Ivan II-297
 Luis-Pérez, Félix Emilio II-249, II-419
 Luna Sánchez, Marlon II-443
 Lukin, Vladimir II-487
- Madrigal, Francisco II-407
 Mannan, Abdul I-524
 Maravall, Darío I-443
 Marin-Castro, Heidy M. II-297
 Markowska-Kaczmar, Urszula II-386
 Martínez, Luis G. II-46
 Martinez-Enriquez, Ana Maria I-382, I-524
 Martínez-Molina, Mario II-189

- Martínez-Velazco, Wilebaldo II-419
 Marzani, Franck II-431
 Matos, Tiago I-454
 Mayers, André I-478
 Mejía-Alvarez, Pedro II-261, II-273
 Melin, Patricia II-1, II-131, II-331, II-363
 Méndez-Cruz, René I-560
 Meza, Ivan I-258
 Mezura-Montes, Efrén II-225
 Minaei-Bidgoli, Behrouz I-394
 Mishin, Andrey II-24
 Molina, Alejandro I-316
 Monroy, Raúl I-174
 Montes-Gonzalez, Fernando I-466
 Montes y Gómez, Manuel I-232, I-357
 Montiel Soto, Romyna I-305
 Mora-Lumbreras, Marva Angelica II-443
 Moreno-Armendáriz, Marco A. II-189
 Mota, Yailé Caballero II-351
 Mújica-Vargas, Dante II-474
 Muñoz, Juan Francisco Reyes I-515
 Murguía-Romero, Miguel I-560
 Nápoles, Gonzalo II-82
 Naredo, Enrique II-58
 Nguifo, Engelbert Mephu I-478
 Nitta, Katsumi I-52
 Nkambou, Roger I-478
 Noguez, Julieta I-222
 Ochoa, Alberto II-119
 Ochoa-Luna, José Eduardo I-28
 Ortiz-Bayliss, José Carlos I-125
 Osorio, Mauricio I-16
 Özcan, Ender I-125
 Pakray, Partha I-267
 Parkes, Andrew J. I-125
 Pastrana Palma, Alberto I-515
 Paryvin, Hamid II-153
 Pazos R., Rofolfo A. I-277
 Pereira-Barretto, Marcos R. II-343
 Pérez, Luis Rodrigo Valencia I-515
 Pérez-Meana, Héctor II-499
 Perrussel, Laurent I-40
 Pineda, Luis I-258
 Pogrebnyak, Oleksiy II-487
 Ponomarenko, Nikolay II-487
 Portilla-Flores, Alberto II-443
 Posadas-Durán, Juan Pablo I-328
 Pozos-Parra, Pilar I-40
 Puchalski, Adam II-386
 Quiñonez, Yadira I-443
 Quiroz Castellanos, Marcela I-137
 Ramírez-Acosta, Alejandro II-499
 Ramírez Uresti, Jorge A. I-101
 Ramos Fernández, Julio Cesar II-37, II-374
 Ramos Velasco, Luis Enrique II-37, II-374
 Rangel-Valdez, Nelson I-198, I-210
 Rasooli, Mohammad Sadegh I-394
 Rechy-Ramírez, Fernando II-225
 Revoredo, Kate I-28
 Reyes-García, Carlos A. I-537
 Ríos-Mercado, Roger Z. II-307
 Rivera, Gilberto II-119
 Rivera, Mariano II-407
 Rivera-Islas, Ivan II-107
 Rodríguez, Lisbeth II-273
 Rodríguez-Cristerna, Arturo II-107
 Rodríguez-Díaz, Antonio II-46
 Rodríguez-Gómez, Gustavo II-443
 Romero-Monsivais, Hillel II-107
 Romero-Rodríguez, Wendoly J. Gpe. II-177
 Rosales-Pérez, Alejandro I-537
 Rosales-Silva, Alberto J. II-474
 Rutko, Dmitrijs I-149
 Salazar-Acosta, Juan C. II-307
 Sánchez, Daniela II-331
 Sánchez, Erika I-174
 Sánchez Fernández, Luis Pastor II-522
 Sanchez-Yanez, Raul E. II-452
 SanJuan, Eric I-316
 Santillán, Claudia Gómez I-137
 Seck Tuoh Mora, Juan Carlos II-189
 Segovia Domínguez, Ignacio II-165
 Sidorov, Grigori I-328
 Sierra, Gerardo I-316
 Silva, Valdinei Freire da I-454
 Solorio, Thamar I-232, I-357
 Sosa-Sosa, Victor J. II-297
 Sotelo-Figueroa, Marco Aurelio II-177
 Soto, Emilio I-572

- Starostenko, Oleg II-443
Stoytcheva, Margarita I-583
Subirats, Laia I-549
Sucar, L. Enrique I-429
Syed, Afraz Zahra I-382, I-524
- Tanveer, Saad I-524
Tejada, Javier I-338
Terashima-Marín, Hugo I-125
Thevenin, Jean Marc I-40
Torres-Jiménez, José I-137, I-198,
I-210, II-107
Torres-Moreno, Juan-Manuel I-316
Trejo, Luis A. I-174
Trejo-Baños, Daniel II-235
Trueba, Adrián I-187
Trujillo-Romero, Felipe II-419
- Ulyanov, Sergey II-24
Uriarte Adrián, Jesus II-463
- Valdés Amaro, Daniel II-463
Valdez, Fevrier II-131, II-363
Valdez, Guadalupe Castilla I-137
Vanhoof, Koen II-82
Vásquez, Juan Irving I-429
Vecchietti, Aldo II-285
Velázquez-Morales, Patricia I-316
Villa Diharce, Enrique II-165
Villalobos-Molina, Rafael I-560
Voronkov, Andrei I-1
- Waissman Vilanova, Julio II-37
- Yeap, Wai I-407
Yu, Wen II-261
- Zamudio Rodríguez, Victor Manuel
II-177
Zavala, Laura I-88
Zepeda, Claudia I-16