

LNAI 7557

Allan Ramsay  
Gennady Agre (Eds.)

# Artificial Intelligence: Methodology, Systems, and Applications

15th International Conference, AIMSA 2012  
Varna, Bulgaria, September 2012  
Proceedings

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*

Allan Ramsay Gennady Agre (Eds.)

# Artificial Intelligence: Methodology, Systems, and Applications

15th International Conference, AIMA 2012  
Varna, Bulgaria, September 12-15, 2012  
Proceedings



## 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

Allan Ramsay

University of Manchester, School of Computer Science

Oxford Road, Manchester, M13 9PL, UK

E-mail: allan.ramsay@manchester.ac.uk

Gennady Agre

Bulgarian Academy of Sciences

Institute of Information and Communication Technologies

2 Acad. G. Bonchev, 1113 Sofia, Bulgaria

E-mail: agre@iinf.bas.bg

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-642-33184-8

e-ISBN 978-3-642-33185-5

DOI 10.1007/978-3-642-33185-5

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012945548

CR Subject Classification (1998): I.2.4, I.2.6-8, I.2.10-11, H.2.8, H.3.3-5, H.4, H.5.3, I.4.9, I.6.3

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2012

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](http://www.springer.com))

# Preface

AIMSA-2012 was the 15th in a biennial series of AI conferences that have been held in Bulgaria since 1984. The series started as a forum for scientists from Eastern Europe to exchange ideas with researchers from other parts of the world, at a time when such meetings were difficult to arrange and attend. The conference has thrived for nearly 30 years, and now functions as a place where AI researchers from all over the world can meet and present their research. There may no longer be a need for special meetings that allow people from different parts of the world to be in one place, but there is clearly still a need for high-quality conferences, and I am delighted that AIMSAs is still fulfilling that role.

AIMSA continues to attract submissions from all over the world, with submissions from around 35 countries. The range of topics is almost equally broad, from traditional areas such as computer vision and natural language processing to emerging areas such as mining the behavior of Web-based communities. I enjoyed hearing the mix of well-grounded work extending our understanding of areas that have been represented at AIMSAs over the past 30 years with research in areas that did not even exist at the time of the last AIMSAs two years ago. It is good to know that the discipline is still broadening the range of areas that it includes at the same time as cementing the work that has already been done in its various established subfields.

The Programme Committee selected just over 30% of the submissions as long papers, with a further seven accepted as short papers. I am extremely grateful to the Programme Committee, who reviewed the submissions thoroughly, fairly, and very quickly.

July 2012

Allan Ramsay

# Organization

## Program Committee

Gennady Agre	Institute of Information Technologies - BAS, Bulgaria
Galja Angelova	Bulgarian Academy of Sciences, Bulgaria
Annalisa Appice	University Aldo Moro of Bari, Italy
Franz Baader	TU Dresden, Germany
Roman Bartak	Charles University, Czech Republic
Brandon Bennett	University of Leeds, UK
Gavin Brown	University of Manchester, UK
Stefano Cerri	LIRMM: University of Montpellier and CNRS, France
Silvia Coradeschi	Örebro University, Sweden
Ulises Cortès	Universitat Politècnica de Catalunya, Spain
Dan Cristea	Faculty of Computer Science Iasi, Romania
Christo Dichev	Winston-Salem State University, USA
Darina Dicheva	Winston-Salem State University, USA
Vania Dimitrova	University of Leeds, UK
Danail Dochev	Institute of Information Technologies - BAS, Bulgaria
Ben Duboulay	University of Sussex, UK
Stefan Edelkamp	University of Bremen, Germany
Floriana Esposito	Università di Bari, Italy
Vladimir Fomichov	National Research University, Moscow, Russia
David Hogg	University of Leeds, UK
Vladimir Khoroshevsky	Computer Centre Russian Academy of Sciences, Russia
Igor Kononenko	Faculty of Computer and Information Science, Ljubljana, Slovenia
Irena Koprinska	The University of Sydney, Australia
Oliver Kutz	University of Bremen, SFB/TR 8 Spatial Cognition, Germany
Derek Long	King's College London, UK
Zdravko Markov	Central Connecticut State University, USA
Tim Morris	University of Manchester, UK
Radoslav Pavlov	Bulgarian Academy of Sciences, Bulgaria
Steve Pettifer	University of Manchester, UK
Enric Plaza	IIIA-CSIC, Bellaterra, Spain
Ian Pratt-Hartmann	University of Manchester, UK
Allan Ramsay	University of Manchester, UK

## VIII Organization

Ioannis Refanidis	University of Macedonia, Greece
Marco Rospocher	Fondazione Bruno Kessler, Italy
Francesca Rossi	University of Padova, Italy
Paolo Rosso	Polytechnic University Valencia, Spain
Evgeni Smirnov	MICC-IKAT, Maastricht University, The Netherlands
Doina Tatar	University Babes-Bolyai, Romania
Annette ten Teije	Vrije Universiteit Amsterdam, The Netherlands
Dan Tufis	Academia Romana, Romania
Xiaojun Zeng	University of Manchester, UK

## Additional Reviewers

Bejar, Javier	Mititelu, Verginica
Bensalem, Imene	Monov, Vladimir
Di Mauro, Nicola	Mossakowski, Till
Drole, Miha	Outahajala, Mohamed
Esposito, Gennaro	Pevec, Darko
Fanizzi, Nicola	Pičulin, Matej
Fomichov, Vladimir	Sattler, Ulrike
Koprinkova-Hristova, Petia	Stefanescu, Dan
Kormushev, Petar	Szlavik, Zoltan
Loglisci, Corrado	Taranto, Claudio
Martin, Mario	Vračar, Petar
Mencar, Corrado	

# Table of Contents

## Natural Language Processing

Experiments with Filtered Detection of Similar Academic Papers .....	1
<i>Yaakov HaCohen-Kerner and Aharon Tayeb</i>	
Corpus-Based Semantic Filtering in Discovering Derivational Relations .....	14
<i>Maciej Piasecki, Radosław Ramocki, and Paweł Minda</i>	
Lexical Activation Area Attachment Algorithm for Wordnet Expansion .....	23
<i>Maciej Piasecki, Roman Kurc, Radosław Ramocki, and Bartosz Broda</i>	
Metaheuristics for Tuning Model Parameters in Two Natural Language Processing Applications .....	32
<i>Łukasz Kłyk, Paweł B. Myszkowski, Bartosz Broda, Maciej Piasecki, and David Urbansky</i>	
Clustering a Very Large Number of Textual Unstructured Customers' Reviews in English .....	38
<i>Jan Žížka, Karel Burda, and František Dařena</i>	
A Workbench for Temporal Event Information Extraction from Patient Records .....	48
<i>Svetla Boytcheva and Galia Angelova</i>	
Analyzing Emotional States Induced by News Articles with Latent Semantic Analysis .....	59
<i>Diana Lupan, Mihai Dascălu, Ștefan Trăusan-Matu, and Philippe Dessus</i>	

## Social Networks

The Emergence of Cultural Hierarchical Social Networks in Complex Environments .....	69
<i>Mostafa Z. Ali and Robert G. Reynolds</i>	
Ensuring Relevant and Serendipitous Information Flow in Decentralized Online Social Network .....	79
<i>Udeep Tandukar and Julita Vassileva</i>	

## Knowledge Representation and Reasoning

Ontology-Based Information and Event Extraction for Business Intelligence . . . . .	89
<i>Ernest Arendarenko and Tuomo Kakkonen</i>	
Modelling Highly Symmetrical Molecules: Linking Ontologies and Graphs . . . . .	103
<i>Oliver Kutz, Janna Hastings, and Till Mossakowski</i>	
Personalizing and Improving Tag-Based Search in Folksonomies . . . . .	112
<i>Samia Beldjoudi, Hassina Seridi-Bouchelaghem, and Catherine Faron-Zucker</i>	
Views and Synthesis of Cognitive Maps . . . . .	119
<i>Aymeric Le Dorze, Lionel Chauvin, Laurent Garcia, David Genest, and Stéphane Loiseau</i>	
Identification of the Compound Subjective Rule Interestingness Measure for Rule-Based Functional Description of Genes . . . . .	125
<i>Aleksandra Gruca and Marek Sikora</i>	
Automatic Generation and Learning of Finite-State Controllers . . . . .	135
<i>Matteo Leonetti, Luca Iocchi, and Fabio Patrizi</i>	
FactForge: Data Service or the Diversity of Inferred Knowledge over LOD . . . . .	145
<i>Mariana Damova, Kiril Simov, Zdravko Tashev, and Atanas Kiryakov</i>	
From Path-Consistency to Global Consistency in Temporal Qualitative Constraint Networks . . . . .	152
<i>Nouhad Amaneddine and Jean-François Condotta</i>	

## Machine Learning

Rule Quality Measure-Based Induction of Unordered Sets of Regression Rules . . . . .	162
<i>Marek Sikora, Adam Skowron, and Lukasz Wróbel</i>	
Swarm Capability of Finding Eigenvalues . . . . .	172
<i>Jacek Kabziński</i>	
A Study on the Utility of Parametric Uniform Crossover for Adaptation of Crossover Operator . . . . .	178
<i>Farhad Nadi and Ahamad Tajudin Khader</i>	
Decomposition, Merging, and Refinement Approach to Boost Inductive Logic Programming Algorithms . . . . .	184
<i>Andrej Chovanec and Roman Barták</i>	

Learning Teamwork Behaviors Approach: Learning by Observation Meets Case-Based Planning .....	195
<i>Banafsheh Rekabdar, Bita Shadgar, and Alireza Osareh</i>	
Evolutionary Approach to Multiobjective Optimization of Portfolios That Reflect the Behaviour of Investment Funds .....	202
<i>Krzysztof Michalak, Patryk Filipiak, and Piotr Lipiński</i>	
Recurrent Neural Identification and an I-Term Direct Adaptive Control of Nonlinear Oscillatory Plant .....	212
<i>Ieroham Baruch, Sergio M. Hernandez, Jacob Moreno-Cruz, and Elena Gortcheva</i>	
Towards Automatic Structure Analysis of Digital Musical Content .....	223
<i>Adrian Simion and Stefan Trăușan-Matu</i>	
<b>Planning and Agents</b>	
Methodology for a New Agent Architecture Based on the MVC Pattern .....	230
<i>Yassine Gangat, Denis Payet, and Rémy Courdier</i>	
An Improved Method of Action Recognition Based on Sparse Spatio-temporal Features .....	240
<i>Junwei Zhu, Jin Qi, and Xiangbin Kong</i>	
Echelon Platoon Organisation: A Distributed Approach Based on 2-Spring Virtual Links .....	246
<i>Madeleine El Zaher, Jean-Michel Contet, Franck Gechter, and Abderrafiaa Koukam</i>	
Multi-agent Task Division Learning in Hide-and-Seek Games .....	256
<i>Mohamed K. Gunady, Walid Gomaa, and Ikuo Takeuchi</i>	
Modeling Actions Based on a Situative Space Model for Recognizing Human Activities .....	266
<i>Juan Carlos Nieves, Dipak Surie, and Helena Lindgren</i>	
Management of Urban Parking: An Agent-Based Approach .....	276
<i>Nesrine Bessghaier, Mahdi Zargayouna, and Flavien Balbo</i>	
Modeling Smart Home Using the Paradigm of Nets within Nets .....	286
<i>Yacine Kissoum, Ramdane Maamri, and Zaidi Sahnoun</i>	
<b>Search</b>	
A Formal Model for Constructing Semantic Expansions of the Search Requests about the Achievements and Failures .....	296
<i>Vladimir A. Fomichov and Anton V. Kirillov</i>	

Parallel CHC Algorithm for Solving Dynamic Traveling Salesman Problem Using Many-Core GPU . . . . .	305
<i>Patryk Filipiak and Piotr Lipiński</i>	
<b>Computer Vision</b>	
Modelling and Recognition of Signed Expressions Using Subunits Obtained by Data–Driven Approach . . . . .	315
<i>Mariusz Oszust and Marian Wysocki</i>	
Biometrics from Gait Using Feature Value Method . . . . .	325
<i>Tianxiang Zhang and Gentiane Venture</i>	
<b>Author Index</b> . . . . .	335

# Experiments with Filtered Detection of Similar Academic Papers

Yaakov HaCohen-Kerner and Aharon Tayeb

Dept. of Computer Science, Jerusalem College of Technology, 91160 Jerusalem, Israel  
[kerner@jct.ac.il](mailto:kerner@jct.ac.il), [aharontayeb@gmail.com](mailto:aharontayeb@gmail.com)

**Abstract.** In this research, we investigate the issue of efficient detection of similar academic papers. Given a specific paper, and a corpus of academic papers, most of the papers from the corpus are filtered out using a fast filter method. Then, 47 methods (baseline methods and combinations of them) are applied to detect similar papers, where 34 of the methods are variants of new methods. These 34 methods are divided into three new method sets: rare words, combinations of at least two methods, and compare methods between portions of the papers. Results achieved by some of the 34 heuristic methods are better than the results of previous heuristic methods, comparing to the results of the "Full Fingerprint" (FF) method, an expensive method that served as an expert. Nevertheless, the run time of the new methods is much more efficient than the run time of the FF method. The most interesting finding is a method called CWA(1) that computes the frequency of rare words that appear only once in both compared papers. This method has been found as an efficient measure to check whether two papers are similar.

**Keywords:** Corpus, Detection, Filtering, Fingerprinting, Heuristic methods, Similar academic papers.

## 1 Introduction

A wide range of research carried out in the field of detection of plagiarism in general and detection of similar papers in particular. Plagiarism has been defined as "the taking and using as one's own of the thoughts, writings, or inventions of another<sup>1</sup>". Loui [1] explains that plagiarism existed already in the olden days. Authors used sentences, concepts, ideas, etc. without citing the original authors. Martin [2] identified various levels of plagiarism. The two highest levels are word-for-word plagiarism and paraphrasing plagiarism. Ceska [3] claims that most authors that copy parts of papers do not try to hide it. Authors who try to hide their plagiarism usually replace words by suitable synonyms in order to break up the continuous copied sentences.

The policy for ACM journals and transactions is that "the submitted manuscript must contain at least 25% new content material (i.e., material that offers new insights,

---

<sup>1</sup> The Shorter Oxford English Dictionary of Historical Principles. Oxford, Oxford University Press, 1973.

new results, etc.)<sup>2</sup>. The IEEE policy<sup>3,4</sup> is that "authors should only submit original work that has neither appeared elsewhere for publication, nor which is under review for another refereed publication. If authors have used their own previously published work(s) as a basis for a new submission, they are required to cite the previous work(s) and very briefly indicate how the new submission offers substantial novel contributions beyond those of the previously published work(s)."

Various systems were developed to automatically identify similar documents (e.g., Schleimer et al. [4], Collberg et al. [5], Sorokina et al. [6], Keuskamp and Sliuzas [7], and HaCohen-Kerner et al. [8]). Furthermore, systems that identify similar computer programs were also developed (e.g., Wise [9], Burrows et al. [10], Chen et al. [11] and Jadalla and Elnagar [12]).

The Internet contains millions of articles in various fields, so it is necessary to develop quick and efficient software to discover articles similar to a tested article.

There are two main approaches to detect similar documents: ranking and fingerprinting. Ranking methods (e.g., the cosine measure, the inner product, and the normalized inner product) belong to the information retrieval domain. Fingerprint methods compare between documents based on their fingerprints. In this research, we deal with fingerprinting methods.

A full fingerprint (FF) of a document is a group of all its possible sequential substrings of length  $\alpha$  in words (or characters). There are  $N-\alpha+1$  such substrings, where  $N$  is the length of the document in words (or characters). This method selects overlapping sub-strings. Comparing between a tested paper ( $T$ ) and a retrieved paper ( $R$ ), where  $T$ 's size is  $|T|$  and  $n$  is the number of substrings common to both documents can be calculated by  $n/|T|$ , which is the measure of how much of  $T$  is contained in  $R$ .

Fingerprint methods have been applied by many previous researches, e.g., Manber [13], Heintze [14], Shivakumar and Garcia-Molina [15], Broder [16], Lyon et al. [17], and Hoad and Zobel [18].

There are various versions of selective fingerprints (SF) that reduce the size of a FF in order to reduce the run time of the comparison algorithm. The most simple kind of a SF is the "All substrings selection" described in (Hoad and Zobel [18]). In contrast to the FF method, this method does not select overlapping substrings. Rather, it chooses all non-overlapping substrings of size  $\alpha$  from the document.

Two other selective fingerprint methods: CA and CR were proposed and applied in HaCohen-Kerner et al. [8]. CA compares between the abstracts of  $T$  and  $R$  using FF(3), and CR calculates the number of references that are common to  $T$  and  $R$  divided by the number of the references in  $T$ .

Additional kinds of selective fingerprint methods are as follows. Hoad and Zobel [18] suggested positional, frequency-based (e.g., The Rarest-in-Document method), and structure-based methods. Monostori et al. [19] defined a measure called

<sup>2</sup> [http://www.ieee.org/publications\\_standards/publications/rights/ID\\_Plagiarism.html](http://www.ieee.org/publications_standards/publications/rights/ID_Plagiarism.html)

<sup>3</sup> [http://www.ieee.org/publications\\_standards/publications/rights/ID\\_Plagiarism.html](http://www.ieee.org/publications_standards/publications/rights/ID_Plagiarism.html)

<sup>4</sup> [http://dl.comsoc.org/livepubs/ci1/info/Self\\_Plagiarism.pdf](http://dl.comsoc.org/livepubs/ci1/info/Self_Plagiarism.pdf)

Symmetric Similarity (SS), Bernstein and Zobel [20] defined a few additional similarity measures, such as S2 and S3. Forman et al. [21] applied a content-based chunking, method as presented in Muthitacharoen et al. [22]. This method breaks a file into a sequence of chunks so that chunk boundaries are determined by the local contents of the file in contrast to using fixed size chunks.

This paper is organized as follows: Section 2 describes 47 methods that were applied to detect similar papers. These methods are divided into 7 method sets. Section 3 presents the experiments that have been performed and their analysis. Section 4 details an illustrative example. Section 5 concludes and proposes future directions for research.

## 2 Methods That Detect Similar Papers

In HaCohen-Kerner et al. [8], 28 methods have been applied to detect similar papers. These 28 methods were divided into four method sets: full fingerprint methods, selective fingerprint methods, anchor methods, and combinations of two methods. These methods were activated without any preceding filter method.

In the current research, the detection process contains the following two stages for each tested document: (1) filter out non-similar documents, and (2) apply all baseline methods and combinations of methods to detect similar documents.

CR was chosen as the filter method since its constant order of run time. CR calculates the number of references that are common to T and R divided by the number of the references in T. The chosen filter threshold was 25%. That is to say, for each tested document only documents with a CR value of 25% or above will remain for the second stage. All other documents assumed to be non-similar, and therefore they are not selected for the next stage.

Since most of the papers in the discussed corpus (see Section 3) include less than 50 papers mentioned in the reference section (the average number of references for each paper in the whole corpus is only 12.87), we can use 50 as a constant upper bound for the number of references in a paper, without depending on the number of words in the paper.

The relative success of CR in HaCohen-Kerner et al. [8] shows that many document pairs that are suspected as similar papers include similar lists of references. For example, if the tested paper is an extended conference paper, which contains 15 papers in its references' section and the retrieved paper is a poster paper that contains only 5 papers in its references section. Assuming that 4 out of the 15 papers are among the 5 papers in the retrieved paper, we have a CR-value of 26.67% (4/15), which is above the filtering threshold. If the tested paper was the poster paper and the extended conference paper was the retrieved paper, then we will have a CR-value of 80% (4/5).

Below, we present the 47 implemented methods. The first 13 methods are part of the 28 methods that were implemented in HaCohen-Kerner et al. [8]. The last 34 methods, which are variants of new methods, are divided into three new method sets: Rare words, Combinations of at least two methods, and Compare methods between portions of the papers.

## Full Fingerprint Methods

Various full fingerprint methods have been applied in HaCohen-Kerner et al. [8]. These methods deal with overlapping substrings with length of 3 words from the beginning of the document. In the current research, we applied the standard FF(3). In addition, we apply the CA method, which was one of the novelties presented in [8]. All other full fingerprint methods have not been applied, since they were less successful than FF.

1. FF(3) - Full Fingerprints with length of 3 words.
2. CA - Compares between the Abstracts of the two documents using FF(3).

All the selective fingerprint methods that were applied in [8] have not been applied in this research, since they were found as less successful than FF.

## Anchor Methods

For the same reasons (unsuccessful results), we did not apply any anchor method, except for the CR method.

3. CR - CompareReferences. This method compares between the titles of the papers included in the references section of the two examined papers.

As shown in HaCohen-Kerner et al. [8], the anchor methods were unsuccessful, probably because they use a small portion of data. Therefore, also combinations of two methods have been defined and implemented. Most of these methods work on only one of the three portions of the paper: the first third (first), the middle third (middle), and the last third (end) of the paper according to the number of the words in the tested paper. All the first, middle and end methods use FF(3). These methods were combined with CA or CR. CA was not combined with the first methods because the abstract is included in the first part of the paper. CR was not combined with the last methods because the references are included in the end part of the paper.

## Combinations of CA and CR

4. CARA- CompareAbstractReferencesAverage. This method calculates the average value of CA and CR.
5. CARM - CompareAbstractReferencesMin. This method calculates the minimal value between CA and CR.

## Combinations of Two Methods (One of Them Deals with One Portion of the Paper)

6. CAMA - CompareAbstractMiddleAve. This method calculates the average value of CA and FF(3) computed for the middle parts of the two examined papers.
7. CAMM - CompareAbstractMiddleMin. This method calculates the minimal value between CA and FF(3) computed for the middle parts of the two examined papers.
8. CAEA - CompareAbstractEndAverage. This method calculates the average value of CA and FF(3) computed for the end parts of the two examined papers.

9. CAEM - CompareAbstractEndMin. This method calculates the minimal value between CA and FF(3) computed for the end parts of the two examined papers.
10. CRFA - CompareReferencesFirstAverage. This method calculates the average value of CR and FF(3) computed for the first parts of the two examined papers.
11. CRFM - CompareReferencesFirstMin. This method calculates the minimal value between CR and FF(3) computed for the first parts of the two examined papers.
12. CRMA - CompareReferencesMiddleAverage. This method calculates the average value of CR and FF(3) computed for the middle parts of the two examined papers.
13. CRMM - CompareReferencesMiddleMin. This method calculates the minimal value between CR and FF(3) computed for the middle parts of the two examined papers.

The three next method sets contain only novel methods that were implemented for the current research.

## Rare Words

14-19. CWA(N) - Calculates the relative frequency of Words that Appear at least once and no more than N (1, 2, 5, 10, 15, 20) times in both T and R.

## Combinations of at Least Two Methods

20. CTAAA - Average value of the Comparisons of Titles, Authors, and CA.
21. CTAAM - Minimal value of the Comparisons of Titles, Authors, and CA.
22. CTARA - Average value of the Comparisons of Titles, Abstract, and References name
23. CTARM - Minimal value of the Comparisons of Titles, Abstract, and References name
24. CTARAA - Average value of the Comparisons of Titles, Authors, CR and CA.
25. CTARAM - Minimal value of the Comparisons of Titles, Authors, CR and CA.
26. CTARAFA - Average value of the Comparisons of Titles, Authors, CR, CA and first third of the papers.
27. CTARAFM - Minimal value of the Comparisons of Titles, Authors, CR, CA and first third of the papers.
28. CTARAMA - Average value of the Comparisons of Titles, Authors, CR, CA and middle third of the papers.
29. CTARAMM - Minimal value of the Comparisons of Titles, Authors, CR, CA and middle third of the papers.
30. CTARALA - Average value of the Comparisons of Titles, Authors, CR, CA and last third of the papers.
31. MCAR - Maximal value of the comparisons of Abstracts and References.
32. MCAF - Maximal value of the comparisons of Abstracts and First parts.
33. MCAM - Maximal value of the comparisons of Abstracts and Middle parts.
34. MCAE - Maximal value of the comparisons of Abstracts and End parts.
35. MCRF - Maximal value of the comparisons of References, and First parts.
36. MCRM - Maximal value of the comparisons of References and Middle parts.
37. MCRE - Maximal value of the comparisons of References and End parts.

38. MCARF - Maximal value of the comparisons of Abstracts, References and First parts.
39. MCARM - Maximal value of the comparisons of Abstracts, References, and Middle parts.
40. MCARE - Maximal value of the comparisons of Abstracts, References, and End parts.
41. CARW\_4\_6 - A weighted average of the comparisons of Abstracts and References with a weight of 0.4 assigned to the Abstracts comparison and a weight of 0.6 assigned to the references' comparison.
42. CARW\_3\_7 - A weighted average of the comparisons of Abstracts and References with a weight of 0.3 assigned to the Abstracts comparison and a weight of 0.7 assigned to the references' comparison.
43. CARW\_6\_4 - A weighted average of the comparisons of Abstracts and References with a weight of 0.6 assigned to the Abstracts comparison and a weight of 0.4 assigned to the references' comparison.
44. CARW\_7\_3 - A weighted average of the comparisons of Abstracts and References with a weight of 0.7 assigned to the Abstracts comparison and a weight of 0.3 assigned to the references' comparison.

### **Compare Methods between Portions of the Papers**

In addition, we define and apply the First, Middle and Last methods. All these methods use FF(3).

45. First - Compare between the First thirds of the two documents using FF(3).
46. Middle - Compare between the Middle thirds of the two documents using FF(3).
47. End - Compare between the Last thirds of the two documents using FF(3).

## **3 Experimental Results**

The examined corpus includes 10,100 academic papers in computer science containing about 53M words. Most of the documents are papers related to NLP and are from the last dozen years. Most of the documents were downloaded from <http://www.aclweb.org/anthology/>. The test set includes 1,500 documents that were randomly selected from the corpus. For each tested document, all the other 10,099 documents were compared using the various methods.

At the first stage of our algorithm, the filtering stage, on average, 10,097.9 documents were filtered out for each tested document. That is to say, for every tested document, on average, about only one document (out of 10,099 documents) was left for the second stage. The median value of the number of the documents that were filtered out was 10,040 for each tested document. The minimal and maximal numbers of documents that were filter out for one tested document were 9,363 and 10,099 documents, respectively. The standard deviation was 13.37, which is rather reasonable.

Table 1 presents the results of the 47 implemented methods. The first left column indicates an ordinal number. The second left column presents the method name. The IDN, VHS, HS, MS columns present respectively the number of the document pairs found as identical, very high similar, high similar, and medium similar relatively to

the 1,500 tested documents. Document pairs with the following similarity values: [96%, 100%]<sup>5</sup>, [80%, 96%), [60%, 80%), and [40%, 60%), got the IDN, VHS, HS, MS levels, respectively.

According to sample checks made in HaCohen-Kerner et al. [8], FF(3) has been discovered as the best detection method among the examined methods. FF(3) detects all simple types of similar papers that were defined in [8], as papers that include at least 40% of identical overlapping sub-strings with length of 3 words. In this research, we refer to the FF(3)'s results as an expert's results.

**Table 1.** Results of the 47 implemented methods for 1,500 tested papers

	Method	IDN	VHS	HS	MS	#	Method	IDN	VHS	HS	MS
1	FF(3)	64	21	16	38	25	CTARAM	24	4	0	2
2	CA	45	17	11	12	26	CTARAFA	26	14	24	83
3	CR	92	42	181	1520	27	CTARAFM	24	4	0	1
4	CARA	39	19	22	303	28	CTARAMA	26	14	24	83
5	CARM	37	15	16	15	29	CTARAMM	24	4	0	1
6	CAMA	42	14	12	27	30	CTARAEA	26	2	33	129
7	CAMM	40	11	12	13	31	MCAR	93	41	146	1328
8	CAEA	42	14	12	27	32	MCAF	49	23	12	36
9	CAEM	40	11	12	13	33	MCAM	49	23	12	36
10	CRFA	44	23	32	343	34	MCAE	49	23	12	36
11	CRFM	43	18	13	40	35	MCRF	97	44	181	1519
12	CRMA	44	23	33	342	36	MCRM	97	44	181	1519
13	CRMM	43	18	13	40	37	MCRE	97	44	181	1519
14	CWA(1)	36	14	24	57	38	MCARF	94	44	146	1327
15	CWA(2)	41	17	29	120	39	MCARM	94	44	146	1327
16	CWA(5)	42	28	58	387	40	MCARE	94	44	146	1327
17	CWA(10)	45	41	71	1044	41	CARW_4_6	40	18	274	148
18	CWA(15)	45	52	74	1561	42	CARW_3_7	39	19	286	193
19	CWA(20)	45	53	93	1882	43	CARW_6_4	40	20	15	278
20	CTAAA	27	1	32	98	44	CARW_7_3	40	21	13	26
21	CTAAM	26	2	0	2	45	FIRST	48	20	13	39
22	CTARA	26	5	84	325	46	MIDDLE	44	23	12	19
23	CTARM	25	4	1	23	47	END	47	12	17	15
24	CTARAA	26	2	33	129						

Due to the size of the corpus, the number of tested papers and the number of methods, the results of all other methods were automatically evaluated according to the results of the FF method. The error rates of all other methods are calculated based on their results and their deviation from the results of FF.

Out of a sample of 1,500 tested papers, FF reports on 139 papers (i.e., about 9.3% of 10,099 papers) that are similar to papers included in the data set. These 139 papers are composed of 64 IDN, 21 VHS, 16 HS, and 38 MS papers. None of these 139 papers was a case of “pure” plagiarism detection (with totally different authors). That is to mean, each one of these pairs of similar papers contains in most cases the same list of authors while in a few cases there was at least one author in common.

To evaluate the various methods, we have defined two kinds of error measures: ERROR-1 and ERROR-2. Tables 2 and 3 present respectively the ERROR-1 and

<sup>5</sup> Based on our experiments, IDN has been defined from 96% and above because sometimes the same paper is stored in two places in slightly different versions.

ERROR-2 analysis (in %) of the 47 chosen methods for 1,500 tested papers. The TP, FP, FN, TN, and FP+FN columns present respectively the values in % of the True Positive, False Positive, False Negative, True Negative, and the sum of FP and FN.

ERROR-1 is an aggravating error measure while ERROR-2 is a mitigating error measure. For example, if FF identifies a tested document pair as HS, and CA identifies the same tested pair as MS, then for ERROR-1 it is regarded as a FP (since their results are not at the same similarity level), while for ERROR-2 it is regarded as a TP (since both methods identify the same pair as similar).

The ERROR-1 results (shown in Table 2) are defined as follows. TP is discovered when a certain method (any method except for FF) identifies a document pair with a certain similarity level (MS or above) and FF identifies the same document pair with the same similarity level. FP is discovered when a certain method identifies a document pair at a higher similarity level than the similarity level identified by FF. FN is discovered when a certain method identifies a document pair at a lower similarity level than the similarity level identified by FF. TN is discovered when a certain method didn't identify a document pair as similar (i.e., less than MS) while also FF didn't identify this pair as similar.

The ERROR-2 results (shown in Table 3) are defined as follows. TP is discovered when both FF and the discussed method (any method except for FF) identify a certain document pair at any level of similarity (even they are not the same level). FP is discovered when a certain method identifies a document pair as similar (MS or above) while FF didn't identify this pair as similar. FN is discovered when a certain method didn't identify a document pair as similar while FF identifies this pair as similar. TN is defined exactly as it is defined for ERROR-1.

**Table 2.** ERROR-1 analysis (in %) of the 47 implemented methods for 1,500 tested papers

#	Method	TP	FP	FN	TN	FP+FN	#	Method	TP	FP	FN	TN	FP+FN	
1	FF(3)	2.658	0.000	0.000	97.342	0.000	25	CTARAM	0.575	2.047	0.054	97.324	2.101	
2	CA	1.203	1.149	0.413	97.234	1.563	26	CTARAFIA	0.988	1.563	1.060	96.390	2.622	
3	CR	1.545	0.467	34.950	63.039	35.417	27	CTARAFM	0.575	2.047	0.036	97.342	2.083	
4	CARA	1.293	1.078	5.172	92.457	6.250	28	CTARAMA	0.988	1.563	1.060	96.390	2.622	
5	CARM	1.096	1.347	0.323	97.234	1.670	29	CTARAMM	0.575	2.047	0.036	97.342	2.083	
6	CAMA	1.419	1.096	0.251	97.234	1.347	30	CTARAEA	0.970	1.563	2.047	95.420	3.610	
7	CAMM	1.275	1.311	0.090	97.324	1.401	31	MCAR	1.545	0.449	31.017	66.990	31.466	
8	CAEA	1.419	1.096	0.251	97.234	1.347	32	MCAF	1.437	0.790	0.754	97.019	1.545	
9	CAEM	1.275	1.311	0.090	97.324	1.401	33	MCAM	1.437	0.790	0.754	97.019	1.545	
10	CRFA	1.742	0.647	5.891	91.721	6.537	34	MCAE	1.437	0.790	0.754	97.019	1.545	
11	CRFM	1.473	1.096	0.341	97.091	1.437	35	MCRF	1.706	0.198	35.075	63.021	35.273	
12	CRMA	1.724	0.647	5.909	91.721	6.555	36	MCRM	1.706	0.198	35.075	63.021	35.273	
13	CRMM	1.473	1.096	0.341	97.091	1.437	37	MCRE	1.706	0.198	35.075	63.021	35.273	
14	CWA(1)	1.509	1.149	0.180	97.162	1.329	38	MCARF	1.580	0.323	31.106	66.990	31.430	
15	CWA(2)	1.868	0.665	1.401	96.067	2.065	39	MCARM	1.580	0.323	31.106	66.990	31.430	
16	CWA(5)	1.580	0.467	7.381	90.571	7.348	40	MCARE	1.580	0.323	31.106	66.990	31.430	
17	CWA(10)	1.473	0.359	19.917	78.251	20.277	41	CARW_4_6	1.293	1.024	6.879	90.805	7.902	
18	CWA(15)	1.383	0.359	29.544	68.714	29.903	42	CARW_3_7	1.257	1.024	7.884	89.835	8.908	
19	CWA(20)	1.383	0.359	35.668	62.590	36.027	43	CARW_6_4	1.293	1.096	4.705	92.906	5.801	
20	CTAAA	0.844	1.706	1.688	95.761	3.394	44	CARW_7_3	1.311	1.078	0.557	97.055	1.634	
21	CTAACM	0.611	2.011	0.054	97.324	2.065	45	FIRST	1.634	0.826	0.467	97.073	1.293	
22	CTARA	0.898	1.563	6.537	91.002	8.100	46	MIDDLE	1.437	1.185	0.054	97.324	1.239	
23	CTARM	0.611	1.994	0.449	96.947	2.443	47	END		1.329	1.185	0.144	97.342	1.329
24	CTARAA	0.970	1.563	2.047	95.420	3.610								

**Table 3.** ERROR-2 analysis (in %) of the 47 implemented methods for 1,500 tested papers

#	Method	TP	FP	FN	TN	FP+FN	#	Method	TP	FP	FN	TN	FP+FN
1	FF(3)	2.658	0.000	0.000	97.342	0.000	25	CTARAM	0.647	0.018	2.011	97.324	2.029
2	CA	1.580	0.108	1.078	97.234	1.185	26	CTARAF	1.814	0.952	0.844	96.390	1.796
3	CR	2.532	34.303	0.126	63.039	34.429	27	CTARAFM	0.647	0.000	2.011	97.342	2.011
4	CARA	2.155	4.885	0.503	92.457	5.388	28	CTARAMA	1.814	0.952	0.844	96.390	1.796
5	CARM	1.545	0.108	1.114	97.234	1.221	29	CTARAMM	0.647	0.000	2.011	97.342	2.011
6	CAMA	1.760	0.108	0.898	97.234	1.006	30	CTARAEA	1.634	1.922	1.024	95.420	2.945
7	CAMM	1.509	0.018	1.149	97.324	1.167	31	MCAR	2.389	30.352	0.269	66.990	30.621
8	CAEA	1.760	0.108	0.898	97.234	1.006	32	MCAP	1.994	0.323	0.665	97.019	0.988
9	CAEM	1.509	0.018	1.149	97.324	1.167	33	MCAM	1.994	0.323	0.665	97.019	0.988
10	CRFA	2.496	5.621	0.162	91.721	5.783	34	MCAE	1.994	0.323	0.665	97.019	0.988
11	CRFM	1.958	0.251	0.700	97.091	0.952	35	MCRF	2.622	34.321	0.036	63.021	34.357
12	CRMA	2.496	5.621	0.162	91.721	5.783	36	MCRM	2.622	34.321	0.036	63.021	34.357
13	CRMM	1.958	0.251	0.700	97.091	0.952	37	MCRE	2.622	34.321	0.036	63.021	34.357
14	CWA(1)	2.335	0.180	0.323	97.162	0.503	38	MCARF	2.443	30.352	0.216	66.990	30.568
15	CWA(2)	2.604	1.275	0.054	96.067	1.329	39	MCARM	2.443	30.352	0.216	66.990	30.568
16	CWA(5)	2.658	6.771	0.000	90.571	6.771	40	MCARE	2.443	30.352	0.216	66.990	30.568
17	CWA(10)	2.658	19.091	0.000	78.251	19.091	41	CARW_4_6	2.245	6.537	0.413	90.805	6.950
18	CWA(15)	2.658	28.628	0.000	68.714	28.628	42	CARW_3_7	2.299	7.507	0.359	89.835	7.866
19	CWA(20)	2.658	34.752	0.000	62.590	34.752	43	CARW_6_4	2.065	4.436	0.593	92.906	5.029
20	CTAAA	1.383	1.580	1.275	95.761	2.856	44	CARW_7_3	1.670	0.287	0.988	97.055	1.275
21	CTAAM	0.647	0.018	2.011	97.324	2.029	45	FIRST	2.047	0.269	0.611	97.073	0.880
22	CTARA	1.706	6.340	0.952	91.002	7.292	46	MIDDLE	1.886	0.018	0.772	97.324	0.790
23	CTARM	0.700	0.395	1.958	96.947	2.353	47	END	1.832	0.000	0.826	97.342	0.826
24	CTARAA	1.634	1.922	1.024	95.420	2.945							

From the first glance, the results of FF are rather different from those of all other methods. Nevertheless, we have to take into consideration that for each tested paper, there is an agreement that most documents of the corpus are TN. These findings can be easily seen in Tables 2 and 3. There is a consent between FF and most other methods about TN and also TP. For about the ten best methods, the sum of FP and FN is only about between 0.5% and 1.5%. That is, the success rate of the top ten methods is around 99%.

As it was expected, the corresponding FP and FN values of the ERROR-1 and ERROR-2 rates were aggravating and mitigating, respectively. That is to say, the FP and FN values of ERROR-1 for each method (except for FF) were greater than the corresponding FP and FN values of ERROR-2. Nevertheless, the ERROR-1 and ERROR-2 values are rather similar with slight changes.

According to FP+FN (the overall error) the best 4 heuristic methods compared to FF are CWA(1), FIRST, MIDDLE, and END. The run time of all these methods are significantly lower than the run time of FF. The ERROR-1 and ERROR-2 rates of CWA(1), FIRST, MIDDLE, and END were about (1.3% and 0.5%), (1.3% and 0.9%), (1.2% and 0.8%), and (1.3% and 0.8%), respectively. According to ERROR-1's rates, the best method is MIDDLE. According to ERROR-2's rates, the best method is CWA(1).

The success of FIRST, MIDDLE, and END is not surprising since these methods are variants of FF(3), which is regarded as our expert. Nevertheless, the success of CWA(1) is the most interesting finding. CWA(1) calculates the relative frequency of rare words that appear only once in both T and R. This method has been found as a very good measure (with a success rate of about 99%) to check whether two papers are similar.

The results of this research are in contrast to the results presented by HaCohen-Kerner et al. [8], where CA and CR have been the best heuristic methods, in this research CA and especially CR, were not so successful.

The ERROR-1 and ERROR-2 rates of CA were about 1.6% and 1.2%, respectively. These errors are quite low, but still they are worse than those of the best methods. CA discovers only 45 IDN, 17 VHS, 11 HS, and 12 MS paper pairs. The meaning of this finding is that the abstracts of many similar paper pairs were significantly changed although the rest of the two papers in each paper pair are rather similar. The ERROR-1 and ERROR-2 rates of CR were about 35.4% and 34.4%, respectively. Due to these big errors rates, CR is one of the worst methods.

CR identified 92 documents as IDN, many more than the 64 documents identified as IDN by FF. This is because there were 28 additional tested papers have the same references as in 28 other papers included in the data set, although these 28 paper pairs are different in the other paper's sections. Furthermore, CR reports on relatively high number of suspicious paper pairs, especially at the MS level (1520). The meaning of this finding is that the references in many paper pairs are not significantly different although there are large differences in other sections (e.g., a conference paper and a journal paper).

## 4 Illustrative Example

Table 4 shows a partial comparison between two papers, which their similarity measure was found as HS (High Similar) according to FF(3), the best detection method, with similarity value of 63.50%. The result of CA was only 30 %since the abstracts of T and R are quite different. According to the result of CR (80%) these papers are VHS. The reason for this finding is the fact that 80% of T's references are included in R's references. The results of the best four methods are better from those of CA and CR since they are closer to those of FF(3) as follows: CWA(1) - 56.7%, FIRST - 65.62%, MIDDLE - 62.20%, and END - 28.62%. CWA(1)'s value was 56.7% since out of 462 words that appear only once in T, only 262 of them appear also only once in R.

The result of the END method is quite low probably because most of the changes between the two papers are found in the last thirds of the papers. In contrast to T, which contains only 4 pages, R contains 8 pages that include more one more section, more sub-sections, more sentences and more references. It is important to point out that when we exchange between the two papers (old R becomes T and old T becomes R), the FF(3)'s result is 30.18%, which is below the MS level. That is to say, the papers are not similar from the viewpoint of the new T. An explanation to this interesting finding is that the old T paper, which contains only 4 pages, is the original paper and the old R paper, which contains 8 pages, is a meaningful extension and therefore R is not a plagiarism. When we compare theses papers from the viewpoint of old T the similarity level is HS and when we compare theses papers from the viewpoint of old R (the extended paper) the similarity level is below MS.

The paper on the left side of Table 4 is the tested paper (T) [23]. It contains 4 pages and it was published as a short paper on April 06. The retrieved paper (R) [24] is a regular conference paper that contains 8 pages. It was published on June 06.

The bold color in Table 4 indicates identical sentences or phrases in both papers. All the six words that composed T's title are included in R's title. The authors of both papers are the same and their names appear in the same order. Several phrases or sentences in the abstracts or in the introduction sections in both papers are identical. Titles of several sections, sub-sections, and sentences are identical.

**Table 4.** A partial comparison between two papers, which were found as HS (High Similar)

	Tested paper (T)	Retrieved paper (R)
Title	Semantic Role Labeling for Coreference Resolution	Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution
Full Bibliographic details	Ponzetto, S. P. and Strube, M. Semantic role labeling for coreference resolution. <i>Companion Volume to the Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, 3-7 April 2006</i> , pp. 143-146, short paper.	Ponzetto, S. P. and Strube, M. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. <i>Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, New York, N.Y., 4-9 June 2006</i> , pp. 192-199.
Authors	<b>Simone Paolo Ponzetto and Michael Strube</b>	<b>Simone Paolo Ponzetto and Michael Strube</b>
Abstract	Extending a machine learning based coreference resolution system with a feature capturing automatically generated information about semantic roles improves its performance.	In this paper we present an extension of a machine learning based coreference resolution system ...
Beginning of the Introduction section	The last years have seen a boost of work devoted to the development of machine learning based coreference resolution systems (Soon et al., 2001; Ng & Cardie, 2002; Kehler et al., 2004, inter alia). ...  This paper explores whether coreference resolution can benefit from SRL, more specifically, which phenomena are affected by such information. The motivation comes from the fact that current coreference resolution systems are mostly relying on rather shallow features ...	The last years have seen a boost of work devoted to the development of machine learning based coreference resolution systems (Soon et al., 2001; Ng & Cardie, 2002; Yang et al., 2003; Luo et al., 2004, inter alia) ...  This paper explores whether coreference resolution can benefit from semantic knowledge sources. More specifically, whether a machine learning based approach to coreference resolution can be improved and which phenomena are affected by such information. ...
Titles of several sections, sub-sections, and sentences	2 Coreference Resolution Using SRL 2.1 Corpora Used 2.2 Learning Algorithm 2.3 Baseline System Features  Following Ng & Cardie (2002), our baseline system reimplements the Soon et al. (2001) system. The system uses 12 features. Given a ...  3 Experiments 3.1 Performance Metrics  We report in the following tables the MUC score (Vilain et al., 1995). Scores in Table 2 are computed for all noun phrases appearing in either the key or the system response, ...	3 Coreference Resolution Using Semantic Knowledge Sources 3.1 Corpora Used 3.2 Learning Algorithm 3.3 Baseline System Features  Following Ng & Cardie (2002), our baseline system reimplements the Soon et al. (2001) system. The system uses 12 features. Given a ...  4 Experiments 4.1 Performance Metrics  We report in the following tables the MUC score (Vilain et al., 1995). Scores in Table 2 are computed for all noun phrases appearing in either the key or the system response, ...

## 5 Conclusions and Future Work

In this research, we defined and tested various heuristic methods (most of them are variants of new methods and their combinations) to detect similar papers. Some of the new methods achieve better results than previous heuristic methods. The run time of the new methods is much more efficient than the run time of the FF method. Furthermore, the detection process was accelerated due to the use of a quick filtering method.

The most interesting finding is a method called CWA(1) that computes the frequency of rare words that appear only once in both papers (the tested paper and the retrieved paper). This method has been found as a good measure to check whether two papers are similar. The overall error rate (false positive + false negative) of CWA(1) is about 1%.

Specific potential future research directions are: (1) Given two similar papers, investigating which paper is the original one and what are the differences between them, and (2) Given more than two similar papers, investigating what is the chronological order between the papers and what are the changes and developments along the time axis.

## References

1. Loui, M.C.: Seven Ways to Plagiarize. *Science and Engineering Ethics* 8(4), 529–539 (2002)
2. Martin, B.: Plagiarism: a Misplaced Emphasis. *Journal of Information Ethics* 3(2), 36–47 (1994)
3. Ceska, Z.: The Future of Copy Detection Techniques. In: Proceedings of the First Young Researchers Conference on Applied Sciences (YRCAS), pp. 5–10 (2007)
4. Schleimer, S., Wilkerson, D.S., Aiken, A.: Winnowing: Local Algorithms for Document Fingerprinting. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD 2003), pp. 76–85. ACM, New York (2003)
5. Collberg, C., Kobourov, S., Louie, J., Slattery, T.: Self-Plagiarism in Computer Science. *Commun. ACM* 48(4), 88–94 (2005)
6. Sorokina, D., Gehrke, J., Warner, S., Ginsparg, P.: Plagiarism Detection in arXiv. In: ICDM, pp. 1070–1075 (2006)
7. Keuskamp, D., Sliuzas, R.: Plagiarism Prevention or Detection? The Contribution of Text-Matching Software to Education about Academic Integrity. *Journal of Academic Language and Learning* 1(1), 91–99 (2007)
8. HaCohen-Kerner, Y., Tayeb, A., Ben-Dror, N.: Plagiarism Detection in Computer Science Papers. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), pp. 421–429 (2010)
9. Wise, M.J.: YAP3: Improved Detection of Similarities in Computer Program and Other Texts. *ACM SIGCSE* 28, 130–134 (1996)
10. Burrows, S., Tahaghoghi, S., Zobel, J.: Efficient and Effective Plagiarism Detection for Large Code Repositories. In: Proceedings of the Second Australian Undergraduate Students' Computing Conference, pp. 8–15 (2004)

11. Chen, X., Francia, B., Li, M., Mckinnon, B., Seker, A.: Shared Information and Program Plagiarism Detection. *IEEE Trans. Information Theory* 50(7), 1545–1550 (2004)
12. Jadalla, A., Elnagar, A.: PDE4Java: Plagiarism Detection Engine for Java Source Code: A Clustering Approach. *International Journal of Business Intelligence and Data Mining (IJBIDM)* 3(2), 121–135 (2008)
13. Manber, U.: Finding Similar Files in a Large File System. In: USENIX Technical Conference on USENIX Winter, San Francisco, CA, pp. 1–10 (1994)
14. Heintze, N.: Scalable Document Fingerprinting. In: USENIX Workshop on Electronic Commerce (1996)
15. Shivakumar, N., G-Molina, H.: SCAM: A Copy Detection Mechanism for Digital Documents. In: Proceedings of the Second Annual Conference on the Theory and Practice of Digital Libraries, vol. 24(2), pp. 398–409 (1995)
16. Broder, A.Z.: On the Resemblance and Containment of Document. In: Compression and Complexity of Sequences (SEQUENCES 1997), pp. 21–29 (1997)
17. Lyon, C., Malcolm, J., Dickerson, B.: Detecting Short Passages of Similar Text in Large Document Collections. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, pp. 118–125 (2001)
18. Hoad, T.C., Zobel, J.: Methods for Identifying Versioned and Plagiarised Documents. *Journal of the American Society for Information Science and Technology* 54(3), 203–215 (2003)
19. Monostori, K., Finkel, R., Zaslavsky, A., Hodász, G., Pataki, M.: Comparison of Overlap Detection Techniques. In: Sloot, P.M.A., Tan, C.J.K., Dongarra, J., Hoekstra, A.G. (eds.) *ICCS 2002, Part I. LNCS*, vol. 2329, pp. 51–60. Springer, Heidelberg (2002)
20. Bernstein, Y., Zobel, J.: A Scalable System for Identifying Co-derivative Documents. In: Apostolico, A., Melucci, M. (eds.) *SPIRE 2004. LNCS*, vol. 3246, pp. 55–67. Springer, Heidelberg (2004)
21. Forman, G., Eshghi, K., Chiocchetti, S.: Finding Similar Files in Large Document Repositories. In: *KDD 2005: Proceeding of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, New York, NY, USA, pp. 394–400 (2005)
22. Muthitacharoen, A., Chen, B., Mazieres, D.: A Low-Bandwidth Network File System. In: *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP 2001)*, Banff, Canada, pp. 174–187 (2001)
23. Ponzetto, S.P., Strube, M.: Semantic Role Labeling for Coreference Resolution. In: Companion Volume to the Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, April 3-7, pp. 143–146 (2006)
24. Ponzetto, S.P., Strube, M.: Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, New York, N.Y., June 4-9, pp. 192–199 (2006)

# Corpus-Based Semantic Filtering in Discovering Derivational Relations<sup>\*</sup>

Maciej Piasecki, Radosław Ramocki, and Paweł Minda

G4.19 Research Group,  
Institute of Informatics  
Wrocław University of Technology  
[{maciej.piasecki,radoslaw.ramocki}@pwr.wroc.pl](mailto:{maciej.piasecki,radoslaw.ramocki}@pwr.wroc.pl)

**Abstract.** Derivational relations are an important part of the lexical semantics system in many languages, especially those of rich inflection. They represent wide variety of semantic oppositions. Analysis of morphological word forms in terms of prefixes and suffixes provides limited information about their semantics. We propose a method of semantic classification of the potential derivational pairs. The method is based on supervised learning, but requires only a list of word pairs assigned to the derivational relations. The classification was based on a combination of features describing distribution of a derivative and derivational base in a large corpus together with their morphological and morpho-syntactic properties. The method does not use patterns based on close co-occurrence of a derivative and its base. Two classification schemes were evaluated: a multiclass and a cascade of binary classifiers, both expressed good performance in experiments on the selected nominal derivational relations.

**Keywords:** semantic filtering, derivational relations, corpus, semantic classification, wordnet.

## 1 Introduction

Derivational relations link two word forms where one – *a derivative* – was formed from the second – *a derivational base* by a kind of the word form transformation, mostly by adding a prefix or suffix. These relations express morpho-syntactic oppositions, e.g. gerunds–verbs, but they often encode primarily semantic oppositions, e.g. *actr-ess* – *actor*, *paint-ed* – *paint*. That is why, derivational relations has been included in many wordnets, starting with a few in Princeton WordNet [4], more in EuroWordNet [18], and in larger numbers in wordnets for Slavic languages, e.g. there are 50 relations defined in Polish plWordNet and many in CzechWordNet [7]. Many derivational relations are encoded by productive prefixes or suffixes, others can be described by limited sets of patterns. Tools based on rules that are manually written or automatically discovered in the training data were proposed for the generation and recognition of derivationally associated words. However, the patterns (built of suffixes and/or prefixes) mostly are not enough precise to generate only pairs that are truly associated by the given derivational

---

\* Partially financed by the Polish Ministry of Education and Science, Project N N516 068637.

relation, e.g. *pierwiastka* ‘a woman giving birth for the first time’ is not a female form of *pierwiastek* ‘root’, in spite of the suffix ‘*ka*’ which is typical for the femininity relation, e.g. *aktor-ka* ‘actress’ – *aktor* ‘actor’. Such pairs can be rejected only on the basis of semantic analysis. In order to avoid dependency on any previously existing resource, the analysis should be based on the knowledge extracted from a large text corpus. This is a kind of the word relation classification problem, but here classification is limited to the words that are formally associated, i.e. by specific suffixes or prefixes. The association of the word forms is not sufficient but necessary. The process consists of two phases: a pattern-based deterministic recognition of formal associations, and semantic relation classification.

Our goal is to develop a method for the classification of Polish word pairs that are derivationally associated into a set of semantic derivational relations. The method is aimed to be used in semi-automatic expansion of the Polish wordnet with new instances of derivational relations.

## 2 Background

There are several methods for learning derivational morphology patterns, cf [10]. However extracted lemma pairs are not semantically classified in applications to the automated wordnet expansion with derivational relations, e.g. [2]. Proper formal relations between word forms are not always correlated with a semantic relation, e.g. *bednarka* ‘cooperage’ is not related by femininity to *bednarz* ‘cooper’ (but -*ka* marks typically femininity) or *przepychaczka* ‘declogger’ is not a feminine from *przepychacz* ‘plunger’ – potential derivatives are neither +Human nor +Animal. It is very hard to find any works on semantic classification of derivational relations based on the data extracted from a corpus.

The problem seems to be very similar to the general problem of the classification of semantic relations between pairs of nominals, e.g. a definition of SemEval-2010 [6]. In both cases: the goal is to assign a lemma pair to one of the several semantic classes or none. However, we can notice significant differences in both tasks. Firstly, a technical difference, lemmas of different Parts of Speech participate in the derivational relations, while the known methods of semantic classification were mostly applied to noun-noun pairs. Secondly, a very significant discrepancy, a derivative and its base almost never occur together in the same sentence and or even in a larger text passage. Both lemmas must be described separately, on their separate occurrences, and there is no context of a derivational pair. Contrary to this, the vast majority of methods cited in [6] assumes that the descriptions of lemma pairs are extracted from sentences in which both pair elements occur. For instance, [16] presented an algorithm based on Vector Space Model for solving verb analogy questions. [9] in their seminal paper presented the *Espresso* algorithm for iterative discovering patterns and relation instances. *Espresso* is relation independent, but requires both instance elements to co-occur within limited text context. [15] presented a method for classifying hypernymic lemma pairs based on patterns extracted from sentences including such pairs. In [11] a classifier is trained on data collected from both individual occurrences of lemmas and co-occurrences matching a limited set of generic patterns, As a result, a set of features was defined and used in

classification of lemma pairs into a few basic wordnet relations. The approach was applied only to frequent lemmas. Thirdly, an important source of information is available in the case of the classification of derivational relations: both lemmas must be formally associated by a derivation patterns, the domain of classification is clearly delimited. Finally, derivatives, even created from frequent bases, are often infrequent in the corpus. This can be a serious challenge, especially as differences between different subtypes of derivational relations, briefly presented below, can be very subtle.

As a source of training-testing examples we used *plWordNet*<sup>1</sup> – a very large Polish wordnet [10] including more than 142 000 lexical units. Polish has rich system of derivational relations and *plWordNet* includes 50 derivational and derivationally motivated relations (together with subtypes), e.g.:

- **cross-categorial synonymy**: (N-V) deverbal nouns (gerunds) – verb base, e.g. *pływanie* ‘swimming’ < *pływać* ‘swim<sub>impf</sub>’, (A-V) adjectival participle – verb, (N-A) deadjectival N. – Adj., e.g. *bladość* ‘paleness’ < *blady* ‘pale’;
- **femininity** (N-N), e.g. *psycholożka* ‘female psychologist’ < *psycholog* ‘psychologist’;
- **markedness**, three subtypes, all of the scheme (N-N): *diminutives*, e.g. *piesek* ‘little or pleasant dog’ < *pies* ‘dog’, *augmentatives*, e.g. *komarzysko* ‘huge or terrible mosquito’ < *komar* ‘mosquito’, *young being*;
- **semantic role**, 7 subtypes following EuroWordNet, i.e. agent, instrument, location, object, patient, product, time and other, all (N-V), e.g. for agent *spawacz* ‘welder’ < *spawać* ‘weld’ or for *szpieg* ‘spy’ < *szpiegować* ‘spy’;
- **role of hidden predicate** a version of *role* linking two nouns (N-N), it has three subtypes, namely: *agent*, *location* and *product*, e.g. agent *gołębiarz* ‘pigeon keeper’ < *gołąb* ‘pigeon’, location *kwiaciarnia* ‘flower shop’ < *kwiat* ‘flower’, product *kapuśniak* ‘sauerkraut soup’ < *kapusta* ‘cabbage’;
- **role inclusion** (V-N) with the same 7 subtypes as for the semantic role, e.g. for instrument *pieprzyć* ‘to pepper’ < *pieprz* ‘pepper’;
- **state/feature bearer** (N-Adj), e.g. *głupiec* ‘a fool, idiot’ < *głupi*, *mędrzec* ‘sage’ < *mqdry* ‘wise’;
- **inhabitant** (N-N), e.g. *Bułgar* ‘Bulgarian’ < *Bułgaria* ‘Bulgaria’;
- **aspectuality** (V-V), e.g. expresses aspectual and *Aktionsart* differences.

In this paper we focused on selected most difficult relations, less regular, between nouns and linking nouns to verbs. We assumed a two stage process. First a kind morphological analyser is applied to Polish word forms as potential derivatives to identify their derivational bases and name the relations, if there are any. Next, the triples returned by the morphological tool are semantically classified on the basis of knowledge extracted from a corpus. For the first stage, we used *Derywator* [10] a morphological recogniser of Polish derivatives working according to the specification given above. *Derywator* is trained on instance examples collected from *plWordNet*, but it does have any access to the corpus.

---

<sup>1</sup> [www.plwordnet.pwr.wroc.pl](http://www.plwordnet.pwr.wroc.pl)

### 3 Semantic Classification of Derivational Relations

Derivatives and derivational bases almost never co-occur, so we could not use lexico-syntactic patterns to collect data. Instead, we used an indirect approach based on building the description of derivatives and derivational bases using methods of Distributional Semantics, and next comparing those descriptions to recognise potential derivational relation instances. On the basis of huge corpus we constructed a large co-occurrence matrix in which both derivatives and derivational bases are assigned to the rows while columns correspond to features describing contexts. As context features, we applied three lexico-morpho-syntactic constraints proposed in [12]. The constraints were written in a formal constraint language called *WCCL* developed from a language for morpho-syntactic tagging rules [14]. The constraint finds word-to-word syntactic dependency of the given type: modification of a noun, coordination with a noun and subject relation to a verb, cf [12]. Due to rich inflection of Polish, the constraints achieved high precision above 90% [12], however, the recall is probably lower.

As a result, we obtained a matrix in which cell values are frequencies of co-occurrence of a word (a derivative or a base) and a lexical feature (a specific word in a specific syntactic relation) across all sentences in the corpus.

As many words described are infrequent the initial matrix contains a lot of statistical noise: non-informative and too noisy columns were next removed. Three thresholds were applied: on the *minimal values* of the sum over a column (feature frequency), on *minimal number* of non zero cells in a column (number of lemmas described), and on the *column entropy* value (feature ability to differentiate). We also filtered out words that are not enough described by columns (i.e. lack of evidence to infer about them): the sum over a row – the frequency of co-occurrences with any feature – had to be above some threshold. Detailed values for the filtering thresholds were described in the evaluation Sec. 4.1.

After filtering the matrix was weighted by applying *Pointwise Mutual Information* with discount factor, e.g. [8], in order to give more weight to surprising events and less to expected ones. Generally weighted matrices of event frequencies perform better than not weighted [17]. Finally, *Latent Dirichlet Allocation*, (LDA) [1] was applied to reduce the size and smooth the matrix. In lower dimensional space more abstract meaning of a word can be captured, and the reduced number of features improves training of classifiers. The transformed matrix was next used to generate feature vectors describing derivatives and derivational bases for training and testing classifiers. Because we have to classify lemma pairs as derivational relation instances we used two matrix rows to form a training/test example. The vector was next extended with other features described below.

Corpus based features do not provide information concerning morpho-syntactic properties of lemmas including potential suffixes and prefixes. Such information can be important for derivational relations that was confirmed during experiments. In [10], morpho-syntactic constraints on the properties of a derivative and its base were defined for different subtypes of derivational relations. The constraints refer to grammatical class, gender and case. In order to allow a classifier to learn those constraints we added

morphological analysis for a derivative and it's base to the feature set. Because one lemma can be ambiguous every lemma morphological analysis obtained from morphological analyser *Morfeusz SGJP*<sup>2</sup> was added to the feature set.

*Derywator* achieves relatively good precision using only morphological features, cf [IO], i.e. prefixes, suffixes of a derivative. Those prefixes and suffixes provide fine grained information concerning derivational relation type. However, there are too many different prefixes and suffixes to encode them in training/testing vectors. A classifier would be too sensitive to any change in them. In order to solve this problem we selected a set of the most frequent suffixes in the training set (only suffixes occur in noun relations). For each word we calculated Levenshtein edit distance to every suffix from the selected set of the 200 most frequent suffixes. Vector of edit distances was added to the set of features. In addition the suffix length of the given word was also added to the feature set.

## 4 Evaluation

We evaluated our method using two models of classification: multi-class and binary. In multi-class approach, the training set was divided into three categories: coarse-grained relations without subtypes (i.e. femininity, inhabitant), fine-grained for *markedness*, and fine-grained for semantic *role of hidden predicate*. Next we trained multiclass classifier for each category. Following this approach we obtained better results than by using a single classifier for all subtypes.

In second method we trained a single classifier for each relation subtype. This approach allows us to classify also multiple relations between one word pair. Two strategies for the application of a set of binary classifiers were tested: *parallel application* of all classifiers to a test sample, and a *cascade model* in which classifiers are organised into a sequence: a test example is delivered to the first one, and goes along the sequence until a positive decision has been made. Finally, the cascade model was chosen, because individual classifiers express some error that led to the over-generation of the parallel model. The classifiers were organised according to their accuracies measures during cross-validation.

### 4.1 Experiment Setup

Instances of 7 derivational relation subtypes were selected from *plWordNet 1.6*, see Tab. II. For each lemma in *plWordnet* we used trained *Derywator* to generate its potential derivative base and derivational relations. If pair  $\langle \text{base}, \text{derivative} \rangle$  was included in the full training set and the proposed relation is different from those in the training set, we added this pair to the *negative class*. The semantic classifier is intended to be a post-filter for results produced by *Derywator*. This strategy is justified by the extensive coverage of *Derywator* and the assumption that the word form association (by means of prefixes and suffixes) is primary in the case of derivational relations.

---

<sup>2</sup> <http://sgjp.pl/morfeusz/>

**Table 1.** Distribution of training set

Relation	Subtype	Instances
negative	-	1286
femininity	-	1837
inhabitant	-	188
markedness	deminutivity	1799
markedness	augmentativity	275
markedness	young being	53
semantic role	agent of hidden predicate	1710
semantic role	location of hidden predicate	170

**Table 2.** Cross-validation results for Derywator

Relation	Precision	Recall	F-score
femininity	97.00	63.60	76.83
inhabitant	71.70	12.40	21.14
deminutivity	98.30	67.30	79.90
augmentativity	73.90	22.00	33.91
young being	40.00	20.50	27.11
agent of hidden predicate	89.40	34.90	50.20
location of hidden predicate	65.90	18.30	28.65

We used a Polish corpus of about  $1.6 \times 10^9$  tokens combining: *IPI PAN Corpus* [13], electronic edition of Polish daily newspaper *Rzeczpospolita* [19], Polish Wikipedia<sup>3</sup> and texts collected from Internet, cf [12]. Co-occurrence matrix was built, filtered and transformed with the help of *SuperMatrix* system [2].

We used SVM classifiers with linear kernels from *LibLINEAR* package [3] that can be used effectively on huge data. For binary classification L2-regularized L2-loss support vector classification (dual) was used, and multi-class support vector classification by Crammer and Singer for multiple classes. In addition *Weka* [5] was utilised as a wrapper for cross-validation.

Co-occurrence matrix was filtered using the following parameters: minimal sum over a column: 100, minimal number of non zero values in a column: 5, we removed 1% of columns with maximal entropy, minimal sum over a row: 50. Before filtering the co-occurrence matrix contained: 659176 columns, and after 45548 (filtering reduced this value to: 45548). Next we applied LDA to reduce the matrix dimensionality to 500. Full training set contained 7318 examples, filtered matrix contained evidence for 3118 of them. Other examples did not pass filtering (a derivative and/or a derivational base).

For all experiments a modified 10-fold cross validation scheme was applied. Pairs for each relation subtype were randomly divided into 10 subsets. One subset per relation subtype was used for testing in each iteration.

## 4.2 Cross-Validation Results

First we tested *Derywator* as the only source of classification. Average results across folds of the fine grained classification are presented in Tab. 2. A comparison with the coarse-grained results presented in [10] reveals that most errors occurs among subtypes of the main relations. Lower precision is clearly correlated with the low number of training examples presented in Tab. 1. For *femininity* *Derywator* presents good precision, but *femininity* is encoded by characteristic suffixes. *Agent of hidden predicate* has similar number of training examples like *femininity*, but much more varied suffixes and that is visible in lower precision and much lower recall. As the number of relation

<sup>3</sup> [pl.wikipedia.org](http://pl.wikipedia.org)

**Table 3.** Cross-validation results of multiclass semantic classification

<b>Relation</b>	Distributional features			Distributional, text and morphological features		
	<b>Precision</b>	<b>Recall</b>	<b>F-score</b>	<b>Precision</b>	<b>Recall</b>	<b>F-score</b>
<i>femininity</i>	86.13	93.58	89.70	95.09	98.36	96.70
<i>inhabitant</i>	91.92	85.05	88.35	91.26	87.85	89.52
<i>markedness</i>	83.95	93.71	88.56	89.70	98.16	93.74
deminutivity	81.04	97.27	88.42	89.31	97.17	93.08
augmentativity	57.14	15.50	24.39	70.80	62.02	66.12
young being	0.0	0.0	-	53.85	26.92	35.90
<i>semantic role</i>	85.03	90.79	87.82	84.79	93.23	88.81
agent of hidden predicate	85.50	96.45	90.65	86.74	96.09	91.17
location of hidden predicate	89.06	79.17	83.82	83.33	83.33	83.33

**Table 4.** Cross-validation results of binary semantic classification

<b>Relation</b>	Distributional features			Distributional, text and morphological features		
	<b>Precision</b>	<b>Recall</b>	<b>F-score</b>	<b>Precision</b>	<b>Recall</b>	<b>F-score</b>
<i>femininity</i>	89.94	88.12	89.02	96.20	98.21	97.19
<i>inhabitant</i>	93.62	82.24	87.56	91.84	84.11	87.80
<i>markedness</i>	87.26	91.00	89.09	91.78	93.71	92.73
deminutivity	82.09	94.54	87.88	90.03	95.85	92.85
augmentativity	61.90	10.08	17.33	66.96	58.14	62.24
young being	-	0.0	-	58.33	26.92	36.84
<i>semantic role</i>	86.68	86.77	86.73	85.86	89.95	87.86
agent of hidden predicate	87.35	95.73	91.35	87.68	95.26	91.31
location of hidden predicate	92.98	73.61	82.17	83.08	75.00	78.83

instances per relation subtype is not still representative for Polish in *plWordNet* we did not expect high recall in the results of *Derywator*. It should be emphasised that *Derywator* is not a baseline classifier for the semantic classifiers tested in the experiments presented below. Results achieved with *Derywator* show how far we can go with the analysis of the morphological forms. *Derywator* cannot come closer to the lexical units (words senses) and presents lower results for words from the outside of *plWordNet*. It is based on transducers and have limited generalisation capacity, as well as prefixes and suffixes express significant ambiguity.

Next two selected approaches to classification were tested: a multiclass classifier and a cascade of binary classifiers. Both were performed on the same data sets as the first experiment and the results are presented in Tab. 3 and 4. Both classification schemes express good results for (main) relations and almost all subtypes. *Augmentativity* and *young being* are notable exceptions, but the number of training examples for those classes was much lower. However, a similarly low number of training examples in the case of *inhabitant* did not reduce performance. A possible explanation is that

*inhabitant* relation has much narrower and semantically better delimited domain than *augmentativity* and *young being*. The cascade of binary classifiers is characterised by slightly higher precision than multi-class approach, but lower recall. It means that borders among relations are more sharply identified by the binary classifiers and the idea of the cascade ordered according to the precision works well for the whole solution.

Semantic classifiers present better results than *Derywator*. They use also information about morpho-syntactic tags and suffixes but in a reduced form. Initial experiments on the application of the semantic classifiers to filtering the output of *Derywator* showed very good precision in the case of identical subtypes assigned by both, but for the price of low recall. Unfortunately, both semantic classifiers are correlated and *Derywator* does not provide scores, so there is no possibility to combine them in voting scheme. Larger number of examples could improve recall of *Derywator*. Classifiers are also limited to words occurring frequently enough in a large corpus. *Derywator* can produce good decisions for very infrequent words.

## 5 Conclusions

Derivational relations are an important part of lexical semantics system in many languages, especially those of rich inflection, e.g. all Slavic languages. They represent wide variety of semantic oppositions. Many derivational forms are too infrequent to be described in semantic lexicons and also are created *ad hoc* in everyday language use. So their effective recognition can increase performance of tools applied to semantic processing of texts. Analysis of morphological word forms provides a limited information about their semantics. As we argued in Sec. 2 there is a substantial level of ambiguity in prefixes and suffixes. We need to apply a kind of semantic processing.

We have proposed a method of semantic classification of the potential derivational pairs. The method is based on the supervised learning, but it only needs a list of pairs assigned to the derivational relations (e.g. from a wordnet). The method is language independent to a very large extent (only a small table of internal stem alternations must be defined, cf [10]) as the assumed level of syntactic processing of the corpus is very limited to simple morpho-syntactic constraints. *SuperMatrix* system for co-occurrence matrix construction is language independent and is available on GPL licence. The classification was based on a combination of features describing distribution of a derivative and derivational base in a large corpus together with their morphological and morpho-syntactic properties. The method does not use patterns based on close co-occurrence of a derivative and its base that happens very rarely.

The achieved result of the semantic classification outperformed significantly classification based only on the word form analysis. However, both methods are complementary to some extent as the proposed semantic classification requires larger number of word occurrences in a corpus. We combined both approaches in a tool called WordnetWeaver [12] supporting linguists in wordnet expansion with new instances of derivational relations. Semantic classifiers combined with automated wordnet expansion method create a possibility to go further and analyse derivational pairs as consisting of lexical units, not lemmas.

## References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* 3, 993–1022 (2003)
2. Broda, B., Piasecki, M.: SuperMatrix: a General Tool for Lexical Semantic Knowledge Acquisition. In: Proc. of IMCSIT — 3rd International Symposium Advances in Artificial Intelligence and Applications (AAIA 2008), pp. 345–352 (2008)
3. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. *J. Mach. Learn. Res.* 9, 1871–1874 (2008)
4. Fellbaum, C. (ed.): WordNet — An Electronic Lexical Database. The MIT Press (1998)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. *SIGKDD Explor. Newsl.* 11(1), 10–18 (2009)
6. Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S.: SemEval-2010 Task 8: Multi-way classification of semantic relations between pairs of nominals. In: Proc. of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions, pp. 94–99. ACL (2009)
7. Pala, K., Hlaváčková, D.: Derivational Relations in Czech WordNet. In: Proc. Workshop on Balto-Slavonic NLP, Prague, pp. 75–81 (2007)
8. Pantel, P.: Clustering by committee. Ph.D. thesis, Edmonton, Alta., Canada, (2003); adviser- Dekang Lin
9. Pantel, P., Pennacchiotti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: ACL 2006 (ed.) Proc. COLING-ACL 2006, Sydney, pp. 113–120. ACL (2006)
10. Piasecki, M., Ramocki, R., Maziarz, M.: Automated generation of derivative relations in the wordnet expansion perspective. In: Proceedings of 6th International Global Wordnet Conference, pp. 273–280. The Global WordNet Association, Matsue (2012)
11. Piasecki, M., Szpakowicz, S., Marcińczuk, M., Broda, B.: Classification-Based Filtering of Semantic Relatedness in Hypernymy Extraction. In: Nordström, B., Ranta, A. (eds.) GoTAL 2008. LNCS (LNAI), vol. 5221, pp. 393–404. Springer, Heidelberg (2008)
12. Piasecki, M., Szpakowicz, S., Broda, B.: A Wordnet from the Ground Up. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2009)
13. Przeźiórkowski, A.: The IPI PAN Corpus: Preliminary version. Institute of Computer Science PAS (2004)
14. Radziszewski, A., Wardyński, A., Śniatowski, T.: WCCL: A Morpho-syntactic Feature Toolkit. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNCS, vol. 6836, pp. 434–441. Springer, Heidelberg (2011)
15. Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems, vol. 17, pp. 1297–1304. MIT Press, Cambridge (2005)
16. Turney, P.D., Littman, M.L.: Corpus-based learning of analogies and semantic relations. *Machine Learning* 60(1–3), 251–278 (2005)
17. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37, 141 (2010)
18. Vossen, P.: EuroWordNet General Document Version 3. Tech. rep., Univ. of Amsterdam (2002)
19. Weiss, D.: Korpus Rzeczypospolitej, corpus from the online edition of *Rzeczypospolita* (2008), <http://www.cs.put.poznan.pl/dweiss/rzeczypospolita>

# Lexical Activation Area Attachment Algorithm for Wordnet Expansion

Maciej Piasecki, Roman Kurc, Radosław Ramocki, and Bartosz Broda

G4.19 Research Group, Institute of Informatics

Wrocław University of Technology

{maciej.piasecki,radoslaw.ramocki}@pwr.wroc.pl

**Abstract.** The paper presents an algorithm of the automated wordnet expansion which can utilise results produced by both pattern-based and Distributional Similarity methods. It is based on the assumption that all relation extraction methods express some error, so we cannot identify the exact place (synset) for a new lemma on their bases, but an area (a wordnet subgraph). Support for a particular attachment point generated by knowledge should be expanded on the surrounding synsets. Moreover, the wordnet structure is modelled on the level of links between lexical units. Evaluation of the algorithm and comparison with top algorithm from literature in large scale experiments on Princeton WordNet is presented.

**Keywords:** automated wordnet expansion, semantic relation extraction, wordnet.

## 1 Introduction

Lexical semantic networks emerged as important language resources providing relational description of lexical meanings. Among them, wordnets achieved size that is sufficient for practical applications, e.g. WordNet [2], or plWordNet [6]. However, even the largest networks created manually include gaps in coverage. Several methods for the automated extraction of lexico-semantic relations from text corpora have been developed. In our work, we focus on the hypernymy structure as the skeleton, however, we go beyond the taxonomy towards a more complete description of the hypernymy, as it is provided in wordnets, especially those based on linguistic intuitions, e.g. plWordNet.

Upper levels of a wordnet hypernymy relation describe more general, often highly abstract *lexical units* (LUs) (LU is a pair: lemma and its sense), but those levels are mostly built manually in most wordnets. Thus, our goal is to develop a method of automated wordnet expansion based on lexico-semantic relations extracted from a large corpus and the prior partial wordnet structure. In order to make the method enough general, we want to use only text corpora and avoid using any kind of semantic meta-information, e.g. from encyclopaedia. Most known taxonomy induction methods utilise only the existing hypernymy structure in incremental wordnet expansion, e.g. [7,6]. In our approach, the wordnet hypernymy structure is perceived as intrinsically interlinked to other wordnet relations (conceptual and lexical [2]). Thus, we aim at utilising all different types of links in our wordnet expansion method.

## 2 Related Works

In [10] wordnet hypernymy is treated as a kind of decision tree applied to word meanings described by Distributional Semantics methods, but both approaches were tested on a limited domains of concrete and frequent nouns. [9] represented word meaning by *semantic neighbours* –  $k$  most similar words according to Measure of Semantic Relatedness (MSR).

[7] proposed Probabilistic Wordnet Expansion (PWE) method based on a probabilistic model of the taxonomy. It was described by two relations: (transitive) *hypernymy* and  $(m,n)$ -*cousin*. Probabilities of relation links were estimated on the basis of a corpus. Probabilities of the relation links were estimated on the basis logistic regression applied to: lexico-syntactic patterns in the case of hypernymy and MSR-based similarity to word clusters in the case of cousin relation. To prevent adding a new word to overly-specific hypernym  $\lambda$  coefficient was introduced penalized by:  $\lambda^{k-1}$  factor, where  $k$  is number of links between attachment synset and it's hypernym. The cousin relation is a generalization of the co-hyponymy relation. Hypernymy and  $(m,n)$ -cousinhood instances imply sets of other instances, e.g. a direct hypernym of one word sense implies all other indirect hypernyms: to add a new word to the taxonomy, the whole taxonomy must be (locally) searched for an attachment place that maximises probabilities of all the implied relations. Attachment of new elements transforms the structure  $T$  into a new  $T'$ . The appropriate  $T'$  maximises the probability of the change in relation to the evidence at hand. *Multiplicative change* computation is based on all added relation instances (i.e. links), including the instances *implied* by hyponymy. According to the probabilistic model of PWE all relation links: direct and implied should be taken into account in calculating multiplicative change. However, many implied links have zero probability and they are simply excluded from the calculation in the implementation (contrary to the model!).

Two step taxonomy induction was presented in [4]. First, hyponym-hypernym pairs are extracted from Internet by “doubly-anchored lexico-syntactic patterns” and ranked by lemma-pattern co-occurrence frequencies. In the second step, additional patterns are used to find support for different links, but also some constraints are imposed on the hypernym graph structure.

Taxonomy induction in [5] is focused on ontology learning, identification of overt definitions in text and the extraction of hypernymy instances from them. However definitions are infrequent and occur only in specific text genres. The initial graph emerging from the extracted pairs is next weighted and prunned.

A *metric-based taxonomy induction framework* [11] utilises 15 different extraction methods producing *feature functions*: a term pair → a real value or  $\{0, 1\}$  for pattern results. The process starts with an initial partial taxonomy  $T^0$ , used also to estimate values of parameters. Taxonomy expansion is controlled by two principles: *Minimum Evolution Assumption* and *Abstractness Assumption*. The former results in minimising “the overall semantic distance among the terms”. The total distance and change are characterised by the Information Function of a taxonomy  $T$ :  $Info(T) = \sum_{x < y, c_x, c_y \in C} d(c_x, c_y)$ , where  $C$  is a set of all terms and  $d(\cdot)$  is a *ontology metric*. According to the second principle, weights for different *feature functions* can be estimated in supervised training for each taxonomy level separately by approximating

ontology metrics for term pairs. The approximation was performed by ridge regression, and Multi-Criterion Optimization Algorithm (MCOA) finds a place for each new term by joint application of both conditions, i.e. minimising both: the change in the taxonomy Information Function and the sum over the square error of the difference between new ontology metrics and their estimation based on the weighted feature functions. [11] performed evaluation on WordNet and an ontology. Concerning the first, 50 “hypernym taxonomies” were extracted from 12 topics (mostly concrete nouns) and 50 “meronymic taxonomies” from 15 topics (mostly concrete). The size of the test taxonomies and the way of their delimitation was not defined. Feature functions were built on the basis of a corpus including English Wikipedia and 1000 top documents per each term from Google. Leave-one-out cross validation was applied on the level of taxonomies (1 test taxonomy, 49 for the weight estimation). Precision and recall were calculated on the level of relation links. So, the number of the correctly attached terms is not known. Comparison with several algorithms was presented, including PWE, however, the reimplementation of the latter was not described (PWE source code is not available) and MCOA achieved slightly better precision and f-measure for the reconstruction of the hypernymic taxonomies than PWE.

### 3 Lexical Area Attachment Algorithm

There is no one best methods for extracting lexico-semantic relations from text, e.g. methods based on Distributional Semantics requires large numbers of occurrences and a varied corpus, while pattern-based methods are prone to errors caused by processing tools. We need to use all several different methods and use every bit of information extracted but trust almost to nothing. This is the basic idea behind our wordnet expansion method called *Lexical Area Attachment Algorithm* (LAAA). LAAA is a modification of AAA [6], but it is less heuristic than AAA, based directly on the LU relations and better utilising context.

A *knowledge source* (KS) is a result produced by a relation extraction method and is represented as a set of triples:  $\langle x, y, w \rangle$ , where  $x$  and  $y$  are lemmas and  $w \in R$  is a weight. Some KS can be interpreted in terms of probabilities but not all, e.g. those produced by patterns. So, firstly, we assumed that weights provided by KSs cannot be universally interpreted against a common probabilistic model. Secondly, all KSs express intrinsically some error. Thirdly, new LUs can be added by LAAA to an already existing synset (if it exist) as synonyms, otherwise, a new synset linked by a hyper/hyponymy or holo/meronymy should be created. For a new lemma  $x$ , not yet described in a wordnet, a KS  $K$  can include several triples  $\langle x, y_i, w \rangle$  such that  $y_i$  corresponds possibly to several LUs  $l_m^{y_i}$  (synset members) in the wordnet. The support for  $x$  provided the  $K$  triple can pertain to any subset of  $l_m^{y_i}$  – the meaning of  $y_i$  is not disambiguated in  $K$ . Moreover, due to the potential errors in  $K$  we cannot be sure whether semantic associations:  $x - l_m^{y_i}$  suggested by  $K$  should not be in fact switched to one of the LUs that are semantically close to  $l_m^{y_i}$ . Let  $S_j$  be synset and  $l_m^{y_i} \in S_j$ . LUs included in synsets linked to  $S_j$  by short paths in the wordnet graph are semantically related to  $l_m^{y_i}$ . The support of  $K$  for a pair  $\langle x, l_m^{y_i} \rangle$  may in fact rather pertain to one of those LUs, and, *vice versa*, the support for lemmas of those LUs maybe should be assigned to the pair  $\langle x, l_m^{y_i} \rangle$  instead.

The main idea of LAAA is to treat each association suggested by a triple  $\langle x, y_i, w \rangle$  from a KS as pertaining to *an area* in the wordnet graph around the point identified by  $y_i$  (or points if  $y_i$  is polysemous). The most likely place is the synset  $S_j \ni y_i$ , but we cannot exclude the possibility that the support should be pertain to synsets linked to  $S_j$ , e.g. MSR can return for *lorry*: *truck*, but also *vehicle* and *semitrailer*, while synset of *truck* is the most appropriate point. Thus, support should be *replicated* to all synsets that are likely to be associated with  $x$ . However, the amount of replication should depend on how far is the given synset from  $S_j$  and what type of links are included in the path linking to  $S_j$ , e.g. many relation extraction methods barely distinguish among close hyper/hyponyms but are better in differentiating synonyms and antonyms. In LAAA, path properties are modelled by: its length, *transmittance* and *impedance*, defined below. Thus, analysing an association between  $x$  and  $S_j$  we need to consider both direct support from KSs, but also indirect replicated from the local context.

In order to find a place in the wordnet for a new LUs of  $x$  all synsets must be considered. Calculation of the support for  $x$  and the synset  $S$  is based on *combining* support from different KSs and *collecting* it from the whole wordnet subgraphs around  $S$ . In combining we must take into account types of the KSs and weights assigned. In collecting properties of the paths must be taken into account. Link semantics that should influence the way in which indirect support ‘is replicated through’ it, e.g. it is more likely that an association attributed by a KS to the hypernym can be also attributed to its hyponyms than in the case of two antonyms. Thus, we assume that the level up which indirect support influences the surrounding depends on the path length and types of the links comprising it., e.g. it less slowly decreases is along hyponymic paths than meronimic ones.

Assuming that the support strength can be expressed as a real value, for each lexico-semantic relation a *transmittance* function:  $R \rightarrow R$  is defined, which describes the link ability to transmit support. Link-to-link connection is characterised by the *impedance* function:  $R \rightarrow R$ , defined for each relation pair. The impedance describes how much indirect support can be replicated through the given connection, e.g. transmission of support through holonymy–meronymy would mean that the direct support assigned to the whole (a holonym) via a part (a meronym) could be attributed as the indirect support to the another whole (its second holonym), e.g. *car*–holonym–*windscreen*–meronym:substance–*glass*: indirect support could go from *car* to *glass* that is clearly too far.

### 3.1 Formal Model

Let  $WN = \langle J, L, S, \mathbf{A}, \mathbf{B}, f_{Lem}, R_{Syn}, f_{Snst} \rangle$ , where  $J$  is a set of LUs,  $L$  – set of lemmas,  $S \subseteq 2^J$  – a set of synsets,  $\mathbf{A} \subseteq 2^{J^2}$  – a set of lexico-semantic relations defined on  $J$  (or *lexical relations*),  $\mathbf{B} \subseteq 2^{S^2}$  – a set of lexico-semantic relations defined on  $S$  (or *conceptual relations*), i.e. relation instances of both types are ordered pairs;  $f_{Lem} : J \rightarrow L$ .  $R_{Syn} \in \mathbf{B}$  is the synonymy relation,  $f_{Snst} : J \rightarrow 2^J$ , is such that :  $f_{Snst}(j) = \{j' : \langle j, j' \rangle \in R_{Syn}\}$

$\mathbf{B}' \subseteq 2^{J^2}$  is a set of lexico-semantic relations defined on  $J$  that mirrors the synset relations, i.e. for each synset pair  $r$  linked by a relation  $R$  in  $\mathbf{B}$  all LU pairs from the

synsets of  $r$  belong to the relation  $R'$  that corresponds to  $R$ , e.g. for each hypernymic pair of synsets all LU pairs made from it are linked by hypernymy in  $\mathbf{B}'$ . Relations from  $\mathbf{B}'$  do not include any LU pairs that are not supported by the synset relations from  $\mathbf{B}$ .

A knowledge source  $K$  is a set of triples of the type:  $L \times L \times R$  where  $R$  is a set of real numbers. Let  $\mathbf{K}$  is a matrix of the KSs and  $\mathbf{W}$  is a matrix of the *global weights* assigned to the KSs (*local weights* are stored in the triples).

The *transmittion* is represented by:  $f_T : (\mathbf{A} \cup \mathbf{B}') \times R \rightarrow R$

The *impedance* is represented by:  $f_I : (\mathbf{A} \cup \mathbf{B}') \times (\mathbf{A} \cup \mathbf{B}') \times R \rightarrow R$

### 3.2 Algorithm

Algorithm consists of two phases. During the first lemma-to-synset *semantic fit* for a new lemma  $x$  is calculated by combing direct support from the synset lemmas and indirect support from the synset context. In the second phase we look for connected wordnet subgraphs such that each synset in a subgraph expresses semantic fit to  $x$  above a pre-defined threshold. The identified subgraphs are called *activation areas*. The output of the algorithm: attachment areas describing suggested locations for  $x$  are selected from activation areas.

**Input:**  $x$  – a new lemma.

**Parameters:**  $rad \in N$  – a size of the local context,  $min\_fit$  is a threshold for the minimal value of the lemma-to-synset fit that is worth to be considered (established experimentally),  $strong\_fit$  – a threshold that can be exceeded by the fit only if it is collected from several KSs or rich indirect support,  $max_{att}$  – the maximal number of suggested senses that are to presented to the linguist.

*Phase I.* Lemma-to-synset fit calculation: for each synset  $S$  repeat:

1.  $sum \leftarrow \sum_{u \in S} \forall_{k=1 \dots size(\mathbf{K})} \mathbf{W}(k) \mathbf{K}[k](x, f_{Lem}(u))$
2.  $P \leftarrow paths\_in(S, rad)$
3. for each  $p \in P$ 
  - (a)  $sum \leftarrow sum + f_{cnt}(fit\_trans(p, x), length(p))$
4.  $fit[S, x] \leftarrow sum$

where  $paths\_in(S, rad)$ :

- returns a set of paths built over  $\mathbf{A} \cup \mathbf{B}'$  such that they end in a  $u \in S$  and are no longer than  $rad$
- if two paths starts in the same LU, then only the shorter one or randomly selected one is kept;
- a path is of type  $(J^2)^+$ , i.e. links are represented as LU pairs.

$fit\_trans(p, x)$ :

1. if  $p = []$  then return 0
2.  $h \leftarrow head(p), t \leftarrow tail(p)$
3. if  $t = []$  then return  $f_T(h, \sum_{k=1 \dots size(\mathbf{K})} \mathbf{W}[k] \mathbf{K}[k](x, f_{Lem}(h_2|_1)))$
4. else return  $f_T(h, f_I(h, head(t), fit\_trans(tail(t), x)))$

### Phase II. Finding activation and attachment areas

1. Let  $\mathbf{G} \subseteq \mathbf{B}$  is a selected subset of the synset relations (at least hypernymy and hyponymy).
2.  $Act(x) = \{\mathbf{S} : \mathbf{S} = \{S : S \text{ is a synset} \& fit[S, x] \geq min\_fit\} \& \mathbf{S} \text{ is connected subgraph with respect to } \mathbf{G}\}$ .
3.  $Att(x) \leftarrow \{ \text{the top } max_{att} \text{ subgraphs } \mathbf{A} \in Act(x) \text{ according to } max_{S \in \mathbf{A}}(fit[S, x]) \} \cup \{ \text{all } \mathbf{A} \in Att(x) \text{ such that } max_{S \in \mathbf{A}}(fit[S, x]) \geq strong\_fit \}$

**Phase I**, for each synset  $S$  we calculate its lemma-to-synset semantic fit for  $x$ . In Step (1) direct support is computed as the weighted sum of values from KSs. Global weights can be interpreted as the measure of the quality of the given KS and estimated on the basis of the KS accuracy, cf Sec. 4. In Step (2) a set of paths leading to all synsets from the local context of  $S$  is build in the breadth-first scheme up to  $rad$  links from  $S$ . For each path the indirect support originating from the final node (synset) is replicated and functions of transmittance and impedance are applied iteratively to the initial value.

**Phase II**, in Step (1), attachment areas express locations found by LAAA for  $x$ , however the final decision of the linguist must be done for a particular synset. Thus, we look for local maxima of the fit value and use these values as representative for attachment areas. LAAA is focused on supporting linguists, recall is important, so up to  $max_{att}$  activation areas are finally returned as suggested *attachment areas*. However, some lemmas can be highly polysemous and we can have good evidence for their senses. That is why we introduced a threshold for *strong fit* i.e. based on reliable evidence, and several KS.

## 4 Experiment Setup

We follow evaluation methodology proposed in [1]: first we remove a word sample from a wordnet, then we apply expansion algorithm to reattach the words. Only words above a corpus frequency threshold are included in a sample (MSR extraction methods work well for more frequent words) – 200 was used. Moreover, only words that have at least 3 hypernymy links to the top synset were included in a sample, as we assumed that the top part is constructed manually, cf [1]. Sample of 1064 test words from WordNet 3.0 was randomly selected (the size is enough large for the margin of error 3% and 95% confidence level [3]).

LAAA may give several attachment areas for a word, all are sorted by the semantic fit. For evaluation we build histogram of path lengths between suggested synsets and the original positions in a wordnet. Paths of up to 5 links, including hyper/hyponymy links with at most one final meronymic were considered. Longer suggestion distances were discarded as not useful for linguists, i.e. in WordnetWeaver system [6] the attachment areas of AAA are visually presented, and within several link limits a linguist is able to quickly correct the algorithm suggestion, e.g. if LAAA suggests for *molybdenum* a synset including *element* (chemical sense) – this is only one click from the right place. Three evaluation criteria were applied. The first one called *closest path* focuses only

on one attachment proposition that is closest to the original location of a test word. The second, *strongest*, takes into account only the highest score proposition returned by an algorithm.

Only two KSs were used for both compared algorithms. The first one is based on hypernym only classifier presented in [8]. First we parsed Wikipedia corpus with dependency parser *Minipar* and then we extracted all patterns between two nouns from dependency graphs. Every pattern that occurred at least five times between an unique word pair defined a feature for logistic regression classifier from *LibLINEAR*. The second KS is cousin classifier based on distributional similarity in a similar way to [7]. We trained logistic regression classifier using only MSR values to predict  $(m, n)$ -cousin relationship between words. Classifier was trained to recognize the following classes:  $0 \leq m, n \leq 3$  and negative class which indicates more distant cousin relationship or not at all. We tried to follow directly the approach of [7], but many details were not described, e.g. the clustering method used, so we used a classifier based on the same information instead. In our experiments we applied MSR as an input feature to the logistic regression classifier. MSR was calculated as a cosine similarity between word distributional vectors: each vector element corresponds to the frequency of co-occurrences with other words in the selected dependency relations.

PWE [7] performance strongly depends on values of predefined parameters (there is no procedure defined). We tested several combination of values, and selected the following: *minimal probability of evidence*: 0.1, *inverse odds of the prior*:  $k = 4$ , *maximum size of the cousins neighbourhood*:  $(m, n) \leq (3, 3)$ , *maximum links in hypernym graph*: 10, *penalization factor*:  $\lambda = 0.95$ .

As a baseline we used a simple algorithm that applies directly MRS: for a new lemma  $x$  the most similar one  $y$  is identified and looked up in WordNet. The first synset found is returned as the only one attachment point for  $x$ . As we have only two knowledges sources (for the comparison with PWE), both of probabilistic nature and both producing values from the similar range. So, we set global weights to 1, and used local weights provided by the KSs.

Transmittance was used to separate those relations that can participate in the paths defining local context from the others. Almost all selected relations received transmittance of the form  $f_T(r, v) = 1 * v$ , for some meronymy subtypes  $f_T(r, v) = 0.7 * v$ , and for the non-selected  $f_T(r, v) = 0$ . Transmittance can be statistically tuned on the basis of the correlation of the support values, e.g. a MSR observed on both ends of relation links, however for the sake of presentation simplicity we omitted this tuning here. The selected relations are: hyper/hyponymy, holo/meronymy and antonymy (synsets represent synonymy). Impedance allows to control the shape of the local context from which the indirect support is collected. In its simplest form the impedance function returns 1/0 for relation pairs, i.e. opens or block passage for support replication. The ‘zero’ impedance (blocking transfer) was heuristically selected for the following pairs (relation in, relation out):  $\langle \text{hypo.}, \text{anto.} \rangle$ ,  $\langle \text{hypo.}, \text{mero.} \rangle$ ,  $\langle \text{hyper.}, \text{hypo.} \rangle$ ,  $\langle \text{hyper.}, \text{holo.} \rangle$ ,  $\langle \text{anto.}, \text{anto.} \rangle$ ,  $\langle \text{anto.}, \text{mero.} \rangle$ ,  $\langle \text{anto.}, \text{holo.} \rangle$ ,  $\langle \text{mero.}, \text{anto.} \rangle$  and  $\langle \text{holo.}, \text{anto.} \rangle$ . We set the maximal value of  $rad$  to 10, following PWE. Smaller values of  $rad$  can lead to the improved precision, but reduced recall.

**Table 1.** Comparison of the baseline algorithm, LAAA and PWE precision on WordNet 3.0

Method	Hits distance							total
	0	1	2	3	4	5	6	
Baseline								
<b>Closest [%]</b>	15.4	15.6	2.4	0.2	0.1	0.1	0.0	33.7
<b>Strongest [%]</b>	8.7	5.7	1.0	0.2	0.0	0.3	0.0	15.6
<b>All [%]</b>	2.6	5.9	3.5	1.7	0.9	0.4	0.0	9.3
PWE								
<b>Closest [%]</b>	10.5	15.9	3.5	1.0	1.2	0.7	0.1	32.9
<b>Strongest [%]</b>	6.3	6.1	2.1	0.8	0.7	0.0	0.0	16.0
<b>All [%]</b>	2.3	4.7	1.8	0.8	0.5	0.2	0.0	10.4
LAAA: MSR + hyponym classifier only								
<b>Closest [%]</b>	3.7	14.6	9.1	8.3	5.7	4.9	0.0	46.3
<b>Strongest [%]</b>	2.7	7.3	8.4	8.9	7.1	5.5	0.1	40.0
<b>All [%]</b>	0.9	4.2	2.4	2.1	1.6	1.3	0.0	12.5
LAAA: MSR + hyponym classifier + cousins								
<b>Closest [%]</b>	5.1	19.4	8.7	7.9	5.3	4.2	0.0	50.6
<b>Strongest [%]</b>	3.4	12.0	8.5	8.6	6.7	4.9	0.1	44.0
<b>All [%]</b>	1.2	5.6	2.4	2.0	1.5	1.1	0.0	13.8

## 5 Experiments and Results

After initial experiments done on a different set of lemmas for setting up the parameters, we performed three experiments on the test sample for the Baseline, PWE and LAAA, see Tab. II. We can observe that the simple Baseline algorithm performs well, especially for direct attachments. PWE achieved slightly better results than the baseline for the *strongest* and *all* evaluation strategies. This was caused by the fact that baseline algorithm does not perform any sense disambiguation.

LAAA depends on the quality of the KSs employed. This can be seen in Tab. II, where addition of cousins classifiers improved the results for every evaluation strategy. LAAA is characterised by the very high value of strongest attachments. That means, that the scores produced by LAAA are trustworthy. PWE has a higher coverage than LAAA (93% vs 89%), i.e., PWE returned propositions for more words than LAAA. On the other hand, LAAA returned significantly more senses on average than PWE (24.5% vs 16.3%). As both algorithms are designed for expansion of existing wordnets, the slight decrease in coverage is not as important as the increase in sense recall of the algorithm.

## 6 Conclusions

We presented a new wordnet expansion algorithm called Lexical Area Attachment Algorithm based on an idea of combining different heterogeneous KSs and correcting their errors by expanding their support across local wordnet subgraphs. LAAA can use any KS, as it does not assume the probabilistic character of KSs. They can be also partial

as LAAA does not assume that all links are described as it is done in the formal model of PWE [7] (later in PWE algorithm this assumption is not kept). Contrary to [11] we do not assume any shape of the lexical semantic network, but we try to build it in a way following the language data. We aim also at an unsupervised or very weakly supervised algorithm in which training is limited to finding only general properties of the wordnet relations. LAAA was compared with a baseline algorithm and the seminal PWE on Princeton WordNet 3.0 using evaluation methodology defined in [1]. PWE served as a reference point in many works and was hardly beaten by [11]. Our baseline uses MSR values of to attach new words to wordnet synsets. In all conducted experiments LAAA performed significantly better.

**Acknowledgments.** Work partially financed by the Polish Ministry of Education and Science, Project N N516 068637.

## References

1. Broda, B., Kurc, R., Piasecki, M., Ramocki, R.: Evaluation Method for Automated Wordnet Expansion. In: Bouvry, P., Kłopotek, M.A., Leprévost, F., Marciniāk, M., Mykowiecka, A., Rybiński, H. (eds.) SIIS 2011. LNCS, vol. 7053, pp. 293–306. Springer, Heidelberg (2012)
2. Fellbaum, C. (ed.): WordNet — An Electronic Lexical Database. The MIT Press (1998)
3. Israel, G.: Determining sample size. Tech. rep., University of Florida (1992)
4. Kozareva, Z., Hovy, E.: A semi-supervised method to learn and construct taxonomies using the web. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, MIT, Massachusetts, USA, October 9–11, pp. 1110–1118. ACL (2010)
5. Navigli, R., Velardi, P., Faralli, S.: A graph-based algorithm for inducing lexical taxonomies from scratch. In: Proceedings of IJCAI (2011)
6. Piasecki, M., Szpakowicz, S., Broda, B.: A Wordnet from the Ground Up. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2009)
7. Snow, R., Jurafsky, D., Ng, A.Y.: Semantic taxonomy induction from heterogenous evidence, pp. 801–808. The Association for Computer Linguistics (2006)
8. Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. In: NIPS (2004)
9. Widdows, D.: Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In: Proc. HLT of North American Chapter of the ACL (2003)
10. Witschel, H.F.: Using decision trees and text mining techniques for extending taxonomies. In: Proc. of Learning and Extending Lexical Ontologies by using Machine Learning Methods, Workshop at ICML 2005 (2005)
11. Yang, H., Callan, J.: A metric-based framework for automatic taxonomy induction. In: Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp. 271–279. ACL (2009)

# Metaheuristics for Tuning Model Parameters in Two Natural Language Processing Applications

Łukasz Kłyk<sup>1</sup>, Paweł B. Myszkowski<sup>1</sup>, Bartosz Broda<sup>1</sup>, Maciej Piasecki<sup>1</sup>,  
and David Urbansky<sup>2</sup>

<sup>1</sup> Institute of Informatics

Wrocław University of Technology

Wyb. Wyspiańskiego 27, 51-370 Wrocław, Poland

{Pawel.Myszkowski,Bartosz.Broda,Maciej.Piasecki}@pwr.wroc.pl

<sup>2</sup> TU Dresden

Department of System Engineering

01062 Dresden, Germany

davidurbansky@googlemail.com

**Abstract.** Choosing model parameters is an important issue for solving real word problems. Wrong parameter values result in low performance of employed model. Usually, parameters are chosen manual, but one can employ metaheuristics for searching the parameter space in more systematic and automated way. In this paper we test a few optimisation methods such as Evolutionary Algorithms, Tabu Search, Hill Climbing and Simulated Annealing for setting parameters of models in two problems in the domain of Natural Language Processing. Metaheuristics used significantly improve performance in comparison to the default parameter selected manually by domain experts.

**Keywords:** Classification, Parameters Tuning, Metaheuristics.

## 1 Introduction

The efficiency of classification model is influenced by several factors. The first and foremost, one have to choose appropriate classification method and features. The parameters of classifiers should be fine-tuned to the problem domain. Each classifier model has its own parameters that need to be optimised for the data at hand in order to achieve best classification accuracy.

Typically some experiments are performed for fine-tuning of the model parameters. The more systematic approach to selection of parameter values, the better. In many problems the search space is very large so choosing optimal parameters by hand is very difficult. Moreover, exhaustive searches are not possible, because either single experiment is time-consuming, or the search space is too big. Thus, one have to employ heuristic methods for fine-tuning model parameters, e.g., [12]. Usually, those approaches do not guarantee finding an optimal set of parameters. Nevertheless, using a systematic method for selection of parameter values typically improves performance of the model.

## 2 Related Work

There are many possible approaches to model parameters optimisation [3]. We have chosen a few of them, which focus on some practical applications. In [1] a parameter selection method by based on minimizing the estimated error is described. The employed method uses a similar approach to presented in this work, called a *wrapper*. The authors of [1] use best first search in order to find best parameters for the model and cross-validation for the evaluation of a given parameter configuration.

In [4] Evolutionary Algorithms are applied to SVM classifier optimisation. In [5] SVM parameters are optimised too, and the applied method combines knowledge about the internals of SVM with a standard gradient descent approach. A more general work is presented in [6], where a framework for the optimisation of parameters of any classifier is proposed. The goal is achieved by the specification of a set of input parameters and a set of classifier parameters.

## 3 Applied Optimisation Methods

The main idea of the approach proposed here is to optimise the model parameters by applying metaheuristics. By using this universal approach we can ignore an internal structure of a given classifier. To run metaheuristics we need a classifier model, domains of its parameters, an evaluation measure  $f$  and a set of training (and testing) data. Our task is to set optimal values of parameter to gain  $f$  improvement. Such an approach is similar to the manual selection of parameters values. The difference is that we use metaheuristics which offers a more systematic and automated way of the parameter space exploration. In this paper we target four classical optimisation methods: Evolutionary Algorithms (EA), Hill Climbing (HC), Tabu Search (TS) and Simulated Annealing (SA) [3].

## 4 Applications

### 4.1 Dictionary-Based Classifier

A dictionary-based classifier [7] (DBC) performs classification on the basis of *n-grams*. N-grams are token sequences of the length  $n$ . Two types of n-grams can be distinguished: *character level n-grams* and *word level n-grams*. *Character level n-grams* consist of string characters as tokens. *Word level n-grams* include words (separated with white space) as tokens.

The number of *n-grams* that are generated for the classifier can be determined by the equation:  $ngrams = noTokens - n + 1$ . It is possible to create a set of *n-grams* with different lengths, between MinNGramLen and MaxNGramLen.

The DBC classifier defines how probable each *n-gram* is for each given category and assigns the most probable category (or categories) to an input document. There are two external parameters that affect the results and efficiency of classification: **n-grams type**: word or character; **minimum n-grams length**:

$(0, \infty)$ ; **maximum n-grams length**:  $(0, \infty)$ . The minimum and maximum *n-grams* length were limited to  $< 1, 20 >$  in our experiments. A domain expert defined the default values as: **n-gram type**: character; **minimum n-gram length**: 3, **maximum n-gram length**: 7. The standard F1 measure was chosen as an evaluation measure, i.e., calculated on the basis of the resulting classifier precision and recall. F1 equalled to 0.71187 for the default parameter values.

## 4.2 Algorithm of Activation-Area Attachment

A Lexical Level AAA (LL-AAA) is a version of Algorithm of Activation-area Attachment (AAA) for automated wordnet [8] expansion of top performance [9]. However, AAA has a heuristic character and it utilises only synset relations (cf [8] for wordnet structure). LL-AAA works on the graph of lexico-semantic relations (word to word), is based on a *support spreading model* and implements a more transparent model of support replication across wordnet graph nodes than AAA. It takes lemmas that are not in the wordnet and suggests for them synsets to which they can be attached as synonyms or hyponyms.

LL-AAA works on the basis of *knowledge sources* that are extracted from text collections and includes word pairs – potential instances of hyponymy or synonymy. For a new word  $x$ , all word pairs including  $x$  from knowledge sources are utilised to find synsets *supported* by this data. As knowledge sources express some error the main idea of LL-AAA is to calculate support for placing  $x$  in a synset  $S$  as a sum of *local* support for words represented in  $S$  calculated *directly* on the basis of the accuracy of knowledge sources and *contextual* support collected *indirectly* from synsets linked to  $S$  by short paths. The indirect support compensates errors in knowledge sources. LL-AAA model is based on a simple rule: support for linking a lemma  $x$  to a LU  $l$  in the wordnet is collected from knowledge knowledges sources and next spread across the local context of  $l$  defined by LUs linked to  $l$  by short paths ( $x$  is a new lemma,  $L$  is a set of wordnet LUs,  $Syn$  – a set of synsets,  $\mathbf{Q}_0$ ,  $\mathbf{Q}$  – support matrices):

**Step 0** Setting up local support:

$$\forall_{l \in L} \mathbf{Q}_0[l] = \text{sum of weights for } \langle x, l \rangle \text{ in the knowledge sources}$$

**Step 1** Support spreading:  $\forall j \in L \text{ if } flat(\mathbf{Q}_0)[j] > \tau \text{ fitFlow}(j, x, \mathbf{Q}_0[j])$

$fitFlow(j, x, M)$  is defined as

1. **if**  $M < \epsilon$  **then return**
2. **for each**  $p \in dsc(j)$   $fitFlowTrans(p, x, (\mu * M) |dsc(j)|)$
3.  $\mathbf{Q}[j] = \mathbf{Q}[j] - \mu * M$

and  $fitFlowTrans(p, x, M)$  as

1. **if**  $M < \epsilon$  **then return**
2. **for each**  $p' \in dsc(p|_1)$   $fitFlowTrans(p', x, (\mu * M) / |dsc(p|_1)|)$
3.  $\mathbf{Q}[p|_1] = \mathbf{Q}[p|_1] + (1 - \mu) * M$

**Step 2** Identification of attachment areas

1. Identification of *activation areas*: linked synset subgraphs such that  $G_m = \{s \in Syn : synsetFit(s) > 0\}$
2. **foreach**  $G_m$   $score(G_m) = synsetFit(j_m)$ ,  $j_m = max_{j \in G_m}(synsetFit(j))$
3. Attachment areas are returned: those  $G_m$  for which  $score(G_m) > \tau$

In **Step 1** the local support is spread across local contexts of words. Spreading (*fitFlow*) is continued as long as the replicated support portion is larger than  $\epsilon$ ;  $\mu$  (the *viscosity factor*) defines how much support is taken from the source word, and what part of support is kept in each word passed on the path. If  $\mu < 1$ , then replicated support is decreased along the path. In each word, the outgoing support is divided proportionally among all output links. In **Step 2** support for words is transformed into support for synsets by *synsetFit* – a parameter, here,  $synsetFit(s)$  is the sum of support for words belonging to the synset  $s$ . Finally, *activation areas* – linked subgraphs such that each node (synset) has non-zero synset support – are identified and several of them with the top support are returned as suggested *attachment areas*.

## 5 Experimental Results

Results of used metaheuristics are presented in this section. The default configuration of **EA** was: mutation prob.  $P_m=10\%$ , crossover prob.  $P_x=60\%$ , population size=15 and 25 generations. For **TS**: tabu list size=50, neighbourhood size=15 and 25 steps. The **HC** uses: neighbourhood size=15 and 25 steps. The **SA** works with temperature  $T(k) = 0.6^k$  for 374 steps. Such a configuration was set for DBC and AAA problem. However, in the AAA due to the computation limitations methods work in a shorter time (uses only 40 steps) and it is given respectively as generation/steps parameters values.

### 5.1 Dictionary-Based Classifier

Experiments were performed with four heuristic algorithms. As the search space was limited by the domain expert, we could apply exhaustive search in order to find a (global) optimum of parameter values. If we constrain the NGramLen (up to 20) the total number of solutions is limited to 420. It takes less than two hours to check all of the combinations (using Intel i7 2.67Ghz). We can check if the applied metaheuristics can select parameter values close to the optimal ones. If so, we can extend the parameter space and apply the proposed methodology to larger datasets or other problems. Such a parameter space can be plotted. There is a global maximum  $F1=\mathbf{0.75809}$  for the following parameter configuration: n-gram type=character; minimum n-gram length=7; maximum n-gram length=9. There is also local maximum  $F1=\mathbf{0.74445}$  for the given configuration: n-gram type=character; minimum n-gram length=4; maximum n-gram length=20. This local maximum was found by every method during the experiments. Generally, the value of F1 is higher for the *character* value.

Results of the experiments, presented in Table II, shows that metaheuristic work effectively where each of method obtained a satisfactory value of  $F1_{max} = 0.74445$  (default is given in section 3). In this test HC also gives the highest average  $F1 = 0.73891$ . Another advantage of HC is lower value of standard deviation than in the other tested methods. Time taken to find the final solution is similar to an exhaustive search time.

**Table 1.** Results for Dictionary-Based Classifier (averaged from 10 runs)

Method	$F1_{av}$	$F1_{min}$	$F1_{max}$	$\sigma F1$	$t_{av}[min]$	$\sigma t[min]$	no. of evaluate
EA	0.72484	0.66566	<b>0.74445</b>	0.0254	78.3529	21.9841	375
TS	0.73475	0.70083	<b>0.74445</b>	0.0126	83.3977	3.23780	376
HC	<b>0.73891</b>	0.73278	<b>0.74445</b>	<b>0.0036</b>	81.6022	3.70555	376
SA	0.73610	0.70083	<b>0.74445</b>	0.0126	82.9046	2.64182	375

## 5.2 Algorithm of Activation-Area Attachment

Evaluation was based on the methodology defined in [9]: first a sample of lemmas is removed from a wordnet, next LL-AAA is applied to automatically to re-attach the removed lemmas to the wordnet. Attachment suggestions are sorted by their support values. For evaluation we built a histogram of path lengths between suggested synsets and their original positions in a wordnet. Paths of up to 5 links, including hyper/hyponymy links with at most one final meronymic were considered. Longer distances were discarded as not useful for linguists.

The following evaluation criteria were applied: *strongest* suggestion takes into account only the highest score proposition returned by the algorithm and coverage of the algorithm. In order to speed up the optimisation only one knowledge source was used. It was based on a distributional similarity measure (DSM) constructed on the basis of a corpus. The knowledge source included pairs:  $\langle x, y \rangle$ , such that  $y$  is among the top most similar lemmas to  $x$  and *vice versa*.

Default parameters for LL-AAA were chosen by the domain expert by hand in a series of experiments.:  $\tau=1.8$ ,  $\epsilon=0.1$ ,  $\mu=0.5$  and the evaluation measures for this configuration are: *strongest proposition* is 13.7. For optimisation we used an *F measure* calculated as a harmonic mean of *strongest proposition* and *coverage*. The *strongest proposition* shows how good are the choices made by the LL-AAA algorithm in relation to its internal scoring system. We need to balance *strongest* with *coverage*. For default parameters the value of *F measure* is **0.200769**.

Table 2 shows the averaged results (we used Intel Xeon X5650 2.67GHz) of the described methods. The same statistics coefficients as in previous test are used. Almost every method is able to improve the default. The exception is EA for which the best result is close to the default. TS is the most effective method because it obtained the highest evaluation function value **0.257586**. Also, the average *F* is the highest for TS.

**Table 2.** Results for Activation-area Attachment method (averaged 10 runs)

Method	$F_{av}$	$F_{min}$	$F_{max}$	$\sigma F$	$t_{av}[s]$	$\sigma t[s]$	no. of evaluate
EA	0.197829	0.173733	0.226367	0.016108	543.24	70.44	40
TS	<b>0.203997</b>	0.173733	<b>0.257586</b>	0.026461	582.44	85.49	41
HC	0.200921	0.173733	0.241536	0.020067	564.43	78.45	41
SA	0.199763	0.173733	0.254981	0.024166	567.56	130.85	40

The above results are better with respect to every measure than the baseline. However, LL-AAA, especially run only for one knowledge source, presents much lower results than original AAA, cf [9]. Limited spreading of support was also tested in a version of AAA and we could also observed a slight decrease in the performance. The differences in values of the three parameters are very interesting. The main problem of the spreading model seems to be the diminished influence of the indirect support – in every node support is equally divided among the output links and drops down very quickly. The optimisation tried to correct this problem:  $\mu=0.9$  means that more support is transferred to the context,  $\epsilon=0.001776$  expands the area of the local context and  $\tau=0.351105$  decreases the barrier for including a synset into an attachment area. It took us several time consuming experiments to come to the similar observations manually.

## 6 Conclusions

Choosing model parameters values is an important issue for solving real word problems. The improper parameter values result in low performance of classifier. In this paper we tested metaheuristic approaches to model parameters selection. We employed EA, TS, HC and SA in the domain of NLP. In both NLP application, i.e., DBC and LL-AAA we have achieved a significant improvement in performance over a baseline models. In baseline models parameters were set manually by domain experts.

In further works we plan to run more through search of parameter space of LL-AAA. We also plan to test other datasets with LL-AAA. The work on DBC can be extended also with larger datasets and more diverse features.

**Acknowledgments.** Work partially financed by the Polish Ministry of Education and Science, Project N N516 068637.

## References

1. Kohavi, R.: Automatic parameter selection by minimizing estimated error. In: Proc. of 12th Twelfth ICML 1995, p. 304 (1995)
2. Wang, J., Downs, T.: Tuning pattern classifier parameters using a ga with an application in mobile robotics. In: Cong. CEC 2003, pp. 581–586. IEEE (2003)
3. Michalewicz, Z., Fogel, D.B.: How to solve it: modern heur. Springer (2004)
4. Chunhong, Z., Licheng, J.: Automatic parameters selection for svm based on ga. In: 5th Word Congr. WCICA 2004, vol. 2, pp. 1869–1872. IEEE (2004)
5. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for svm. Machine Learning 46(1), 131–159 (2002)
6. Müssel, C., Lausser, L., Maucher, M., Kestler, H.: Multi-objective parameter selection for classifiers. Journal of Statistical Software 46(i05) (2008)
7. Urbansky, D., Muthmann, K., Katz, P., Reichert, S.: Tud palladian overv. (2011)
8. Fellbaum, C. (ed.): WordNet — An Electr. Lexical Database. The MIT Press (1998)
9. Broda, B., Kurc, R., Piasecki, M., Ramocki, R.: Evaluation Method for Automated Wordnet Expansion. In: Bouvry, P., Kłopotek, M.A., Leprévost, F., Marciniak, M., Mykowiecka, A., Rybiński, H. (eds.) SIIS 2011. LNCS, vol. 7053, pp. 293–306. Springer, Heidelberg (2012)

# Clustering a Very Large Number of Textual Unstructured Customers' Reviews in English

Jan Žižka, Karel Burda, and František Dařena

Department of Informatics, FBE, Mendel University in Brno

Zemědělská 1, 613 00 Brno, Czech Republic

{zizka,burda,darena}@mendelu.cz

**Abstract.** Having a very large volume of unstructured text documents representing different opinions without knowing which document belongs to a certain category, clustering can help reveal the classes. The presented research dealt with almost two millions of opinions concerning customers' (dis)satisfaction with hotel services all over the world. The experiments investigated the automatic building of clusters representing positive and negative opinions. For the given high-dimensional sparse data, the aim was to find a clustering algorithm with a set of its best parameters, similarity and clustering-criterion function, word representation, and the role of stemming. As the given data had the information of belonging to the positive or negative class at its disposal, it was possible to verify the efficiency of various algorithms and parameters. From the entropy viewpoint, the best results were obtained with  $k$ -means using the binary representation with the cosine similarity,  $idf$ , and  $H2$  criterion function, while stemming played no role.

**Keywords:** cluster mining, textual data, entropy, term representation, clustering criterion function, similarity, customer opinion.

## 1 Introduction

Many activities in the real, practical world are connected with collecting data that have the textual form. Such data can represent various opinions associated with a certain activity, therefore providing many specific individual standpoints of the activity itself. Those specific standpoints can usually be categorized to provide information and – after generalization – valuable knowledge giving a true picture of the activity, which can be later used for modifications leading to improvements. In this paper, the authors describe a situation when customers of the service that enables on-line booking of hotel accommodations using the Internet connection anywhere can later provide their positive or negative opinions of their accommodation using again the Internet and writing their reviews in any natural language without applying any compulsory structure or form. Expressing such opinions is very easy for anyone who can use an Internet browser, however, because such data is typically human-like, it is not an easy task to process it for machines (computers).

If the service provider needs, for example, to choose negative reviews to reveal what the customers complain about, and than use it as feedback, one of the questions is: “Can a machine automatically separate certain opinions from the whole review collection?” People can do it, however, if the data volume is very large, it is impossible to get the results within an acceptable time. Simultaneously, the more reviews are available, the more valuable knowledge can generally be mined from such data. After splitting the original unformatted non-labeled reviews between relevant categories (classes), various classification and prediction methods could then be used to mine useful knowledge, see for example [17], which dealt with the same textual data when looking for significant category words and phrases.

This study aimed at finding how reliable results can be obtained from the automatic categorization using the unsupervised learning called *clustering*. Clustering was successfully used for organizing and searching large text collections, for example, in automatic creation of ontologies, summarization, disambiguation, and navigation of results retrieved by search engines [2]. In recent years, document clustering was used also for sentiment analysis and opinion mining, as mentioned in [4] or [11]. The research presented here took advantage of knowing which customers’ reviews belonged to which class: *negative* and *positive*. This knowledge was used for evaluating the quality of clustering results – how many positive reviews were assigned to the correct class and how many to the contrary one.

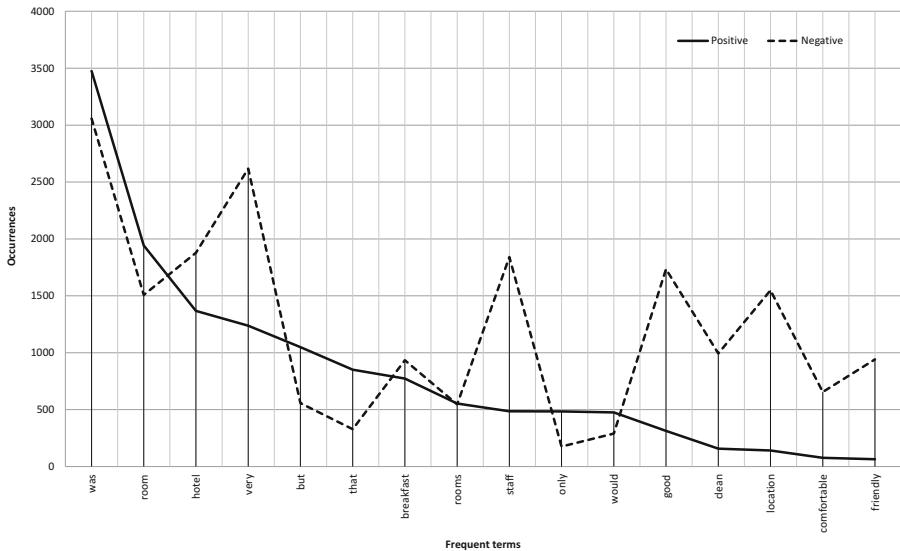
The following sections describe experiments and their results with large real-world textual data. The authors used almost 2,000,000 customers’ reviews written freely in English. Firstly, the properties of the review collection are briefly presented, followed by a brief description of the applied clustering methods. Then, the design of experiments using the review data and selected clustering algorithms is provided. In the final section, the results of experiments are depicted, including interpretation and a brief discussion.

## 2 Clustering the Textual Data

The text data used in the experiments was a subset from data described in [17], containing customers’ opinions written in many languages of several millions customers who – via the on-line Internet service – booked accommodations in many different hotels and countries. The subset used in our experiments contained almost two million opinions written only in English by customers who made reservations through the web and who really stayed in the hotel. Typically, most of reviews embodied all deficiencies typical for texts written in natural languages (mistypings, transposed letters, missing letters, grammar errors, sometimes combinations of two languages in one item, and so like). In Table II, an interested reader can find some basic statistic values of the investigated data.

It is also worth to show that the review clustering problem, based on words without keeping their original mutual connections or succession (*quite bad accommodation, not so bad accommodation*, and so like) that provide the right

meaning of phrases, is not an easy task. Figure 1 illustrates it well: for a random selection of 5,000 reviews from each class (*positive* and *negative* reviews), a reader can see that both classes contain the same significant words. For example, the ‘negative’ terms like *not*, *bad*, *dirty*, and so on, can be also in positive reviews, however, the right meaning is naturally given by connections with other terms. This is a good reason for searching for algorithms and methods that would find not only significant words [17] but also significant phrases [18].



**Fig. 1.** The most frequent words both in the negative (the dashed curve) and positive reviews. The occurrence (frequency) can sometimes be similar, sometimes different, however, it itself is not the only decisive factor.

Clustering algorithms can be generally divided into two categories: *hierarchical* and *partitional* (flat) clustering. Although hierarchical algorithms can get higher precision, they are not always suitable for sets with a large number of items because of their relative slowness [10]. In the text mining literature, various classical clustering algorithms such as *k*-means (and its variants), hierarchical agglomerative clustering, and graph-theoretic methods have been explored [2].

One of the simplest and often used clustering methods is *k-means* [14]. Essentially, using the available training instances, this algorithm iteratively adapts the initial randomly generated *k* cluster centroids’ positions. The *repeated bisection* clustering method is basically a sequence of cluster bisections. The initial collection of documents is firstly bisected. Then, one of the two clusters is selected and it is further bisected, leading to a total of three clusters. The process of selecting and bisecting a particular cluster continues until *k* clusters are obtained [8]. *Graph-partitioning* based clustering comes out from partitioning a graph representing objects to be clustered. Vertices of the graph represent the documents

**Table 1.** Properties of data used for experiments

Review category:	<i>positive</i>	<i>negative</i>
Number of reviews:	1, 190, 949	741, 092
Maximal review length:	391 words	396 words
Average review length:	21.67 words	25.73 words
Variance:	403.34 words	618.47 words

and links their relationships in terms of similarity. The graph is subsequently split into clusters using the min-cut graph partitioning algorithm [8].

The *similarity measure* is a key factor influencing the clustering process and depends on the specific problem. In this research, the following three similarity methods were used: The *cosine similarity* measures the cosine of the angle between couples of vectors representing the documents. When both documents are identical, the value of cosine similarity equals 1, and 0 when the documents are completely dissimilar (the vectors are orthogonal to each other). *Euclidean distance* computes the distance between points representing documents in the abstract space. *Pearson's correlation coefficient* measures linear correlation between values of two vectors [6].

*Cluster validation* is important part of the clustering process. It includes procedures evaluating the results of clustering in a quantitative fashion [15]. External quality measures use category labels for measuring the correctness of clustering. *Entropy*, *purity*, *mutual information*, or *F-measure* were frequently used in evaluating document clustering as the external measures [5].

### 3 Design of Experimental Examinations of the Reviews

In the experiments, the clustering process used the software package Cluto 2.1.2 and gCluto (graphical extension of Cluto 2.1.1) available at the URL referenced in [1]. Cluto 2.1.2 was used when clustering was based on Euclidean similarity measure (gCluto did not implement this feature). This free software provides different clustering methods working with several clustering criterion functions and similarity measures, and it is suitable for operating on very large datasets [8].

The textual data needed to be transformed into a representation acceptable by the clustering algorithm implementation. The data might be structured according to the level on which it is analyzed, from sub-word level to pragmatic level. In many cases, words are meaningful units of little ambiguity even without considering a specific context. A big advantage of word-based representations is their simplicity and straightforward process of creation [7].

Every document was therefore simply transformed into a *bag-of-words*, a sequence of words where the ordering was irrelevant, thus providing the standard vector form. The space dimensions were represented by weights of individual attributes (words) of the text. Several possible methods for determining the weights of the words can be used [12]:

- the weights are binary (0 or 1), representing the presence or absence of the term (Term Presence,  $TP$ ),
- the weights correspond to the word frequency in the text (Term Frequency,  $TF$ ),
- $TP$  and  $TF$  weights can be weighted using inverse document frequency, calculated as  $idf(t_i) = \log N/n(t_i)$ , where  $t_i$  is the term,  $N$  is the number of documents in the collection, and  $n(t_i)$  is the number of documents containing term  $t_i$ .

The quality of the vector representation can be further increased using n-grams, semantics enhancement, removal of very frequent or very infrequent words, stop words elimination, stemming, and others. Although using those procedures can influence the results of text mining algorithms, their effects are often marginal [3]. The described experiments analyzed the role of stemming as the most widely applied morphological technique for information retrieval [9]. As one of Cluto's additional tool provides the stemming functionality, Porter's stemming [13] was put in use. To evaluate the impact of stemming, experiments without it were also carried out. During the preprocessing phase, words having their length  $\leq 3$  characters were automatically removed as well (it included definite/indefinite articles, prepositions, and so like).

For the experiments, several smaller subsets of data were selected. The intention was to select subsets with approximately the same number of positive and negative reviews to get as balanced clusters as possible. The data were selected randomly and every document could have been used only once. Altogether, 14 experiments were run with data subsets of different sizes. The ratios of positive (P) and negative (N) reviews, P:N, in the experiments were as follows: 131:144, 229:211, 987:1029, 1031:1085, 2096:2211, 4932:4757, 4832:4757, 7432:7399, 10023:8946, 10251:9352, 15469:14784, 24153:23956, 52146:49986, and 365921:313752. Afterwards, several algorithms and their settings were applied to the same data in each experiment. The following four steps had to be performed in every experiment:

1. Random selection of desired amount of reviews.
2. Transformation of data into the vector representation.
3. Loading data in Cluto and performing clustering.
4. Evaluating the results.

During the experiments, certain clustering parameters were being investigated (in the following,  $I_i$  stands for an internal criterion function,  $E_i$  for an external criterion function, and  $H_i$  for a hybrid criterion function; for details, see [16]):

- similarity: cosine, Pearson's correlation coefficient, Euclidean distance;
- clustering: repeated bisection, Cluto's  $k$ -means variation, and graph-based;
- criterion functions that are minimized/maximized during optimization of clustering:  $I2$ ,  $H2$ ,  $I1$ ,  $E1$ ;
- document vector representation models: Term Presence ( $TP$ ), Term Frequency ( $TF$ ), Term Frequency×Inverse Document Frequency ( $TF-IDF$ ), Term Presence×Inverse Document Frequency ( $TP-IDF$ ).

**Table 2.** Weighted entropy for different experimental settings

Ratio (P:N)	k-means				repeated bisection			
	TF-IDF		TP-IDF		TF-IDF		TP-IDF	
	I2	H2	I2	H2	I2	H2	I2	H2
131:144	0.792	0.785	0.793	0.741	0.726	0.767	0.774	0.774
229:211	0.694	0.632	0.695	0.627	0.648	0.643	0.650	0.647
987:1029	0.624	0.610	0.618	0.605	0.624	0.609	0.618	0.611
4832:4757	0.601	0.581	0.599	0.579	0.600	0.584	0.598	0.580
7432:7399	0.605	0.596	0.599	0.587	0.605	0.595	0.594	0.586
15469:14784	0.604	0.583	0.598	0.579	0.604	0.582	0.598	0.579
24153:23956	0.597	0.580	0.589	0.572	0.597	0.580	0.589	0.572
52164:49986	0.596	0.582	0.600	0.573	0.604	0.582	0.598	0.574
201346:204716	0.599	0.583	0.592	0.575	0.597	0.583	0.593	0.576
365921:313752	0.602	0.586	0.598	0.584	0.599	0.581	0.598	0.580

Other Cluto's clustering parameters remained set to their default values, that is, *Number of iterations* = 10, *Number of trials* = 10, *Cluster selection*: Best, and *Row model*: none. To measure the quality of clustering results, Cluto provides *purity* and *entropy* based measures. The experiments used the entropy weighted according to the cluster sizes [16]:

$$\text{Entropy} = \sum_{r=1}^k \frac{n_r}{n} \frac{-1}{\log q} \sum_{i=1}^q \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r},$$

where  $q$  is the number of classes in the data set,  $n_r^i$  is the number of documents in the  $i$ -th class that were assigned to the  $r$ -th cluster, and  $n_r$  is the size of the  $r$ -th cluster.

## 4 Results of Experiments

The results are summarized in Table 2 which shows the values of *weighted entropy* for different experimental settings. Only the best results that were achieved by *k-means*, repeated bisection, and cosine similarity are demonstrated.

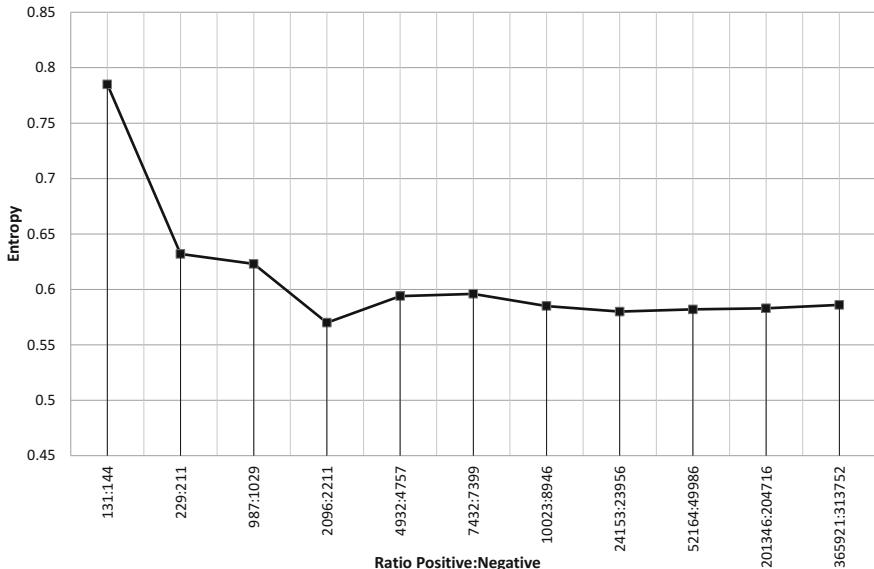
Table 3 shows percentage ratios of documents in the clusters after clustering for selected experimental settings.

Figure 2 illustrates the entropy change by increasing the number of documents. A certain boundary from which the entropy value oscillates and does not change much with increasing number of documents to be clustered was around 10,000 documents, as shown in Figure 2.

Weighting the vector attribute values using the *idf* weight had a considerable positive impact on clustering results in comparison with simple *TP/TF* representation that performed poorly. The *TF-IDF* document representation provided almost the same results as *TP-IDF*. This was the result of a high number of

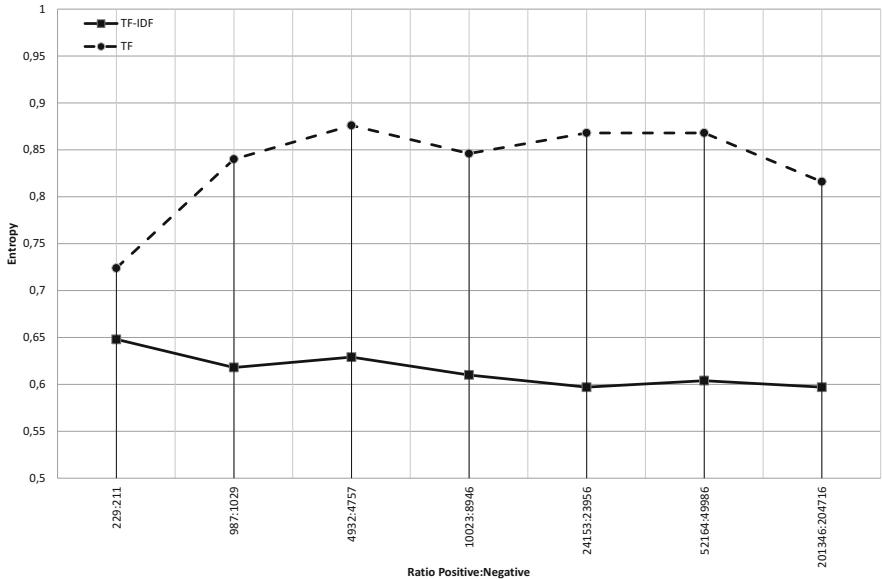
**Table 3.** Ratios of documents in clusters for different experimental settings

Ratio (P:N)	K-means				Repeated bisections			
	I2		H2		I2		H2	
	clust. 0 (P:N)	clust. 1 (P:N)	clust. 0 (P:N)	clust. 1 (P:N)	clust. 0 (P:N)	clust. 1 (P:N)	clust. 0 (P:N)	clust. 1 (P:N)
131:144	76:24	24:74	78:24	22:74	75:22	25:76	78:19	22:78
229:211	84:21	16:79	86:18	14:82	84:20	16:80	84:18	16:82
987:1029	80:12	19:87	85:16	14:83	79:11	20:88	85:15	15:84
4832:4757	83:13	17:87	87:15	13:85	83:12	17:87	86:14	14:86
7432:7399	82:12	17:87	85:14	14:85	82:12	17:86	86:14	14:85
15469:14784	80:11	19:89	85:13	15:86	81:10	19:89	85:13	15:87
24153:23956	81:11	19:89	85:13	14:86	81:10	18:89	86:13	14:87
52164:49986	18:89	81:11	15:87	85:13	19:89	80:10	15:87	85:12
201346:204716	82:11	18:88	85:13	15:86	82:11	18:89	15:87	85:12
365921:313752	19:89	80:10	16:88	83:12	80:10	20:90	16:87	84:12

**Fig. 2.** Weighted entropy for different data set sizes. The clustering was performed using the *k*-means algorithm, *H2* criterion function, *TP-IDF* document representation, and cosine similarity measure.

terms (vector dimensions) while the reviews were rather short. Thus, the *TF-IDF* representation did not differ much from the *TP-IDF* – the vector mostly contained only 0 and 1, and values 2 or more very rarely.

From the entropy point of view, Figure 3 displays the difference between the *TF* and *TF-IDF* word representation. Obviously, *TF-IDF* provided a significant improvement (the average entropy decrease), as the lower curve demonstrates.



**Fig. 3.** The influence of the word representation on the entropy decrease. The upper (dashed) curve illustrates the simple representation: using only the term frequency. The lower curve shows the significant improvement obtained by the representation known as term frequency  $\times$  inverted document frequency, *tf-idf*.

This improvement means that each one of the clusters (*positive* and *negative*) is ‘cleaner’, including less reviews that do not belong to a particular class.

Using cosine similarity provided the best results unlike the Euclidean distance and Pearson’s correlation coefficient. For example, for the set of documents containing 4,932 positive and 4,745 negative reviews, the entropy was 0.594 for cosine similarity, while Euclidean distance provided entropy 0.740, and Pearson’s coefficient 0.838. The  $H_2$  and  $I_2$  criterion functions provided the best results. For the  $I_1$  criterion function, the entropy of one cluster was very low (less than 0.2). On the other hand, the second cluster’s entropy was extremely high. Nevertheless,  $I_1$  together with cosine similarity function might have certain utilisation in practise.  $H_2$  criterion function provided more stable results in terms of the entropy value of each cluster. In other words, the difference between entropies of two clusters was smaller than for other criterion functions.

Stemming applied during the preprocessing phase had no impact on the entropy at all.

## 5 Conclusions

A number of experiments was performed on a large data set containing customer reviews written in English which was the most used natural language. The goal

was to automatically build clusters representing positive and negative opinions as well as finding a clustering algorithm with a set of its best parameters, similarity measure, clustering-criterion function, word representation, and the role of stemming. The main focus was on clustering very large real-world data during a reasonable time, without applying any sophisticated methods that can increase the computational complexity.

Experiments demonstrated that using cosine similarity measure produced results with the lowest entropy values, and was suitable for this type of data more than Pearson's correlation coefficient or the Euclidean distance measures. The Cluto implementation of the  $k$ -means algorithm performed better when compared with the repeated bisection and graph-based algorithms. In addition,  $k$ -means proved itself as a faster algorithm in comparison with others. From the entropy viewpoint, the best results were obtained with  $k$ -means using the binary representation weighted by the cosine similarity,  $idf$ , and  $H2$  criterion function, while stemming did not improve the results. As the authors assume (without any deeper investigation at the time of summarizing the results presented here), this negligible stemming role might be produced by relatively very different composition of individual reviews that were – on average – short (some 20 words), not sharing many words that would have the same stem. This phenomena should also be investigated for the given data type.

Surprisingly, the clustering revealed that in the original data some reviews were evidently wrongly categorized and the clustering was able to assign a correct class corresponding to the word contents also from the human point of view.

In this article, the authors demonstrated the clustering results only for English (almost two millions customer reviews). However, the research team collected approximately five millions reviews – concerning the same hotel service – in more than 25 languages. The continuing research is aimed at processing of the customers' reviews written also in other languages to compare the results for English to results for other languages, in order to show how the language-specific linguistic phenomena (for example, the inflexional morphology) might affect the results, and what may be common even for different language families.

**Acknowledgments.** This work was supported by the research grants of the Czech Ministry of Education VZ MSM No. 6215648904 and IGA of the Mendel University in Brno No. 47/2012.

## References

1. <http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download/> (July 4, 2012)
2. Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. *Machine Learning* 42(1-2), 143–175 (1999)
3. Figueiredo, F., Rocha, L., Couto, T., Salles, T., Goncalves, M.A., Meira, W.: Word co-occurrence features for text classification. *Information Systems* 36, 843–858 (2011)
4. Ganu, G., Kakodkar, Y., Marian, A.: Improving the quality of predictions using textual information in online user reviews. *Information Systems* (in press, 2012)

5. Ghosh, J., Strehl, A.: Similarity-Based Text Clustering: A Comparative Study. In: Grouping Multidimensional Data, pp. 73–97. Springer, Berlin (2006)
6. Huang, A.: Similarity Measures for Text Document Clustering. In: Proceedings of NZCSRSC, pp. 49–56 (2008)
7. Joachims, T.: Learning to classify text using support vector machines. Kluwer Academic Publishers, Norwell (2002)
8. Karypis, G.: Cluto: A Clustering Toolkit. Technical report, University of Minnesota (2003)
9. Korenius, T., Laurikkala, J., Järvelin, K., Juhola, M.: Stemming and lemmatization in the clustering of finnish text documents. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, pp. 625–633. ACM, New York (2004)
10. Li, C., Lin, N.: A Novel Text Clustering Algorithm. Energy Procedia 13, 3583–3588 (2011)
11. Li, N., Wu, D.D.: Using text mining and sentiment analysis for online forums hotspot detection and forecast. Decision Support Systems 48, 354–368 (2010)
12. Nie, J.Y.: Cross-Language Information Retrieval. Synthesis Lectures on Human Language Technologies 3(1), 1–125 (2010)
13. Porter, M.F.: An Algorithm for Suffix Stripping. Program 14(3), 130–137 (1980)
14. Škorpil, V., Šťastný, J.: Back-Propagation and K-Means Algorithms Comparison. In: 2006 8th International Conference on Signal Processing Proceedings, pp. 1871–1874. IEEE Press, Guilin (2006)
15. Wu, J., Chen, J., Xiong, H., Xie, M.: External validation measures for K-means clustering: A data distribution perspective. Expert Systems with Applications 36(3), 6050–6061 (2009)
16. Zhao, Y., Karypis, K.: Criterion Functions for Document Clustering: Experiments and Analysis. Technical report, University of Minnesota (2003)
17. Žižka, J., Dařena, F.: Mining Significant Words from Customer Opinions Written in Different Natural Languages. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNCS, vol. 6836, pp. 211–218. Springer, Heidelberg (2011)
18. Žižka, J., Dařena, F.: Mining Textual Significant Expressions Reflecting Opinions in Natural Languages. In: Proceedings of the 11th International Conference on Intelligent Systems Design and Applications, ISDA 2011, Cordoba, Spain, pp. 136–141. IEEE Press (2011)

# A Workbench for Temporal Event Information Extraction from Patient Records

Svetla Boytcheva<sup>1,2</sup> and Galia Angelova<sup>1</sup>

<sup>1</sup> Institute of Information and Communication Technologies,  
Bulgarian Academy of Sciences (IICT-BAS), Sofia, Bulgaria

<sup>2</sup> American University in Bulgaria, Blagoevgrad, Bulgaria  
sboytcheva@aubg.bg, galia@lml.bas.bg

**Abstract.** This paper presents a research prototype for temporal event information extraction from hospital discharge letters in Bulgarian. An algorithm for extraction of primitive events automatically sets markers for patients' complaints, drug treatment and diagnoses with precision about 90%. Specific domain knowledge is further used to generate compound events and to identify some relations between event time sequences. Absolute and relative time information enables ordering the generated compound events using semi-intervals and fuzzy logic. Some negated events are analyzed as well to better structure the patient history.

**Keywords:** Information Extraction, Health Informatics, Temporal Events.

## 1 Introduction

Temporal information extraction (IE) from narratives in patient records is a relatively new task in biomedical natural language processing (NLP) [19]. These IE systems rely on time-oriented patient data and time-based medical knowledge. Temporal reasoning is their key feature [9, 5, 17]. Various temporal IE systems have been demonstrated: summarizing data from temporal clinical databases, reasoning on temporal clinical data for therapeutic assessments, and modeling uncertainty in clinical knowledge and data [1]. Recent activities are often focused on annotation tasks. Initiatives like "i2b2/VA Track on Challenges in NLP for Clinical Data" [14] were conceived in the context of TimeML and TimeBank. TimeML is an ISO standard for annotation of temporal information [12]; TimeBank is a hand-annotated corpus conforming to TimeML [13]. In this approach all events (medical and others) are tagged. A clinical annotation schema based on TimeML was used to annotate corpus of more than 5000 tokens [15]. The HPI TimeML Corpus contains TimeML-annotated "History of Present Illness" sections of 44 discharge summaries [11]. Additional annotation for tense (past, present, future) corresponding to the whole event time with respect to patient hospital entrance is associated to each event. It is quite difficult to compare and assess the resulting corpora because most of them are not publicly available.

Although there are available significant research results for temporal events extraction in English, they cannot be directly applied for patient records (PRs) in

Bulgarian due to the specific medical terminology and the lack of digital resources. A mixture of terminology in Bulgarian, Latin and transliterated to Cyrillic Latin terms occurs in Bulgarian medical PRs. Another specificity is that the anonymisation procedure removes admission and discharge dates from PRs, which causes difficulties in further linkage between events when the text processing is done outside the hospital system.

Our approach recognizes dates and prepositional phrases containing temporal expressions. Specific domain knowledge is used to combine them into inter-related compound events. Temporal representation and reasoning are based on fuzzy logic.

Section 2 presents the discourse structure of discharge letters and discusses time interval representations. Section 3 considers existing prototypes for section splitting and temporal IE. Section 4 presents the evaluation of the current workbench for automatic extraction of temporal markers. Section 5 sketches further work and the conclusion.

## 2 Materials

We process anonymised hospital Discharge Letters in Bulgarian. We identify temporal events and their inter-relations from the Anamnesis section of the Patient Record (PR) that narrates the patient's disease history with rich temporal references. Our IE experiments were performed on a training corpus of 1,300 and test corpus of 6,200 anonymised hospital PRs for patients with endocrine and metabolic diseases. We adopt the event classification [8] into primitive and composite events. Primitive events can be explicit and temporal (relative and absolute).

The PRs text is split automatically into episodes (1). We assume that the Patient history is represented as a sequence of adjacent episodes:

$$e_1, e_2, \dots, e_n \quad (1)$$

An episode is represented as a vector (2) of primitive events in the episode:

$$e_i = < PID_{i1}, PID_{i2}, \dots, PID_{ik} > \quad (2)$$

Different episodes can contain different numbers of primitive events.

Primitive events contain descriptions for diagnoses, complaints, drugs, procedures or complications related to them and they are associated with time markers:

$$PE(PID, PRID, TimeBegin, TimeEnd, Mode, Type, Data) \quad (3)$$

Here *PID* is an unique ID associated with the primitive event; *PRID* is patient record's ID; *TimeBegin* is the initial time when the event is initiated; *TimeEnd* is the event termination time; *Mode* can be *positive* (the event happened), *negative* (the event is negated) and *conditional* (the event is optional and there are additional conditions that identify whether the event can happen); *Type* specifies information described in this primitive event - about drug, complaints, diagnose, procedure, lab result, or status; *Data* contains structured information ( e.g. diagnose/drug codes).

We use previously developed extractors of ICD-10 (the International Classification of Diseases, v. 10<sup>1</sup>) and ATC (the Anatomical Therapeutic Chemical Classification System<sup>2</sup>) codes. They assign ICD-10 codes to disease names with 84.5% precision and ATC codes to drug names with f-measure 98.42% [6,7].

Most of the time episodes in patient discharge letters contain no explicit information about beginnings and ends of intervals, which is required by Allen's theory of temporal intervals [2]. Several Allen's relations require the equality of two or more interval boundaries. However we process data about chronic deceases and in most cases only the beginning is marked. In order to deal with imperfect and incomplete information in patient records we use Freksa's theory of semi-intervals [10]. According to this theory the semi-intervals provide the following advantages: (i) they are rather natural entities both from a cognitive and from a computational point of view; (ii) coarse knowledge can be processed directly; computational effort is saved; (iii) incomplete knowledge about events can be fully exploited; (iv) incomplete inferences made on the basis of complete knowledge can be used directly for further inference steps. Freska defines eleven semi-interval relationships and depending on the types of deformation of events and their relations, he defines different neighborhood structures: A-neighbor relation - when for two events three of the four semi-intervals are fixed and allow the fourth to be moved; B-neighbor relation – in case we leave the duration of events fixed and allow complete events to be moved in time; C-neighbor relation – when we leave the ‘temporal location’ of an event fixed and allow the duration of the events to vary.

An interesting point in the use of time and tenses in natural language was brought out by Anscombe's investigation into the meanings of *before* and *after* [4]. For instance, from “The infection was present after the fever ended” it does not follow that the fever ended before the infection was present. Thus, *before* and *after* are not strict converses. Note that, however, from “The infection started after the fever started,” we can indeed conclude that the fever started before the infection started. Therefore *before* and *after* are converses when they link instantaneous events.

Another problem occurs in reasoning because in First Order Logic objects exist timelessly, time being just another dimension; in tenser approaches, “now” is a point of time in a separate class. Thus Fuzzy Logic [18] is a more convenient approach for modeling temporal data and reasoning on it.

### 3 Methods

#### 3.1 System Architecture

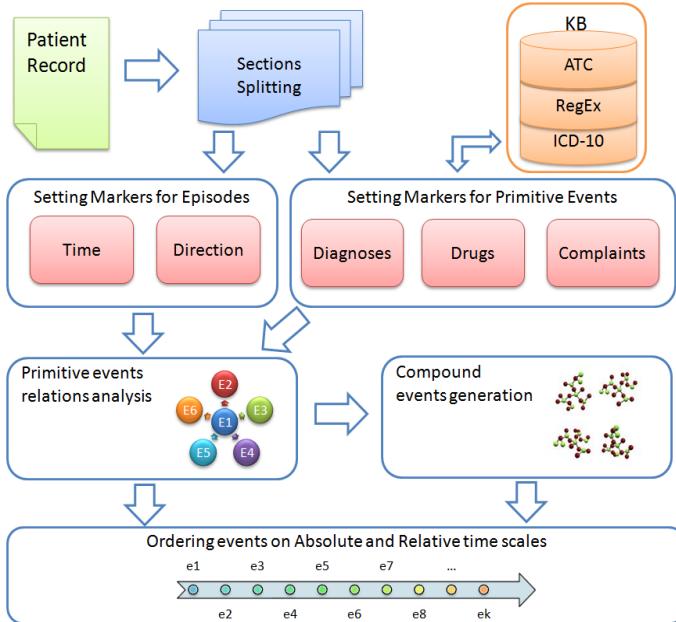
The temporal information extractor processes the input PR text as follows (Fig. 1):

- **Sections splitting** - based on regular expressions that recognize more than 70 keywords for sections names as well as missing or merged sections;
- **Setting markers for primitive events** (drug names, diagnoses, complaints) module - uses about 80 regular expressions. This module also uses especially developed extractors of ICD-10 and ATC codes for diagnoses and drug names [6,7];

---

<sup>1</sup> <http://www.nchi.govtment.bg/download.html>

<sup>2</sup> <http://www.who.int/classifications/atcddd/en/>

**Fig. 1.** System Architecture

- **Setting markers for episodes** (time and direction) module - uses about 50 rules defined as regular expressions based on keywords for directions, absolute and relative time (date and duration) [3,16]. Keyword lists include various types of absolute time representation like e.g. month description – by name, by Roman and Arabic numerals; dates and years for instance “since June 2009”, “in 2010” etc. There are also many keywords/phrases for relative time representation like e.g. “since then”, “after the puberty”, “after X was stopped” etc.
- **Primitive event relation analysis** - this module groups primitive event for drugs, complaints and diagnoses according to episodes, time and directions markers. The module sets TimeBegin and TimeEnd values for primitive events. In principle, an episode contains no/single/many time markers. In the first case we assume that this episode continues the explanations from the last time marker  $t_i$  mentioned in some preceding  $episode_i$  (4). Thus all events from  $e_{im}$  to  $e_{jn}$  in the current  $episode_j$  are related to the same time  $t_i$

$$\dots \underbrace{t_i \quad e_{im} \quad \dots \quad e_{ik}}_{episode_i} \dots \underbrace{e_{j1} \quad e_{j2} \quad \dots \quad e_{jn}}_{episode_j} \dots \quad (4)$$

In case there is a single time marker in the episode (5) – all primitive events from  $e_{i1}$  to  $e_{ik}$  are related to this time  $t_i$ .

$$\underbrace{e_{i1} \quad \dots \quad e_{im-1} \quad t_i \quad e_{im} \quad \dots \quad e_{ik}}_{episode_i} \quad (5)$$

In case of many time markers (6) – the primitive events in the episode are split according to them, and all primitive events between two time markers are related to the time marker preceding them

$$\underbrace{\dots \quad t_{i1} \quad e_{im} \quad \dots \quad e_{ik} \quad t_{i2} \quad e_{ik+1} \quad \dots \quad e_{in}}_{episode_i} \quad (6)$$

In this example the primitive events from  $e_{im}$  to  $e_{ik}$  are related to the time  $t_{i1}$  and primitive events from  $e_{ik+1}$  to  $e_{in}$  are related to the time  $t_{i2}$ .

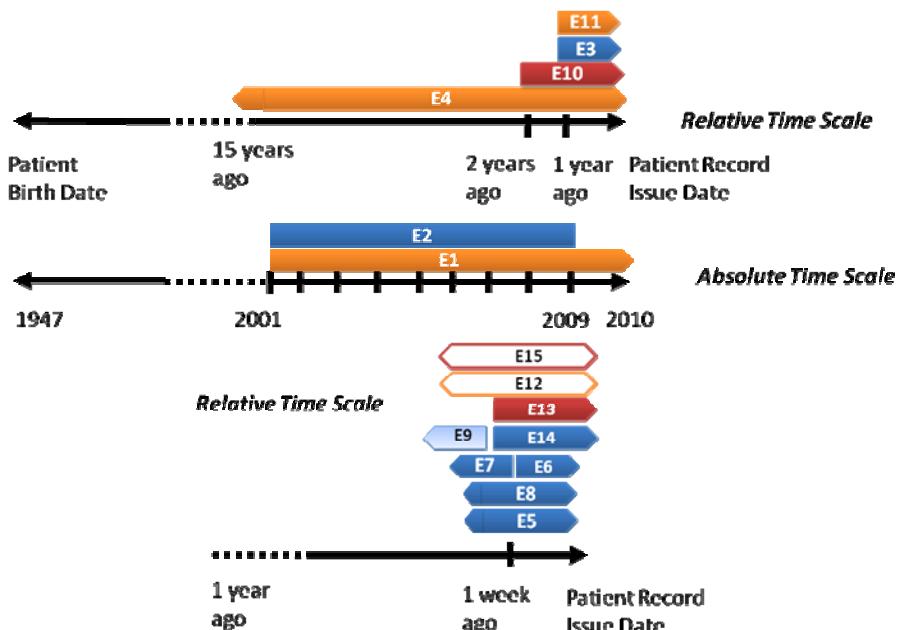
- **Compound event generation** - this module groups primitive events in compound events like e.g. treatment (all drugs prescribed for the period, all diagnoses). This allows further reasoning and finding cause-effect relations. Absence of complaints, symptoms, treatment with some drugs or diagnose is important information for further patient treatment.
- **Ordering events on Absolute and Relative time scales** - the algorithm for event ordering is based on directed multi-graphs representation. Where time markers are nodes (states), the edges represent primitive events, and they are incident with the beginning and end time nodes. Two graphs are generated – one for relative and one for absolute time scales. Some of the fuzzy relations are resolved.

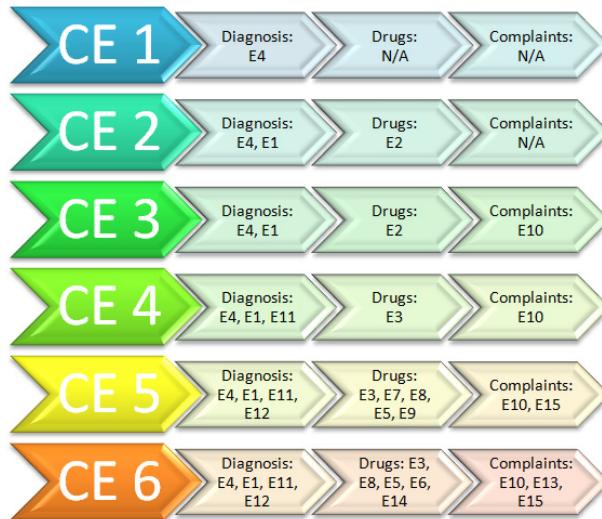
### 3.2 Example

The patient history for PR ID 03211 has been automatically broken into primitive events. Table 1 shows the extracted primitive events: 4 diagnoses (E1, E4, E11 and E12), 8 drugs (E2, E3, E5-E9, E14) and 3 complaints (E10, E13, E15). They are grouped according to the type of time markers – dates and durations. For only two of them (E1 & E2) are associated TimeBegin and TimeEnd values for absolute time scale (Fig. 2). Even in the narrative PR there is nothing mentioned about the value of TimeBegin for E2, therefore it is assigned to the same time as E1, because they are recognized as related primitive events grouped in compound event treatment. The absolute time scale starts at the patient's birth date and ends at the admission date. The relative time scale values later are resolved by mapping it to the absolute time scale and according to the admission date. The Compound Events generated from the primitive events pool by mapping time intervals to absolute and relative time scales and resolving their overlapping using Fuzzy logic are shown on Fig. 3. The resulting set contains 6 treatments (Diagnoses, Drugs, Complaints), but some of them are incomplete.

**Table 1.** Extracted events information from PR ID 03211

Event	ID	Time Begin	Time End	Mode	Type	Data
E1	03211	since 2001	N/A	pos	diagnose	E11, Non-insulin-dependent diabetes mellitus
E2	03211	N/A	until 2009	pos	drug	A10BB09, Diaprel MR, 2 tabl
E3	03211	about 1 year ago	N/A	pos	drug	A10BA02, Metformin NIHFI, 3x850mg
E4	03211	more than 15 years ago	N/A	pos	diagnose	I158, Other secondary hypertension
E5	03211	at present	N/A	pos	drug	C03BA11, Tertensif SR, 1 tabl
E6	03211	during the last week	N/A	pos	drug	C09AA02 , Renapril, 40mg
E7	03211	at present	N/A	pos	drug	C09AA02 , Renapril, 2x10 mg
E8	03211	at present	N/A	pos	drug	C07AB02, Betaloc, 50mg
E9	03211	at present	N/A	cond	drug	C02AC01, Chlophazolin, when needed
E10	03211	about 2 years ago	N/A	pos	complaints	serum creatinine, increased levels
E11	03211	about 1 year ago	N/A	pos	diagnose	O11X, Pre-exist hypertens disorder with superimposed proteinuria
E12	03211	at present	N/A	neg	diagnose	H360, Diabetic retinopathy
E13	03211	during last 10 days	N/A	pos	complaints	Hypertension
E14	03211	during last 10 days	N/A	pos	drug	C02AC01, Chlophazolin
E15	03211	at present	N/A	neg	complaints	decompensation of diabetes mellitus

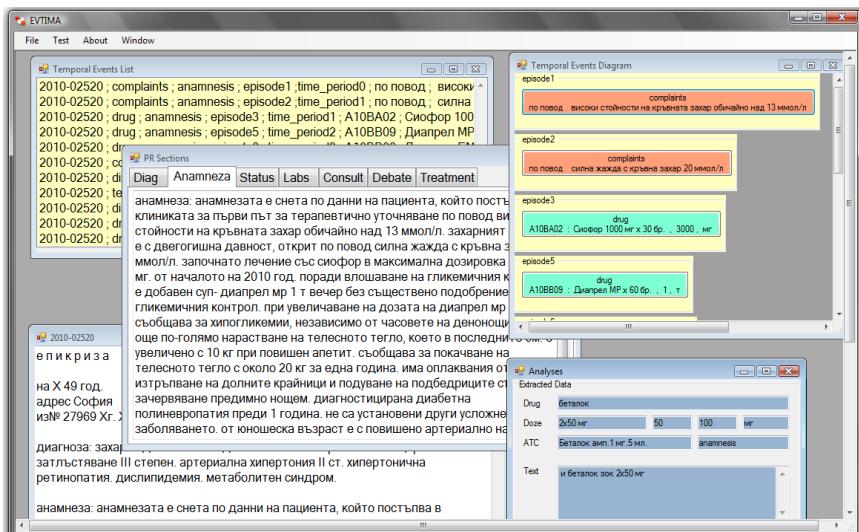
**Fig. 2.** Relative and Absolute time scales for primitive events from PR ID 03211



**Fig. 3.** Compound Events generated for PR ID 03211

### 3.3 Workbench Prototype

The workbench prototype is implemented in C#.Net as multidocument container. It provides functionalities for single and multiple PRs automatic analyses.



**Fig. 4.** Workbench for Temporal Events Information Extraction

For single PR processing the user can start manually each of the separate modules of the system and to monitor the analyses result in separate windows (see Fig. 4). One window contains the original PR text; a tabbed window displays the PR split into sections; another window shows step by step the extracted drug and diagnoses information. The list of extracted primitive events is presented in CSV format and the generated compound events and primitive events relations are presented in automatically generated graphical representation where different types of markers (for time, complaints, drugs, diagnoses) are shown by different colors and the compound events are shown in Group Boxes (see Fig. 4, the Temporal Events Diagram on the right side).

The multiple PRs processing mode generates CVS and XML files containing information about all PRs in a selected folder.

## 4 Evaluation and Results

In NLP the performance accuracy of text extraction procedures is usually measured by the *Precision P* (percentage of correctly extracted entities as a subset of all extracted entities), *Recall R* (percentage correctly extracted entities as a subset of all entities available in the corpus) and their harmonic mean f-score:  $F1 = 2.P.R / (P+R)$ .

The experiments were made with a training corpus containing 1,300 PRs and the evaluation results are obtained using a test corpus, containing 6,200 PRs. The evaluation results (Table 2) show high percentage of success in events and time information recognition in the PRs texts. These results are comparable with best systems accuracy like [5, 11] which reports results for relatively small corpora.

**Table 2.** Extraction sensitivity according to the IE performance measures

		Precision	Recall	F-Score
EVENT	<b>Drugs</b>	97.28%	99.59%	98.42%
	<b>Diagnoses</b>	97.30%	74.68%	84.50%
	<b>Complaints</b>	97.98%	96.82%	97.40%
TIME	<b>Dates</b>	98.86%	98.21%	98.53%
	<b>Duration</b>	99.14%	98.26%	98.70%
	<b>Frequency</b>	92.25%	95.51%	93.85%

For the Test set (6,200 PRs) were set 104,426 temporal markers in total and for the Training set (1,300 PRs) were set 24,924 temporal markers. The obtained results show that these markers are distributed approximately in the same percentage for both sets – about 38% of events present diagnosis descriptions, 47% are drug events and the remaining 15% are complaints. Table 3 presents summary statistics for the corpus. The average amount of primitive events per PRs is 20.69. The training sets contains information about 530 different drug names and 371 different diagnoses; in the test set data were found about 666 different drugs and 565 different diagnoses. The results are not surprising, because we are processing data for patients from a specialized hospital for treatment of endocrine disorders and most of them have similar diagnoses and treatment.

**Table 3.** Summary statistics for the corpus

	Sentences	Tokens	EVENTS	TIME
<b>Training</b>	27,155	250,773	14,137	10,796
<b>Test</b>	108,424	1,010,996	58,472	45,954
<b>Total</b>	<b>135,579</b>	<b>1,261,769</b>	<b>72,609</b>	<b>56,750</b>

In the test set the IE prototype has identified 1,349 temporal markers for complete date information (day/month/year), 2,698 absolute time markers with incomplete date information (year and/or month only), 2,362 markers for relative time periods and only 2,351 concerning the admission date.

Most incorrect event recognitions are due to: misspelling errors, unrecognized drug events for allergies, incorrect detection of negation scope, drug events occurrence in other context, unrecognized abbreviations, incorrect transliteration of Latin terminology and descriptions of specific pathological states which are hard to classify according to ICD-10 even for humans.

## 5 Conclusion and Further Work

The paper presents software modules which support the automatic extraction of temporal events information from PR texts.

The IE modules are strictly oriented to Bulgarian language and the structure of Bulgarian discharge letters. The plans for their further development and application are connected primarily to Bulgarian local context. Future enhancements are planned for the extension of the drug name and dosage recognition rules, to cope with certain specific exceptions. The preliminary correction of spell errors and other kinds of typos will also increase the IE accuracy. Regarding the diagnoses recognition task we plan improvement of the rules for more precise code assignments. Further work includes improvement of compound events generation and events ordering algorithms.

**Acknowledgements.** The research work presented in this paper is supported by grant DO 02-292 "Effective search of conceptual information with applications in medical informatics", funded by the Bulgarian National Science Fund in 2009-2012.

## References

1. Adlassnig, K.-P., Combi, C., Das, A.K., Keravnou, E.T., Pozzi, G.: Temporal Representation and Reasoning in Medicine: Research Directions and Challenges. *AI in Medicine* 38(2), 101–113 (2006)
2. Allen, J.F.: Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11), 832–843 (1983)
3. Angelova, G., Boytcheva, S.: Towards Temporal Segmentation of Patient History in Discharge Letters. In: Proc. of Biomedical NLP Workshop, in Conjunction with the Int. Conference, RANLP 2011, Hissar, Bulgaria, pp. 49–54 (2011)
4. Anscombe, G.E.M.: Before and after. *The Philosophical Review* 73, 3–24 (1964)
5. Boland, M.R., Tu, S.W., Carini, S., Sim, I., Weng, C.: EliXR-TIME: A Temporal Knowledge Representation for Clinical Research Eligibility Criteria. In: Proc. of AMIA 2012 Clinical Research Informatics Summit, San Francisco, CA, pp. 71–80 (2012)
6. Boytcheva, S.: Shallow Medication Extraction from Hospital Patient Records. In: Koutkias, V., Niès, J., Jensen, S., Maglaveras, N., Beuscart, R. (eds.) *Studies in Health Technology and Informatics Series*, vol. 166, pp. 119–128. IOS Press (2011)
7. Boytcheva, S.: Automatic Matching of ICD-10 codes to Diagnoses in Discharge Letters. In: Proc. of Biomedical NLP Workshop, in Conjunction with the Int. Conference, RANLP 2011, Hissar, Bulgaria, September 15, pp. 19–26 (2011)
8. Chakravarthy, S., Lee, H., Kim, H.: Support for Temporal Events in Sentinel. Design, Implementation and Preprocessing, RL-TR-97-214 Final Technical Report, Air Force Research Laboratory, Rome Research Site, New York (February 1998)
9. Combi, C., Keravnou-Papailiou, E., Shahar, Y.: Temporal Information Systems in Medicine, 1st edn., 397 pages. Springer (2010)
10. Freksa, C.: Temporal reasoning based on semi-intervals. *J. of AI* 54(1), 199–227 (1992)
11. Galescu, L., Blaylock, N.: A Corpus of Clinical Narratives Annotated with Temporal Information. In: Proc. 2nd ACM SIGHIT Int. Health Inf. Symp. (IHI 2012), Miami (January 2012)
12. Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust Specification of Event and Temporal Expressions in Text. In: Proc. of the Fifth Int. Workshop on Computational Semantics (IWCS-5), Tilburg (2003)
13. Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., Lazo, M.: The TIMEBANK Corpus. In: Proc. of Corpus Linguistics 2003, Lancaster, pp. 647–656 (2003)
14. Rink, B., Harabagiu, S., Roberts, K.: Automatic extraction of relations between medical concepts in clinical texts. *J. of AMIA* 18, 594–600 (2011)
15. Savova, G., Bethard, S., Styler, W., Martin, J., Palmer, M., Masanz, J., Ward, W.: Towards Temporal Relation Discovery from the Clinical Narrative. In: Proc. AMIA Annual Symposium 2009, pp. 568–572 (2009)
16. Tcharaktchiev, D., Boytcheva, S., Angelova, G.: Empirical Approach to Event Sequencing in Automatic Analysis of Patient Records. Poster Presentation at the 23rd Int. Conf., MIE 2011, Norway (August 2011)

17. Weng, C., Wu, X., Luo, Z., Boland, M., Theodoratos, D., Johnson, S.B.: EliXR: An Approach to Eligibility Criteria Extraction and Representation. *JAMIA* (suppl. 1), i116–i124 (2011)
18. Zadeh, L.A.: A theory of approximate reasoning. In: Hayes, J.E., Michie, D., Mikulich, L.I. (eds.) *Machine Intelligence*, pp. 149–194. Elsevier, Amsterdam (1979)
19. Zhou, L., Hripcsak, G.: Temporal reasoning with medical data - a review with emphasis on medical natural language processing. *J. Biom. Informatics* 40(2), 183–202 (2007)

# Analyzing Emotional States Induced by News Articles with Latent Semantic Analysis

Diana Lupan<sup>1</sup>, Mihai Dascălu<sup>1,2</sup>, Stefan Trăusan-Matu<sup>1</sup>, and Philippe Dessus<sup>2</sup>

<sup>1</sup> Politehnica University of Bucharest, Computer Science Department, Romania

<sup>2</sup> LSE, UPMF Grenoble-2 & IUFM–UJF Grenoble-1, France

diana.lupan@cti.pub.ro,

{mihai.dascalu,stefan.trausan}@cs.pub.ro,

philippe.dessus@upmf-grenoble.fr

**Abstract.** Emotions are reflected both in verbal and written communication. If in the first case they can be easier to trace due to some specific features (body language, voice tone or inflections), in the second it can be quite tricky to grasp the underlying emotions carried by a written text. Therefore we propose a novel automatic method for analyzing emotions induced by texts, more specifically a reader's most likely emotional state after reading a news article. In other words, our goal is to determine how reading a piece of news affects a person's emotional state and to adjust these values based on his/her current state. From a more technical perspective, our system (*Emo2 – Emotions Monitor*) combines a context independent approach (actual evaluation of the news employing specific natural language processing techniques and Latent Semantic Analysis) with the influences of user's present emotional state estimated through his/her specific feedback for building a more accurate image of a person's emotional state.

**Keywords:** emotional state, Latent Semantic Analysis, automatic evaluation of news articles.

## 1 Introduction

Emotions are one of the defining elements of human nature, as they are present in daily life and almost in any context. They add value to our interactions in a unique way and their induced effect can change the meaning of an entire message.

Although emotions are observably present in face-to-face or verbal communication, they can also be traced in written messages. If in the first case emotions are mostly expressed using body language and different voice features (for example, tone, rhythm, frequency), in the second one all these features are eliminated, as the communication channel cannot support them. Therefore, it can be difficult to detect emotions in written communication due to the lack of characteristics and our sole elements of analysis are words, grouped into sentences. Despite this fact, analyzing emotions in written texts can provide a better understanding of the underlying message by predicting its most likely impact on the person reading it.

Our approach covers the automatic evaluation of a person's mood and sentiment change when he/she reads an article. Our focus is to analyze short news consisting of

a headline and a brief description that are written with the intention of “provoking” emotions and attracting the reader’s attention.

This paper covers two approaches implemented in the Emo2 ([Emotions Monitor](#)) system: *context independent* and *environment-influenced* evaluation of emotional states. The context independent approach implies an evaluation of the article’s content with different natural language processing techniques from the perspective of the intended emotional state induced by the author of the article, while the environment-influenced approach takes into account the user’s current emotional state for adjusting the results, based on the influence of other similar articles to one’s current emotional state.

The remainder of the paper first presents similar related works and a general overview of the architecture describing each module. The fourth section focuses on the context independent evaluation process including details about the word valence and article similarity approaches. The fifth section presents the environment-influenced evaluation method, while the last sections comprise the evaluations, results and conclusions of our survey.

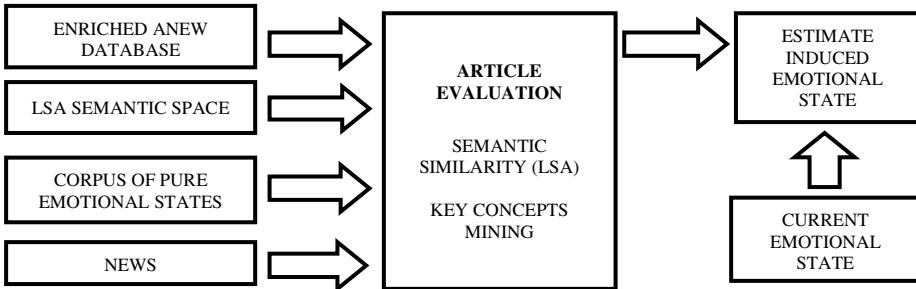
## 2 Related Work

The UA-ZBSA system [8] aimed at classifying a set of news headlines into six types of emotions: “anger”, “disgust”, “fear”, “joy”, “sadness”, “surprise”. Our approach is similarly comprised of six types of emotions, but we modified them to best fit our dimensions of analysis. Therefore, the included categories include “in control of the situation”, “fear”, “joy”, “sadness”, “excitement” and “boredom” described in detail in the next sections). UA-ZBSA’s emotion classification uses a bag of word approach comprising of frequency and co-occurrence counts of concepts collected from the World Wide Web. It applies the idea that “words which tend to co-occur across many documents with a given emotion are highly probable to express that emotion”. Moreover, it considers that adjectives with the same polarity tend to appear together.

On the other hand, UPART7’s design [9] is based on the assumption that all the words from an article’s title potentially carry emotions. The objective of the system was to identify the expression which carries the main topic of the title as this expression has major importance. Moreover, rules for detecting specific emotions were also developed (ex. a negation may be an indicator of a possible surprise). The system consists of the SS-Tagger (a Part-of-Speech tagger) and the Stanford Parser and used WordNet, SentiWordNet and WordNet-Affect as lexical resources. It also counts lexical elements that are later on classified as good indicators of surprise: negations, modal auxiliaries and question marks.

## 3 General Overview of the Functional Architecture

The designed system consists of several modules, grouped into processing layers that are presented in Fig. 1:



**Fig. 1.** System's Architecture

Firstly, regarding the overall processing algorithm, we must determine a person's initial or current emotional state. Therefore, our application tries to determine as accurately as possible the user's current emotional state by analyzing the feedback given by him on a set of 5 news articles, with regards to his/her induced current state. Although these initial news might alter a person's current emotional state, the results are later on used for adapting and personalizing future automatic results provided by the system through regressions in order to best reflect the actual impact of each new article on the user's current state.

From a higher level perspective, our analysis is divided into two separate steps: the initial, objective assessment, performed in the "Core evaluation" module, and the subjective, adaptive component of personalizing the results by considering the current emotional state of the user (integrated within the "Estimate Induced Emotional State" component).

The proposed evaluation method uses multiple inputs: the enriched ANEW database, a trained LSA semantic vector space, a corpus of articles expressing pure states and the news currently analyzed. The enriched ANEW database [1] consists of concepts (words) that have three values associated expressing its emotional state (Happy/ Unhappy, Calm/ Excited, Controlled/ In-Control). The Happy/ Unhappy dimension is quite straightforward as it expresses a state of happiness or sadness. Calm/Excited refers to the intention of the user to keep reading about the information presented in the current news article. The last dimension underlines the feeling of being in control of the situation just read versus being controlled by the events (or persons) described in the present news. The values for each dimension are in the [1, 9] range and the initial database of approximately 1,000 words was enriched using an algorithm that consists of 2 stages that are presented in detail in the following section.

On the other hand, the evaluation process can be divided in two approaches: **semantic similarity** evaluation between texts by means of LSA and **mining key concepts**. The later consists of extracting key concepts from both the body and the title of an article and combining their "emotional values" in order to determine the overall induced emotions. Therefore, a set of reference documents (initially the corpus consisted of 10 "pure" documents per emotional state) is used to obtain the semantic dimension of the analysis, while key-concepts mining offers the lexical dimension of our analysis. Key-concept mining is applied separately on the title and on the body of

news because we empirically considered that the title contains more valuable information than the rest. The title is usually designed to be short and to attract the reader's attention; moreover, we can consider that the title expresses from a human perspective the essence and the synthetic message of the body. Thus, due to their intrinsic value, the key-concepts belonging to the title have a greater contribution in the formula than the ones extracted from the body.

In the process of evaluating the emotional state by means of Latent Semantic Analysis, the result is determined by using the average value of all documents found similar with the content of the currently assessed article, weighted by the similarity between them. By using the similarity of concepts that induce a certain state, we can imply that documents with similar structure in terms of coverage of key concepts inflict a similar emotional state. The previous statement is not generally valid, but under the consideration of comparing news articles from a specific domain, within a short timeframe, our results from this approach proved to be quite accurate and to level the spikes of different dimensions of certain news articles within a predefined timeframe. In essence, news articles covering the same topic in a short timeframe rarely change their messages in terms of induced emotions.

The values obtained from the two perspectives are combined so the final result approximates, in the best way possible, the emotional state that is most likely to be induced when a person reads the article. The "Current emotional state" module integrates the user's input in our analysis (a questionnaire and 5 news articles) by estimating his present state and the fluctuations observed after reading each article. This set of articles contains texts extracted from the latest news feed into our application so their influence on the assessment of other similar articles can be considered relevant and significant. In our experiments, articles consisting of the title and the content (typically 2 or 3 phrases) were extracted from CNN.com's RSS (Really Simple Syndication) feeds.

## 4 Context-Independent Evaluation

The context-independent approach is based on two different stages: the first compares the concepts found in the analyzed article with terms stored in our enriched ANEW database (Word valence approach), while the second determines the similarities between the article and a set of predefined documents that express a dominant emotional state (Article similarity). In the end, the two approaches are combined during the article evaluation step for obtaining the corresponding valences on all 3 dimensions as a final score for each analyzed news article.

### 4.1 Word Valence Approach

Each term is associated with three connotations that express what kind of feelings it arouses when someone reads it. These valences expressed on the 3 dimensions (Happy/ Sad, Calm/ Excited, Controlled/ In-Control) are into a local database, formed through an enrichment process that is described in detail in the following paragraphs.

Initially, the number of stored terms was around 1,000 and the associated values were obtained from the ANEW database (Affective Norms for English Words, [1]) that was developed to provide a set of normative emotional ratings for a large number of words in English. This initial database was afterwards increased by using WordNet [3] synsets and Latent Semantic Analysis – LSA [4, 7] that enabled the determination of similarities between a “known” concept and a “new” one. By combining metrics of similarity from a lexicalized ontology with cosine similarity between concepts of the semantic vector space [5], our results were improved and we were able to better weigh the valence of words not present in the initial ANEW database.

LSA was trained on the Reuters corpus of news that provided adequate materials, relevant from the perspective of topics with regards to currently assessed articles and general enough to grasp underlying connections between concepts. The main reasons of using LSA are that its behavior has been proven to be close to that of human beings [2] and that it can be successfully used to enrich data obtained from experiments with human subjects and to approximate how reading a news article affects someone’s emotional state [6].

The actual enrichment algorithm consists of two stages. During the first stage we select a word and determine its synonyms using synsets from WordNet, whereas in the second stage similar concepts, from similar contexts, are determined through LSA. By considering the cosine similarity as the semantic metric for comparing two concepts, we can introduce new terms into our enriched database based on “known” concepts (already existent in the database), their corresponding values and the similarity with the term to be added. Moreover, only “known” concepts that are similar over a certain threshold are considered synonyms and are taken into account in computations. Therefore, based on the synonyms identified through WordNet of the concepts already present in the ANEW database and on the semantic similarity with  $k$  most nearest concepts, we determine a new word’s valence and introduce it in our enriched database. After running multiple iterations with increasing  $k$  values, in our final experiments we chose  $k$  to be the *three* most dominant and similar concepts by means of cosine similarity to the currently assessed one.

## 4.2 Article Similarity

As previously mentioned, this approach consists of determining similarities between the content of a news article with a corpus of documents that express a pure emotional state. Although the concept of “pureness” is subject to personal evaluation, these selected documents express a dominant dimension, identified and agreed upon by several human evaluators. The initial corpus used within our experiments comprises of 10 general interest news articles per state and LSA was used to compute the similarity between the currently evaluated article and each of the documents within the corpus.

## 4.3 Article Evaluation

The final result for each article is obtained by combining the values of each of the two previous approaches. Firstly, the concepts from the title are considered to be more valuable, in terms of expressed emotions, than the ones from the body.

The evaluation formula from the word valence approach is the following:

$$value_{key-concepts} = p * TitleValue + (1 - p) * ContentValue \quad (1)$$

where *ContentValue* and *TitleValue* are a linear combination of valences of key-concepts contained within the title and the body of the article.

The value of  $p$  was determined experimentally using multiple iterations within the range [0.55, 0.9], with increments of 0.05. The experiments revealed that 0.6 is the value that best combines the message expressed by the title with the one implied by the content and, in this case, the best overall correlation was obtained between the computer-based results and the human evaluators. On the other hand, the content of an article is considered to be similar to a text from the set if the similarity exceeds a threshold of 0.2 (the similarity can take values in the [-1, 1] range). For each emotional state, the arithmetic mean between the values that meet this requirement is computed. Next, the results are linearly normalized in order to be projected on a [1, 9] value range. The final result is computed using the following weighted formula which is designed to give more credit to the value obtained from analyzing the key concepts than to the value determined through article similarity, intrinsically limited by the bag of words approach:

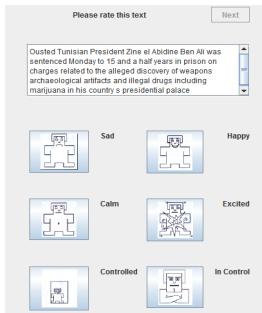
$$finalResult = p * value_{key-concepts} + (1 - p) * value_{setOfDocuments} \quad (2)$$

The best fit for  $p$  was experimentally determined by testing decreasing values within the range [0.55, 0.9] and the experiments revealed that  $p = 0.65$  leads to the best combination of the values associated with the word valence approach and with the semantic similarity between news and pure emotional state documents.

## 5 Environment-Influenced Evaluation

The environment-influenced approach comes as an adjustment to the results obtained in the first stage (context-independent evaluation) and can be seen as a feedback loop used for further personalizing the results provided by our system. Our assumption is that the emotional state of a person is differently influenced by a news article depending on his/hers current state and a set of preconceptions (mentality). For example, a person who is initially happy and reads sad news will be less affected and therefore less sad than a person who was initially unhappy. Moreover, the effects of reading the news also depend on specific features of the person reading the article, such as psychological profile, how geographically close he/she is to the events presented in the news.

Taking into account these observations, some additional features which rate these aspects were introduced. Firstly, the current emotional state is assessed by asking the reader to rate 5 news articles which are selected in order to be as similar as possible to the news that will be afterwards automatically analyzed and displayed (Fig. 2); later on, the user must answer a question referring to his current geographical location. The five presented articles are chosen by sorting all the news in the database after their similarities with the main news and picking up the top five most similar ones. Although these initial articles might change the user's current emotional state, their automatic assessment compared to the user's feedback provides valuable information on the current emotional state and offers the basis for applying regressions in order to fine-tune the results for the newly assessed articles.

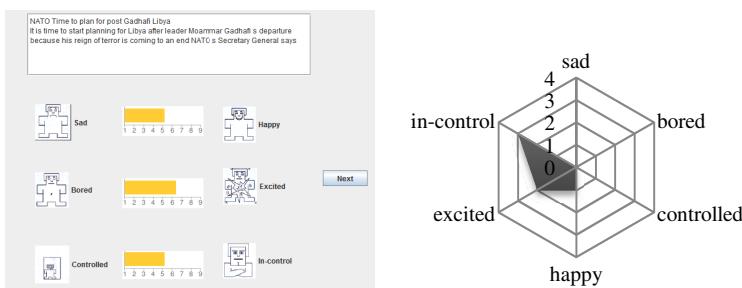


**Fig. 2.** Graphical interface for the initial rating of articles

The current geographical location of the user is used to adjust the results, following the assumption that events that happen nearby a person tend to affect him/her in a greater proportion. Concepts from the news article are used to determine whether the depicted events are located on the same continent introduced by the user. If the case, valences are augmented relatively to the mean value of each dimension. In order to achieve this functionality, GeoNames (<http://www.geonames.org/>) web services were used for determining the location of events.

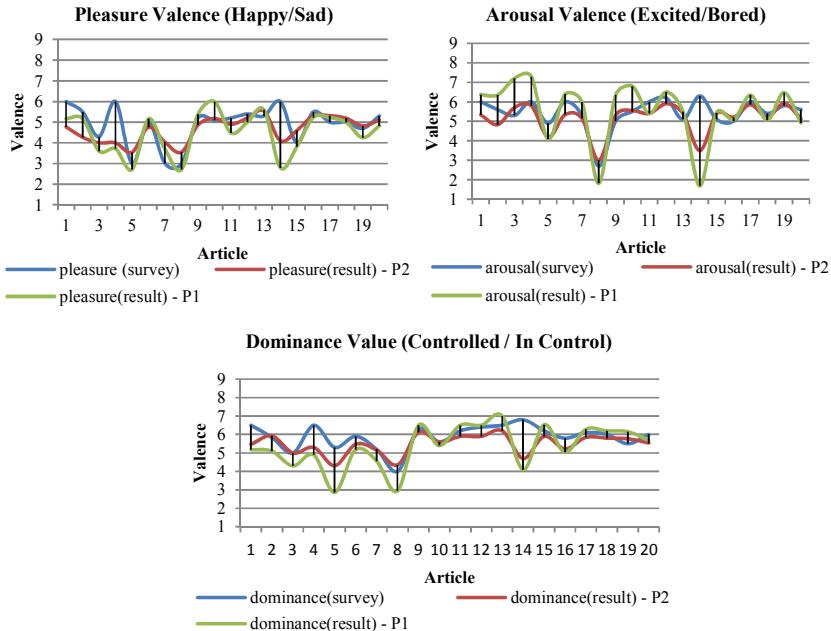
## 6 Results

As mentioned in the previous sections, the designed system was tested on RSS feeds from CNN.com. To test the application, the user is prompted with a window which displays an article and three charts that represent the most likely emotional state induced. Each chart is positioned between two images that suggest a pure emotional state so the user can have an accurate representation of the result. Fig. 3 presents a sample of the application's graphical user interface, including a generated radar graph depicting the 3 dimensions of the analysis:



**Fig. 3.** Graphical interface providing feedback on a specific article and an example of a generated graph

Our validation survey included 10 participants (5 women and 5 men) of age between 20 and 25 with diverse backgrounds (computer science and medical). The survey consisted of 2 stages, each having: 5 initial evaluation news (as previously mentioned, these articles are selected to be as similar as possible to the news to be automatically evaluated by the application) in order to determine the current emotional state, a question referring to the user's geographical position and 10 additional news articles as assessment for obtaining feedback from users regarding the application's results.



**Fig. 4.** Pleasure, Arousal and Dominance valences. P1 denotes the context-independent evaluation; P2 denotes the environment-influenced approach.

Therefore, the tests consisted of 20 RSS feeds from all the categories mentioned above. Synthetic results, covering the three dimensions of the analysis (Pleasure Valence, Arousal Valence, Dominance Valence), are presented in Fig. 4.

The correlations between the approaches and the survey, for all dimensions, are presented in Table 1.

**Table 1.** Correlations between the automatic results and the survey data

	Pleasure Valence	Arousal Valence	Dominance Valence	Average on all 3 dimensions
Survey - P1	0.55	0.48	0.61	0.55
Survey - P2	0.54	0.58	0.56	0.56
P1 - P2	0.86	0.91	0.96	0.91

As it can be observed from the previous table, Pleasure gets equal correlation between the two approaches and the average correlations on all three dimensions are very similar. But, by also taking into consideration the nature of each dimension, we can extrapolate from our experiment that Arousal is a more environment-dependent value in the sense that arousal may vary more than the others depending on the previous mood states, whereas Dominance is more influenced by the actual context and the topics within the article. Additionally, we obtained very high correlations between the two approaches, which was expected due to the fact that P2 refines the initial results.

On the other hand, there are some situations where the second approach obtains significantly inaccurate results in comparison to the survey (that expresses the actual emotions of the user) and also to the other approach. Besides the subjectivity of the humans' evaluation of their own emotional state, a possible explanation for this situation could be the fact that the news from the questionnaire were insufficiently similar to the ones displayed in the results; so, instead of adjusting the values from the first stage to be closer to reality, the algorithm is misguided and modifies them incorrectly, making the final results even worse. A possible solution to this problem could be enriching the news database with more texts, in order for new articles to be evaluated in comparison to other older, but more similar ones.

Additionally, although location provides some valuable input to our analysis, further factors will be addressed in future versions of our system: topics and personal interests highly impacting one's emotional state, gender and similar factors that can be used to differentiate assessment for different categories of users.

## 7 Conclusions

Our goal was to develop an application that estimates how reading a piece of news affects a person's emotional state and our approach consisted of a context-independent and an environment-influenced analysis of emotions. As technical background, we used several natural language processing techniques, mostly centered on Latent Semantic Analysis. The results were validated through a survey and, through the provided feedback, we can consider that our method is promising and relevant, in an area where subjectivity is the central player. On the other hand, there are some aspects that can be improved; therefore we plan to perform a deeper analysis and understanding of the text and to model the user's emotional state evolution.

As future research, we envision conducting surveys with an increased number of RSS feeds and audience so the obtained feedback would be more accurate. Also, increasing our corpus of "pure" emotional texts and including more psychological factors in our adaptive model could lead to improved results.

**Acknowledgment.** This work was partially supported by project 264207, ERRIC-Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1.

## References

1. Bradley, M.M., Lang, P.J.: Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. Gainesville (FL): The Center for Research in Psychophysiology, University of Florida, Tech. Report (1999)
2. Foltz, P.W.: Latent semantic analysis for text-based research. *Behavior Research Methods, Instruments and Computers* 28(2), 197–202 (1996)
3. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
4. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to Latent Semantic Analysis. *Discourse Processes* 25(2/3), 259–284 (1998)
5. Manning, C., Schütze, H.: Foundations of statistical Natural Language Processing. MIT Press, Cambridge (1999)
6. Wang, L., Wan, Y.: Sentiment Classification of Documents Based on Latent Semantic Analysis. In: Lin, S., Huang, X. (eds.) CESM 2011, Part II. CCIS, vol. 176, pp. 356–361. Springer, Heidelberg (2011)
7. Wiemer-Hastings, P.: How latent is Latent Semantic Analysis? In: Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI 1999), vol. 2 (1999)
8. Kozareva, Z., Navarro, B., Vazquez, S., Montoyo, A.: UA-ZBSA: A Headline Emotion Classification through Web Information. In: Proceeding of the 4th International Workshop on Semantic Evaluations, SemEval 2007, pp. 334–337 (2007)
9. Chaumartin, F.-R.: UPAR7: A knowledge-based system for headline sentiment tagging. In: Proceeding of the 4th International Workshop on Semantic Evaluations, SemEval 2007, pp. 422–425 (2007)

# The Emergence of Cultural Hierarchical Social Networks in Complex Environments

Mostafa Z. Ali<sup>1</sup> and Robert G. Reynolds<sup>2</sup>

<sup>1</sup> Jordan University of Science & Technology  
Irbid, Jordan 22110

[mzali@just.edu.jo](mailto:mzali@just.edu.jo)

<sup>2</sup> Wayne State University  
Detroit, Michigan 48202

[Robert.reynolds@wayne.edu](mailto:Robert.reynolds@wayne.edu)

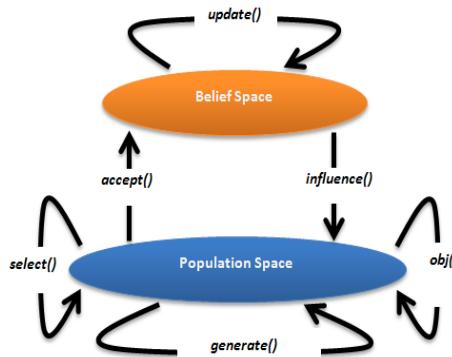
**Abstract.** In this research we present a configurable novel framework based on an enhanced heterogeneous hierarchical social fabric influence function embedded in Cultural Algorithms, as a powerful vehicle for the solution of complex problems. We motivate the discussion by investigating the extent to which these emergent phenomena are also visible within novel hybrid complex composition environments whose properties and complexity can be blurred and controlled easily, for the sake of overcoming any shortcomings of existing test functions that some of the current algorithms take advantage of during the optimization process. This environmental complexity induces an increase in the complexity of social roles within our system. We demonstrate how the well-configured hierarchical social fabric enhances Cultural Algorithms performance relative to other evolutionary algorithms from the literature.

**Keywords:** Cultural Algorithms, Knowledge source interaction, Social evolution, Social networks.

## 1 Introduction

Cultural Algorithms are one of several techniques that model the use of social intelligence to solve optimization problems. Other techniques include Ant Colony Optimization and Particle Swarm Optimization [1]–[3]. The Cultural Algorithm (CA) is a class of computational models derived from observing the cultural evolution process in nature [4], [5]. The CA has three major components: a belief space, a population space, and a communication protocol that describes how produced knowledge is exchanged between the belief and population spaces. The population space can support any population-based computational model, such as Genetic Algorithms, agent-based systems, and Evolutionary Programming. The basic framework is shown in Fig. 1.

A Cultural Algorithm is a dual inheritance system that characterizes evolution in human culture at both the macro-evolutionary level, which takes place within the belief space, and at the micro-evolutionary level, which occurs in the population space. Knowledge produced in the population space at the micro-evolutionary level



**Fig. 1.** The framework of cultural algorithm

is selectively accepted or passed to the belief space, and used to adjust the knowledge structures there. This knowledge can then be employed to influence the changes made by the population in the next generation.

Cultural Algorithms are a good fit here since in Cultural Algorithms the knowledge needed to solve a problem is decoupled from the individual agents. Thus, rather than locating large amounts of knowledge separately with each individual, it was stored in one place, the Belief Space, and used to selectively guide agents in each time step. Cultural Algorithm's performance has been studied using benchmark optimization problems [6] as well as applied successfully in a number of diverse application areas each with a large amount of domain knowledge [7] [8].

First, in order to simplify our model of the Belief Space five basic categories of knowledge sources were identified that were useful across a wide spectrum of problems, regardless of the application. From the standpoint of semiotics, it was found that there was evidence to support a genetic basis for each of the basic knowledge source in non-human species [9], [10]. Each of these knowledge sources was viewed to exist in the shared cognitive belief space of the culture in nature and was used to direct the decision-making process in the population space through their search for a solution for a given problem.

This research tries to set up a prototype of knowledge source interaction that is based on the Repast toolkit [11], in terms of the influence function that creates a desired procedural result on the benchmark landscape. Hence, the pattern etched on the landscape is the result of a non-procedural process of knowledge swarming.

## 2 Previous Work

Cultural Algorithms was successfully applied to many real word applications earlier, but we are interested in studying the fundamental computational processes involving the use of Cultural Systems as problem solvers on a new type of challenging problems with new specification that proves their worth trying-to-solve complexity.

Coello and Mezura-Montes [12] [13] presented an enhanced Evolutionary Algorithm that does not require the definition of extra parameters other than those

used by the evolutionary strategy. The Niched-Pareto Genetic Algorithm (NPGA) targeted multi-objective optimization problems where individuals are chosen based on Pareto dominance.

Mallipeddi and Suganthan [15] proposed an ensemble of mutation and crossover strategies and parameter values in which a pool of crossover and mutation techniques, together with a pool of values that correspond to each associated parameter, known as EPSDE, compete for the production of the offspring population. They tested the implemented algorithm's performance on a set of constrained benchmark problems designed for the conference on evolutionary computation (CEC) 2005.

Reynolds [16] implemented a version of Cultural Algorithms that uses the Marginal Value Theorem (MVT) for the predator/prey types of problems to where the individuals in the population get better solutions through the guidance of Knowledge evolution. This research expands that approach by using a well-configured Social Fabric Influence function with heterogeneous layered social networks to solve a set of complex problems that are used to test the power of any Evolutionary Algorithm.

### 3 The Social Fabric Influence Function

#### 3.1 Basic Concept

The Social Fabric is a social framework formed of agents and their interaction with their environment and the interaction among them. The purpose of the Social Fabric metaphor is to portray group behaviors and emergent phenomena [17].

Assume that  $\zeta$  represent the set of individuals. Assume that the social interactions among individuals  $\zeta$  are denoted by an undirected graph  $G(V;E)$ , where  $V$  is the set of vertices,  $V = \{v_1; v_2; \dots; v_I\}$  and  $E$  is a one-to-one mapping of the set of individuals  $\zeta$  onto itself, and  $I = |V|$  is the number of vertices (nodes),  $E$  is a collection of edges between all the vertices and  $q = |E|$  is the number of edges, (known as the *size* of the graph). Agent  $i$  is assumed to interacts with agent  $j$  if there is an edge in  $G(V;E)$  between nodes  $i$  and  $j$ : Let  $v(i)$  denote the local neighborhood of agent  $i : v(i) = \{j \in \zeta | j \neq i, \{i, j\} \in E\}$ : The number of  $i$ 's neighbors is the *degree* of node  $i : d_i = |v(i)|$ . Graph  $G(V;E)$  may be represented equivalently by its adjacency matrix,  $\mathbf{r}$ ; an  $I \times I$  matrix whose element  $(i; j)$  is equal to 1, if there exists an edge from agent  $i$  and to  $j$ ; and is equal to 0, otherwise.

Assuming that the set of all knowledge sources in the belief space is denoted as  $\theta$ , where:

$$\mathcal{E}_{ind(i)} = \begin{cases} Ks_{ind(i-1)} = Ks_{ind(i+1)}, Ks_{ind(i-1)}, Ks_{ind(i+1)} \in \theta \\ Ks_{ind(i-1)} \neq Ks_{ind(i+1)}, Ks_{ind(i-1)} \in \theta - \{Ks_{ind(i+1)}\} \end{cases} \quad (1)$$

$\epsilon_{ind(i)}$  is the possible neighborhood situation for individual  $i$ . Let us assume for illustration that the topology is a ring topology. In that case  $e_1$  and  $e_2$  are the two possible neighborhood situations.  $Ind(i+1)$  is the neighbor to the right of the individual in the ring topology, and  $Ind(i-1)$  is the neighbor to the left of the individual in the ring topology.

The Evaluation function ( $Ev$ ) is then:

$$Ev(e) = \begin{cases} Ks_{ind(i-1)}, & e=e_1 \\ Ks_{cr}, & e=e_2 \end{cases} \quad (2)$$

Where  $Ks_{cr}$  is the knowledge source that we get from applying the conflict resolution approach, where individuals might use the most frequently used one, the least frequently used knowledge, a random choice among the 5 types of knowledge sources, the direct knowledge source that affects individual  $i$  at any iteration.

The propagation of the original signal from the KS in the belief space depends on the applied neighborhood topology, and the tie breaking rules using the conflict resolution approach suggested earlier. The Social Fabric approach can be generalized to other topologies like the square topology. Using the square topology then  $Ev(s)$  can be represented as:

$$Ev(s) = \begin{cases} KS_j, \forall KS_j \in \{KS - KS_i\} \rightarrow weight(KS_i) > weight(KS_j) \\ KS_{cr}, \text{ otherwise} \end{cases} \quad (3)$$

$KS$  is the set of all knowledge sources {Normative, History, Topographic, Domain, Situational},  $KS_i, KS_j \in KS$ .  $Weight(KS_i)$  is the number of neighbors that follow knowledge source  $KS_i$ . The knowledge source with the maximum count will be used by the agent at a certain time step; otherwise the knowledge source produced by the tie breaking rules will be applied.

### 3.2 A Well-Configured Ranked Assorted Social Metaphor

#### *A Heterogeneous Social Fabric Metaphor*

The influence function in Cultural Algorithms is an important that affects how well the individuals can find promising regions. Some influence functions are more successful than others, as a result of the success of the agents that each has influenced in previous iterations. The original approach taken with Cultural Algorithms was to make the social fabric a homogeneous network [18] [19] [20].

The current algorithm supports several configurations and parameters that are crucial to a problem-solving process. The most important ones include social topologies taken from the Particle Swarm Optimization literature [21], the update window and the tie-breaking rules. We focus on the use of a heterogeneous topology that is composed from *lBest* topology (ring), *gBest* topology (global connectivity), *Square topology* where each individual has four connections to other individuals in the population, *Tree* topology (hierarchical network), *Hybrid* tree topology (groups of star-configured networks are connected to a linear information bus backbone).

### *Influence mechanism*

Knowledge sources have different individuals during the course of any run, and this depends on the way these individuals are successful in their search in the problem landscape to find the optimal. This continuously affects the type of informers the individual will follow at every iteration.

Having different ways of updating the individuals makes it probable to obtain groups of expert individuals that have different roles but complete the work of each other for a certain task. The individual might have a role as an “explorer”, “exploiter”, and “historian”. Exploiters usually get engrossed to better solutions found by exploiter knowledge sources.

### *Assorted Neighborhoods*

This type of mixing appears when agents have diverse neighborhood sizes and new connectivity. The population topology is not represented as a regular graph, where agents are shown as nodes with different links between them in the constructed network. Assortment of neighborhood sizes is one of the main factors that were tested to see how it affects the result of any new relationships to enhance the whole problem solving experience.

The accumulated effect of all the previous introduced heterogeneity determines the path of update. Aggregation is taken based on the result of all approaches and conflict resolution approaches are executed to determine the final influence on any individual at any generation.

## **4 Experiments and Analysis of Results**

The algorithm has been tested on a set of 25 high-dimensional challenging benchmark problems that can be found in [22]. These problems include different functions with varying complexity where 5 of them are unimodal, and the other 20 are multimodal problems that were each created as a blend several dimensions, complex enough to make them deceptive for any algorithm. While some comparisons are established in some research papers in the literature, but they are often confusing and limited to the test problems used in the study. In some cases, the test problem and the chosen algorithm are complementary to each other and the same algorithm under study may not work in other problems satisfactorily. It is vital to perform a scalability study signifying how the running time/evaluations upsurge with an increase in the problem size. These problems in [22] are challenging enough to prove how the algorithm might perform in a tricky and complex environment in terms of the problem size, constraints and the nature according to which it is been formed. Experiments are conducted on the 10-D problems. The population was set to 40.

Parameters used in our enhanced social fabric influence function in cultural algorithms (ESF\_CAs) are population size ( $N$ ), the control generation ( $G_c$ ), the number feasible elites ( $EliteN$ ) tie-breaking rule ( $R$ ), The size of the window after which the

Influence function will be applied ( $W_n$ ), The percentage of individuals in the population to be accepted into the belief space ( $P_a$ ) in order to update the knowledge sources. The range is between 0 and 1. The default value is chosen to be 0.25 and topology for the underlying network layer ( $T$ ). It is been proved for the sake of such complex constrained problems that the square topology along with most frequently used KS, and an update window of 10 iterations was suitable for complex constrained problems with different dimensions.

For each function, the algorithm is run 25 times. The best function error values are achieved when the function evaluations Fes=1e+3, Fes=1e+4, Fes=1e+5 for the 25 test functions are presented in Tables 1-3. On the other hand, Table 4 shows the number of function evaluations (min, 7<sup>th</sup>, median, 19<sup>th</sup>, maximum, mean and standard deviation) for a large number of dimensions (50-D) that are needed to reach the neighborhood of the function's global optimum with the accuracy or objective function error value as presented in the *Tol* column. We consider a run as successful if it reaches the *Tol* before  $n \times 10^4$  function evaluations. For functions 13 to 25 none of the runs reach the given accuracy.

Among the first unimodal problems, function 3 is a hard one as it has different condition numbers than functions 1 and 2 respectively. On the non-separable and ill-conditioned function 3, the performance is excellent whereas many other evolutionary algorithms fail to locate the optimum for this type of function.

The invariance against orthogonal transformations using this algorithm leads to a close performance on functions 9 and 10 which are the Rastrigin function and the rotated Rastrigin function, respectively. Function 8 is reported with a success rate of zero which can be explained by the changed scale of the function. A premature convergence for dimension 50 for problem 4, the noisy Schwefel, is due to the fast reduction in the step size. The results are much better than these presented in the best paper in the tournament of CEC 2005 [22] [23] [24].

## 5 Conclusion

In this paper we have investigated the use of a well-configured social function embedded in Cultural Algorithms framework, which is based on assorted aggregations over all the layers of the constructed network between individuals, in a way that positively guides that direction of individuals in the landscape. Our goal was to investigate the brunt that the addition of a well-configured and assorted layered social network would have on problem solving ability of a Cultural system for a set of challenging problems that we never tried before.

We showed that the new configured methodology of approaching how knowledge sources influence the individuals was successful in finding the optimal in most cases and enhancing the number of function evaluations needed to reach the neighborhood of the optimal under a certain predefined tolerance.

**Table 1.** Best objective function error value reached after  $10^3$ ,  $10^4$  and  $10^5$  function evaluations (FEs) for problems 1-8 in dimension n = 10. We stop a run as soon as the objective function error value drops below  $10^{-8}$  and its final value is used for all larger FEs.

FES	Prob.	1	2	3	4	5	6	7	8
<b>1e3</b>	min	<b>2.22e-03</b>	<b>2.96e+03</b>	<b>2.07e+06</b>	<b>1.21e+02</b>	<b>1.12e+01</b>	<b>2.33e+01</b>	<b>8.10e-01</b>	<b>3.11e+01</b>
	7 <sup>th</sup>	8.77e-03	4.61e+03	6.12e+06	7.36e+02	1.95e+01	2.31e+02	9.12e-01	2.65e+01
	med.	2.11e-02	6.42e+03	2.06e+07	3.77e+03	2.79e+01	3.01e+03	1.44e+00	2.72e+01
	19 <sup>th</sup>	3.41e-02	7.59e+03	3.15e+07	6.00e+03	4.45e+01	4.22e+03	1.51e+00	2.72e+01
	max	5.23e-02	9.46e+03	5.20e+07	3.24e+04	7.57e+01	1.02e+06	2.50e+00	2.81e+01
	mean	<b>2.17e-02</b>	<b>6.14e+03</b>	<b>2.24e+07</b>	<b>6.22e+03</b>	<b>3.74e+01</b>	<b>5.29e+04</b>	<b>9.94e-01</b>	<b>2.74e+01</b>
<b>1e4</b>	std	<b>2.01e-02</b>	<b>1.90e+03</b>	<b>2.76e+07</b>	<b>6.71e+03</b>	<b>1.99e+01</b>	<b>2.13e+05</b>	<b>2.37e-01</b>	<b>1.09e-01</b>
	min	1.22e-09	1.23e-10	1.87e-09	5.67e-10	7.13e-10	1.12e-11	4.28e-10	1.65e+01
	7 <sup>th</sup>	2.88e-09	2.09e-10	3.53e-09	7.14e-10	7.34e-10	2.67e-11	6.71e-10	1.65e+01
	med.	4.21e-09	3.39e-10	4.44e-09	8.25e-10	8.13e-10	5.26e-11	9.34e-10	1.65e+01
	19 <sup>th</sup>	5.43e-09	4.84e-10	5.23e-09	9.62e-10	9.68e-10	3.76e-11	1.04e-09	1.65e+01
	max	8.88e-09	6.99e-10	7.97e-09	9.79e-10	9.82e-10	6.58e-11	1.20e-09	1.65e+01
<b>1e5</b>	mean	<b>4.30e-09</b>	<b>3.99e-10</b>	<b>4.62e-09</b>	<b>8.60e-10</b>	<b>7.49e-10</b>	<b>5.35e-11</b>	<b>9.71e-10</b>	<b>1.65e+01</b>
	std	<b>1.27e-09</b>	<b>1.11e-10</b>	<b>1.46e-09</b>	<b>4.99e-10</b>	<b>6.30e-10</b>	<b>1.22e-11</b>	<b>2.65e-10</b>	<b>1.20e-06</b>
	min	0	0	1.87e-09	5.67e-10	7.13e-10	2.30e-12	2.11e-11	1.08e+01
	7 <sup>th</sup>	0	2.11e-15	3.53e-09	7.14e-10	7.34e-10	3.29e-12	1.52e-10	1.08e+01
	med.	0	2.11e-15	4.44e-09	8.25e-10	8.13e-10	6.77e-12	3.22e-10	1.08e+01
	19 <sup>th</sup>	0	1.89e-14	5.23e-09	9.62e-10	9.68e-10	4.24e-12	4.87e-10	1.08e+01
<b>1e5</b>	max	0	6.41e-14	7.97e-09	9.79e-10	9.82e-10	7.41e-12	9.12e-10	1.08e+01
	mean	<b>0</b>	<b>1.93e-14</b>	<b>4.62e-09</b>	<b>8.60e-10</b>	<b>7.49e-10</b>	<b>6.82e-12</b>	<b>3.66e-10</b>	<b>1.08e+01</b>
	std	<b>0</b>	<b>2.09e-14</b>	<b>1.46e-09</b>	<b>4.99e-10</b>	<b>6.30e-10</b>	<b>2.19e-12</b>	<b>2.55e-10</b>	<b>4.71e-04</b>

**Table 2.** Best objective function error value reached after  $10^3$ ,  $10^4$  and  $10^5$  function evaluations (FEs) respectively (rows) on benchmark problems 9-16 (columns) in dimension n = 10. We stop a run as soon as the objective function error value drops below  $10^{-8}$  and its final value is used for all larger FEs.

FES	Prob.	9	10	11	12	13	14	15	16
<b>1e3</b>	min	<b>7.61e+00</b>	<b>7.22e+00</b>	<b>3.12e+00</b>	<b>3.12e+02</b>	<b>4.31e+00</b>	<b>4.78e+00</b>	<b>6.23e+02</b>	<b>3.49e+02</b>
	7 <sup>th</sup>	4.20e+01	4.23e+01	8.08e+00	4.24e+02	5.29e+00	5.12e+00	6.49e+02	4.27e+02
	med.	3.11e+01	5.22e+01	1.11e+01	4.19e+02	5.68e+00	5.23e+00	6.92e+02	4.48e+02
	19 <sup>th</sup>	4.98e+01	6.33e+01	1.01e+01	4.34e+02	6.32e+00	5.31e+00	7.11e+02	5.15e+02
	max	6.13e+01	7.78e+01	1.42e+01	5.51e+02	6.81e+00	6.39e+00	7.87e+02	6.09e+02
	mean	<b>3.40e+01</b>	<b>5.41e+01</b>	<b>1.01e+01</b>	<b>4.35e+02</b>	<b>5.74e+00</b>	<b>5.18e+00</b>	<b>7.04e+02</b>	<b>4.74e+02</b>
<b>1e4</b>	std	<b>4.02e+01</b>	<b>1.10e+01</b>	<b>1.91e+01</b>	<b>3.81e+01</b>	<b>7.56e-01</b>	<b>4.96e-01</b>	<b>8.01e+01</b>	<b>9.11e+01</b>
	min	1.13e-02	1.22e+00	2.88e-02	2.14e-10	1.01e-01	1.00e+00	1.22e+01	5.20e+01
	7 <sup>th</sup>	2.89e-02	3.62e+00	1.72e-02	4.62e-10	3.22e-01	1.52e+00	9.14e+01	9.35e+01
	med.	3.52e-02	4.98e+00	3.54e-02	1.12e+01	4.45e+01	1.65e+00	9.35e+01	9.46e+01
	19 <sup>th</sup>	5.61e-02	5.39e+00	4.79e-02	1.03e+03	5.19e-01	1.77e+00	9.63e+01	9.62e+01
	max	7.08e-02	9.24e+00	5.89e-02	4.66e+04	6.88e-01	1.81e+00	9.95e+01	9.91e+01
<b>1e5</b>	mean	<b>4.01e-02</b>	<b>5.51e+00</b>	<b>3.72e-02</b>	<b>1.35e+03</b>	<b>5.11e-01</b>	<b>1.34e+00</b>	<b>9.46e+01</b>	<b>9.53e+02</b>
	std	<b>1.75e-02</b>	<b>1.25e+00</b>	<b>1.12e-02</b>	<b>1.21e+04</b>	<b>1.80e-01</b>	<b>1.22e-01</b>	<b>9.21e+00</b>	<b>5.26e+00</b>
	min	0	2.44e-11	2.88e-02	1.02e-10	2.37e-02	1.00e+00	0	5.03e+01
	7 <sup>th</sup>	0	4.32e-11	1.72e-02	3.44e-10	1.72e-01	1.52e+00	0	8.12e+01
	med.	0	6.78e-11	3.54e-02	4.44e-10	1.63e-01	1.65e+00	0	8.31e+01
	19 <sup>th</sup>	0	9.89e-10	4.79e-02	5.27e-10	1.79e-01	1.77e+00	0	8.49e+01
<b>1e5</b>	max	0	2.22e-02	5.89e-02	6.82e+00	1.99e-01	1.81e+00	1.13e+01	8.72e+01
	mean	<b>0</b>	<b>8.30e-03</b>	<b>3.72e-02</b>	<b>3.13e-01</b>	<b>1.79e-01</b>	<b>1.34e+00</b>	<b>1.24e+00</b>	<b>8.43e+01</b>
	std	<b>0</b>	<b>2.67e-01</b>	<b>1.12e-02</b>	<b>4.89e+00</b>	<b>1.05e-01</b>	<b>1.22e-01</b>	<b>3.46e+00</b>	<b>2.06e+00</b>

**Table 3.** Best objective function error value reached after  $10^3$ ,  $10^4$  and  $10^5$  function evaluations (FES) respectively (rows) on benchmark problems 17-25 (columns) in dimension n = 10

FES	Prob.	17	18	19	20	21	22	23	24	25
	min	<b>1.65e+02</b>	<b>4.21e+02</b>	<b>2.22e+02</b>	<b>3.78e+02</b>	<b>4.48e+2</b>	<b>8.32e+02</b>	<b>7.23e+02</b>	<b>3.11e+02</b>	<b>2.23e+03</b>
	7 <sup>th</sup>	2.21e+02	8.33e+02	6.47e+02	8.31e+02	4.65e+02	8.79e+02	2.21e+03	3.23e+02	2.32e+03
	med.	2.45e+02	9.12e+02	7.57e+02	8.74e+02	7.24e+02	8.85e+02	3.09e+03	4.45e+02	2.46e+03
<b>1e3</b>	19 <sup>th</sup>	2.89e+02	1.57e+03	9.36e+02	1.02e+03	1.09e+03	8.98e+02	3.10e+03	5.29e+02	2.53e+03
	max	1.32e+03	1.63e+03	9.49e+02	1.31e+03	1.12e+02	9.27e+02	3.23e+03	6.89e+02	2.66e+03
	mean	<b>2.60e+02</b>	<b>9.28e+02</b>	<b>6.97e+02</b>	<b>8.86e+02</b>	<b>7.32e+02</b>	<b>8.91e+02</b>	<b>3.13e+04</b>	<b>4.86e+02</b>	<b>2.41e+03</b>
	std	<b>1.85e+02</b>	<b>2.21e+02</b>	<b>1.63e+02</b>	<b>1.75e+02</b>	<b>3.53e+02</b>	<b>5.12e+02</b>	<b>2.81e+02</b>	<b>1.22e+02</b>	<b>3.78e+01</b>
	min	8.23e+01	2.24e+02	2.02e+02	2.47e+02	4.09e+02	5.23e+02	3.11e+02	1.09e+02	1.97e+02
	7 <sup>th</sup>	8.43e+01	2.34e+02	2.12e+02	2.62e+02	4.13e+02	5.41e+02	3.11e+02	1.25e+02	2.09e+02
	med.	8.53e+01	3.76e+02	2.26e+02	2.73e+02	4.26e+02	5.78e+02	3.11e+02	1.31e+02	2.31e+02
<b>1e4</b>	19 <sup>th</sup>	8.87e+01	5.17e+02	2.33e+02	2.88e+02	4.38e+02	6.58e+02	3.11e+02	1.52e+02	2.47e+02
	max	3.55e+02	6.66e+02	2.68e+02	2.97e+02	4.52e+02	7.33e+02	3.11e+02	1.90e+02	2.71e+02
	mean	<b>8.71e+01</b>	<b>3.82e+02</b>	<b>2.32e+02</b>	<b>2.79e+02</b>	<b>4.32e+02</b>	<b>5.82e+02</b>	<b>3.11e+02</b>	<b>1.39e+02</b>	<b>2.40e+02</b>
	std	<b>9.70e+01</b>	<b>1.95e+02</b>	<b>5.20e+01</b>	<b>1.98e+02</b>	<b>2.22e+01</b>	<b>2.99e+01</b>	<b>4.11e-13</b>	<b>3.32e+01</b>	<b>5.38e+01</b>
	min	8.13e+01	2.05e+02	2.02e+02	2.45e+02	4.09e+02	5.23e+02	3.11e+02	1.09e+02	1.23e+02
	7 <sup>th</sup>	8.22e+01	2.11e+02	2.12e+02	2.48e+02	4.13e+02	5.41e+02	3.11e+02	1.25e+02	1.23e+02
	med.	8.34e+01	2.28e+02	2.26e+02	2.56e+02	4.26e+02	5.78e+02	3.11e+02	1.31e+02	2.02e+02
<b>1e5</b>	19 <sup>th</sup>	8.48e+01	3.36e+02	2.33e+02	2.67e+02	4.38e+02	6.58e+02	3.11e+02	1.52e+02	2.10e+02
	max	1.02e+02	3.72e+02	2.68e+02	2.82e+02	4.52e+02	7.33e+02	3.11e+02	1.90e+02	2.32e+02
	mean	<b>8.59e+01</b>	<b>2.35e+02</b>	<b>2.32e+02</b>	<b>2.61e+02</b>	<b>4.32e+02</b>	<b>5.82e+02</b>	<b>3.11e+02</b>	<b>1.39e+02</b>	<b>1.94e+02</b>
	std	<b>1.17e+01</b>	<b>1.12e+01</b>	<b>5.20e+01</b>	<b>2.07e+01</b>	<b>2.22e+01</b>	<b>2.99e+01</b>	<b>4.11e-13</b>	<b>3.32e+01</b>	<b>2.37e+01</b>

**Table 4.** For the 25 benchmark problems at dimension n = 50, number of function evaluations (min, 7th, median, 19th, maximum, mean and standard deviation) needed to reach the neighborhood of the global optimum with the accuracy as presented in the Tol. column

Prob.	Tol	min	7 <sup>th</sup>	Median	19 <sup>th</sup>	max	mean	std	p <sub>s</sub>
1	1e-06	5.24e+03	5.36e+03	5.46e+03	5.62e+03	5.71e+03	5.34e+03	1.20e+02	1.00
2	1e-06	2.64e+04	2.69e+04	2.73e+04	2.78e+04	2.86e+04	2.75e+04	5.93e+02	1.00
3	1e-06	1.00e+05	1.01e+05	1.02e+05	1.02e+05	1.03e+05	1.02e+05	6.23e+02	1.00
4	1e-06	-	-	-	-	-	-	-	0.00
5	1e-06	-	-	-	-	-	-	-	0.00
6	1e-02	1.03e+05	1.13e+05	1.21e+05	1.40e+05	2.33e+05	1.48e+05	5.95e+04	1.00
7	1e-02	6.66e+03	6.85e+03	6.91e+03	7.33e+03	8.87e+03	6.98e+03	4.74e+02	1.00
8	1e-02	-	-	-	-	-	-	-	0.00
9	1e-02	3.66e+04	3.92e+04	-	-	-	3.81e+05	1.20e+04	0.30
10	1e-02	3.67e+05	-	-	-	-	3.95e+05	1.00e+04	0.15
11	1e-02	-	-	-	-	-	-	-	0.00
12	1e-02	-	-	-	-	-	-	-	0.00

## References

- Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceeding of the IEEE International Conference on Neural Networks, Perth, Australia, pp. 12–13. IEEE Service Center (1995)
- Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics 26(1), 29–41 (1996)
- Reynolds, R.G.: On modeling the evolution of Hunter-Gatherer decision-making systems. Geographical Analysis 10(1), 31–46 (1978)

4. Reynolds, R.G.: An introduction to cultural algorithms. In: Proceedings of the 3rd Annual Conference on Evolutionary Programming, pp. 131–139. World Scientific Publishing (1994)
5. Holland, J.H.: *Emergence*, pp. 1–10. Addison-Wesley Press, Reading (1998)
6. Chung, C., Reynolds, R.G.: CAEP: An evolution-based tool for real-valued function optimization using cultural algorithms. *International Journal on Artificial Intelligence Tools* 7(3), 239–291 (1998)
7. Reynolds, R.G.: An adaptive computer model of plant collection and early agriculture in the eastern valley of Oaxaca. In: Flannery, K.V. (ed.) *Guila Naquitz: Archaic Foraging and Early Agriculture in Oaxaca, Mexico*, pp. 439–500. Academic Press (1986)
8. Kohler, T., Gummerman, G., Reynolds, R.G.: Virtual Archaeology. *Scientific American* 293(1), 76–84 (2005)
9. Wynne, C.D.: *Animal Cognition—the Mental Lives of Animals*. Palgrave Macmillan, Great Britain (2001)
10. Clayton, N.S., Griffiths, D.P., Dickinson, A.: Declarative and Episodic-Like Memory in Animals: Personal Musings of a Scrub Jay. In: Heyes, C., Huber, L. (eds.) *The Evolution of Cognition*. The MIT Press, Cambridge (2001)
11. North, M.J., Collier, N.T., Vos, J.R.: Experiences creating three implementations of the Repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation* 16(1), 1–25 (2006)
12. Coello, C., Mezura-Montes, E.: Handling Constraints in Genetic Algorithms Using Dominance-Based Tournaments. In: Parmee, I. (ed.) *Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002)*, University of Exeter, Devon, UK, vol. 5, pp. 273–284. Springer (April 2002)
13. Horn, J., Nafpliotis, N., Goldberg, D.: A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In: *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Piscataway, New Jersey, vol. 1, pp. 82–87. IEEE Service Center (June 1994)
14. Deb, K., Goldberg, D.: An Investigation of Niche and Species Formation in Genetic Function Optimization. In: Schaffer, J.D. (ed.) *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, California, pp. 42–50. George Mason University, Morgan Kaufmann Publishers (June 1989)
15. Mallipeddi, R., Suganthan, P.N.: Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Dash, S.S. (eds.) *SEMCCO 2010. LNCS*, vol. 6466, pp. 71–78. Springer, Heidelberg (2010)
16. Reynolds, R., Peng, B.: Cultural algorithms: computational modeling of how cultures learn to solve problems: an engineering example. *Cybernetics and Systems* 36(8), 753–771 (2005)
17. Brock, W., Durlauf, S.: Discrete Choice with Social Interactions. *Review of Economic Studies* (2001)
18. Ali, M.Z., Reynolds, R.G.: An Intelligent Social Fabric Influence Component in Cultural Algorithms for Knowledge Learning in Dynamic Environments. In: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, September 15–18, vol. 02 (2009)
19. Reynolds, R.G., Ali, M.Z.: The social fabric approach as an approach to knowledge integration in Cultural Algorithms. In: *IEEE Congress on Evolutionary Computation*, pp. 4200–4207 (2008)

20. Che, X., Ali, M.Z., Reynolds, R.G.: Robust evolution optimization at the edge of chaos: Commercialization of culture algorithms. In: IEEE Congress on Evolutionary Computation, pp. 1–8 (2010)
21. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceeding of the IEEE International Conference on Neural Networks, Perth, Australia, pp. 12–13. IEEE Service Center (1995)
22. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Nanyang Technological University (2005)
23. Auger, A., Hansen, N.: Performance Evaluation of an Advanced Local Search Evolutionary Algorithm. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 1777–1784 (2005)
24. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: Proc. IEEE Congr. Evol. Comput. (CEC 2005), Edinburgh, Scotland, pp. 1785–1791. IEEE Press (September 2005)

# Ensuring Relevant and Serendipitous Information Flow in Decentralized Online Social Network

Udeep Tandukar and Julita Vassileva

Department of Computer Science, University of Saskatchewan, Saskatchewan, Canada  
`udeep.tandukar@usask.ca, jiv@cs.usask.ca`

**Abstract.** This paper presents a novel peer-to-peer architecture for decentralized Online Social Network and a mechanism that allows each node to filter out irrelevant social data, while ensuring a level of serendipity, by letting important information pass, even if it does not fall in the areas of interest of the user. The evaluation of the approach, using an Erlang simulation with 2318 nodes shows that it works as it was designed to: with the increasing number of social data passing through the network, the nodes learn to filter out irrelevant data, while serendipitous important data is able to pass through the network. Future work will implement the mechanism in a decentralized OSN and evaluate it in a real community of users, in an enterprise context.

**Keywords:** Information propagation, Online Social Network, Decentralization, Peer-to-peer system, Relationship modeling.

## 1 Introduction

Users increasingly rely on their friends in Online Social Networks (OSNs), like Facebook, to get news and information relevant to their interests. Current centralized OSNs own all the information of their users; once it is shared, the user loses control over it and cannot remove it or export it into another OSN. This is why OSNs are being viewed as information silos [11]. Decentralized and open OSNs have been proposed by Tim Berners Lee and others [11] as a promising future alternative of current centralized, closed and corporate-owned OSN. Such networks allow users to retain control over their data, rather than surrendering it to a corporation.

A decentralized OSN can grow organically with the number of users who join, starting from a small cluster of friends, serving, for example, the employees of an organizational unit, a class of students, a group of volunteers, or a neighbourhood. Such small-scale OSNs are of particular interest to any organization that attempts to provide a forum and community to share experiences, expertise, but not surrender all the communication data to a commercial company such as LinkedIn, Google+ or Facebook. Decentralized OSNs scale well, just as peer-to-peer systems do, without the need of investing in advance in a powerful server and databases that handles all requests and stores all the data (which may not be necessary, if the community flops). Even better, a decentralized social network can survive at a small scale, and doesn't

need to compete with Facebook for world dominance. With a few committed “hub” users, a decentralized OSN that allows for interoperability would have sufficient material to feed a fairly large social network of users.

While there are already open source projects (e.g. Diaspora), aiming to develop a decentralized OSN alternative for Facebook, there is still relatively little research in this area. There are a lot of interesting issues to explore. Decentralized OSNs share some of the same problems with centralized OSNs, for example, dealing with the cognitive overload caused by too much information flowing (called social network overload), which demands finding ways to filter out irrelevant data or to empower the user to do that. Filtering information, however, gives rise to the personalization bubble problem [6], in both centralized and decentralized OSNs. The decentralization also poses some specific problems. For example, certain filtering approaches (e.g. collaborative filtering), are impossible in a decentralized OSN, due to the unavailability of centrally stored user ratings, or traces. On the other side, decentralization allows for other approaches, that are not possible or practical in centralized OSNs, e.g. developing detailed user models stored locally and shared on demand for a given purpose [4]. The decentralization leads also to some specific problems faced by OSNs, for example, the need to create a virtual “co-presence” of users, in order to maintain the connections among users and the flow of information. Another such problem is the need to delegate some of the maintenance of the social network tasks, which is done by the central server in centralized OSNs to the individual nodes (clients), for example, the user modeling of interests, the filtering of irrelevant data, the identification and protection from spammers, ensuring that important social updates (e.g. status updates, photos, links, videos, etc.) are propagated through the network, ensuring that the users are connected.

This paper presents a novel peer-to-peer architecture for decentralized OSN and a mechanism that allows each node to filter out irrelevant social data, while ensuring a level of serendipity, by letting important information pass, even if it does not fall in the areas of interest of the user.

## 2 Approach

This section presents the architecture of the decentralized OSN that we base our approach on, and the main idea of the proposed mechanism to filter irrelevant information and to ensure serendipity by receiving important / popular information.

### 2.1 Architecture

OSNs can have decentralized architecture either by having the nodes (clients) store their own social data (locally, or on the cloud, at a securely accessible location) or by using an existing P2P architecture. To accommodate familiar functionalities of OSN (for example, status updates, photo uploads, commenting, rating) in decentralized architecture, there are various challenges [3]. The propagation of social data among users in a P2P OSN should be managed so that there is minimal duplication and

latency. Using P2P architecture, data are stored at the peers and the availability of the social data depends on the online behaviour of peers. In contrast, if users store their social data on the web in some standard format such as RDF in a location which can be accessed through URI, they have more control over their data and it is more accessible (e.g. from various devices of the same user). We chose a decentralized architecture for building OSNs in which the client is storing the user data on a cloud accessible via secure connection. The user data include the social updates, the relationship model of the user, and an expandable list of interest areas. When a node forwards a social update to another node (of a friend user), only the URI of the social update is being forwarded, along with a key that the recipient can use to access it. Permissions can be shared and forwarded among the nodes, in the same way as posts are currently reposted by users for their friends to see on Facebook. Yet, unless the recipient makes the effort to copy and save locally the actual social update (Web URL, or text, or image), it remains always in control of the user who shared it in the network, and can be easily removed by him or her. While some copying of other users' social updates will happen, it will happen locally, occasionally, and in will be in no way comparable to the massive surrendering of social updates that users currently have to do on centralized commercial OSNs.

## 2.2 Relationship Modeling

In [8] an approach for reducing the flow of irrelevant information in decentralized OSNs was proposed, but it was not evaluated. Here we modify and extend the mechanism allowing serendipitous data to avoid the personalization bubble problem and present an evaluation of the mechanism.

An interpersonal relationship model is used as a filtering mechanism for irrelevant information. The relationship consists of a representation of relationship strengths between a user and her friends in different areas of interest. Thus, it depends on the strength of relationship between the two users, but this strength is contextualized according to a particular category of interest. The intuition behind this is that two people can be friends, but not share the same level of interest in different topics or categories and not trust each other's judgement with regard to these categories. Therefore, a user may be interested to receive updates from her friend about, say fashion, but not in politics or health. On the other side, the same user may be interested to hear about health topics from another friend, yet, she may not be interested in her updates about fashion. While it adds complexity of representation and computation, it is advantageous to add an interest dimension to a relationship, since it allows the flexibility to filter both based on the source of the update and the category of interest. The strength of a relationship from one user to another in a given category of interest is based on previous interactions related to this category of interest. In general, to determine the area of interest of the shared information, users have to either tag their updates with the interest areas or the system has to extract semantics from the shared social update. However, allowing a full semantic categorization would be probably unnecessary complex and won't add much in terms of filtering functionality. That is why it is better to limit it to a certain fixed number

(for example, 10 or so) of predefined general categories, similar to those used in Yahoo or other news sites, e.g. politics, news, technology, sports, health, fashion, living, art, games, humour, etc.

The relationship strength from one user (sender) to another (recipient) is updated using the feedback that the recipient has provided. The feedback is based on the actions of the recipient and influences differently the relationship strength. For example, if the social data is viewed and re-shared, this would increase significantly the relationship strength. Viewing and commenting or rating will also increase the strength. Just viewing and not doing anything else would slightly reduce the relationship strength in the category of the social update. The relationship strength is calculated using a simple formula for simulated annealing (reinforcement learning). The details about the update formula and feedback values are presented in [8].

The relationship models as defined in [8] were maintained by the sender of social update and used to filter outgoing information to her friends. This approach reduced irrelevant information from flowing through the network but it created a possibility for the model to be tampered by malicious senders to spam other users on the network with irrelevant information. Also, there is no particular incentive for the sender to filter out outgoing information considering that maintaining the relationships model involves some space and computation costs. Therefore, the approach was modified so that the relationship models reside on the recipient's side. Thus the recipient has interest in maintaining the model since it protects it from irrelevant information and from spammers. Instead of sending feedback to the sender so that the sender can update the relationship strength (as in the previous version of the mechanism described in [8], the recipient updates the relationship strength between the recipient and the sender for the area of interest of the shared social data. The updated relationship model is used later by the receiver to filter out incoming social data in this category.

Since the proposed approach is implemented in a decentralized architecture, the computation overhead of maintaining the relationship models is distributed among all the peers. Even in popular OSNs, such as Facebook, an average user has approximately 130 friends, so the calculation and manipulation of relationship models for this number of friends and a limited number of categories of interest is not computationally expensive.

In summary, an improved content-based filtering approach is proposed, building on [8], that uses a model of users' interest in certain topics that is overlayed over a model of social relationships of the user. This extends classic content-based filters by allowing filtering information based both on the origin and on the interest level.

### **2.3 Serendipity**

All information filtering mechanisms suffer from the “personalization bubble” problem [6]. The user becomes gradually isolated from information which could be important but not in her area of interest. Serendipitous information can be defined as information, which is unexpected, but desirable.

In the approach, as described above, like in other information filters, the user will not be able to see serendipitous information coming from friends with whom the relationship in the area of interest of the information has become weak. There should be a way to allow such serendipitous information to reach the user, especially, if it is important information, for example, rated highly by other users. To allow such information to pass through the filter imposed by the relationship model, an extra parameter is considered for the social updates, in addition to their category of interest – the popularity of the social update. The popularity parameter is a property of the social data and can be calculated as the count of the re-shares, or the number of ratings (e.g. Likes or +1s). When a social data with a high popularity parameter is encountered, it is propagated without being filtered by relationship models, i.e. the user will see it despite not being interested in the category of the update or in the source of the update.

Since the information belongs to a category that is not of interest for the user, viewing/sharing of this information should not affect the relationship strength between the receiver and sender. The effect of such popular irrelevant information is not considered while updating the relationship strength.

### 3 Evaluation

To evaluate the discussed approach of using relationship modeling to reduce irrelevant information in decentralized OSNs, a simulation of the system has been developed in Erlang<sup>1</sup>. The simulation consists of many nodes representing users connected with each other in a social graph, acting independently. Each node is connected with other nodes through a social relationship. Each node of the social graph is an Erlang process/thread capable of communication through message passing. The processes in Erlang do not share any memory and are completely independent. This means a process running on one system can communicate with another process running on different system. That is why Erlang-based systems and simulations can be scaled up to incorporate large social graphs.

Relationship strength between users for a particular interest category is the basic variable to determine which social data would be relevant or irrelevant to users. This strength is updated by the recipient node whenever the social data is shared by her friends on a particular category. The update of the strength depends on the interest level of the recipient on that category. The nodes have different interest levels in each category. The categories of interest are at the same level and their number can vary. Experiments have been conducted with the number of categories set at different values from 1 to 20; the results reported in this paper are from a setup with 10 categories. The interests of the nodes in the different categories are distributed following a power law over the nodes in the social network, with some categories being the most common, covering large proportion of population and some – niche categories, popular only among small proportion of the nodes. The mechanism to generate such skewed distributions is known and based on growth (people gain new interests with time) and preferential attachment (areas of interest that are already popular attract newcomers with a higher likelihood) [2].

---

<sup>1</sup> <http://www.erlang.org/>

Social data is seeded to some subset of the nodes, which propagate it to their friends. Each social data belongs to one category of interest (assumption), and is seeded with a higher likelihood to nodes which have higher interest in that category of interest.

### 3.1 Experiments

The social graph which is used to simulate the decentralized OSN is a subset from the StudiVZ<sup>2</sup> dataset. This is a publicly available dataset from a popular German social network for students with over 1 million user profiles. While it is certainly possible to use large sub graphs in the Erlang simulation, an enterprise or neighbourhood social network (the most likely adopters of decentralized OSN) would be smaller. That is why a subset graph consisting of 2318 nodes with 2875 edges was generated from the dataset. This subset graph has an average clustering coefficient of 0.1. This clustering coefficient is in the range of the typical coefficients for OSNs. For example, Facebook's clustering coefficient varies between 0.133 and 0.211 with the average over all 22 regional networks 0.167 [10]. The average node degree in the data subset is 20, which is fairly low. The analysis of Facebook data shows that nodes with lower degree have higher clustering coefficients. That means that the users who have fewer friends on average will be part of denser clusters, as their friends are more likely to be interconnected as well. So our data subset presents a harder problem for propagating social data as it is sparser than existing social networks.

Ten different interest categories are distributed over the nodes following a power law. To simulate social updates being generated or shared by users initially, 10,000 messages are seeded to 5% of the population of the network. The seeded nodes are selected in such a way that first priority is given to those nodes whose interest matches the category of the message and only when there is not sufficient number of interested nodes - to other random nodes. Each node forwards each seeded social update to its friends. A node that receives a social update does the following:

1. It checks the relationship value with the sender on the category of the message. If this value is too low, it does not display the message and does not forward it further.
2. If the relationship value is high enough, it checks its own interest level in this category and updates its relationship strength in this category to the sender. Different values of feedback for updating the model are used depending on the level of interest that the node has in the category. If the level of interest is high, then relationship will be strengthened significantly, and the social update is forwarded to its friends. If the level is in the medium range, the relationship will be slightly strengthened (simulating that the user has provided a rating or a comment). If the level of interest is low (i.e. the user has just viewed the update), the relationship strength slightly decreases. After a certain number of such received marginally interesting updates, the strength of the relationship in this category drops down below the threshold and future updates from this node are filtered away.

---

<sup>2</sup> <http://studivz.irgendwo.org/>

Initially, the relationship strengths for all nodes and categories are set to the maximum value, i.e. 1. This ensures the flow of information at the beginning of the experiment. After that, the system will gradually learn as the relationship model is updated by the recipients of the social updates. The learning rate of the relationship model (i.e. the reinforcement learning formula) is set high enough so that the network can learn fast. However, the relationships should not deteriorate too quickly, to preserve the flow of information. Rationale for choosing the learning rate value is presented in [8].

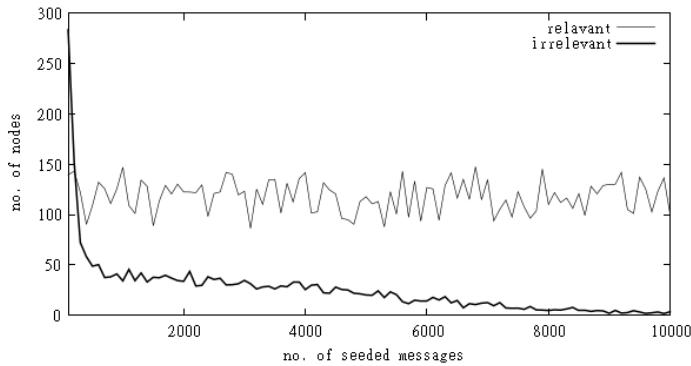
The social data in the network has a static parameter which determines its popularity. Even if the social data belongs to a category for which the relationship strength between the recipient node and the sender node has weakened below the threshold for display, the recipient will forward this social data. This ensures the serendipity through propagating popular social data, despite the different interests of the nodes in the OSN.

Two experiments were conducted. The first aimed to verify the effect of using relationship model in filtering irrelevant information. The second aimed to ensure that the popular messages are being displayed (serendipity) and forwarded by nodes despite their different interests.

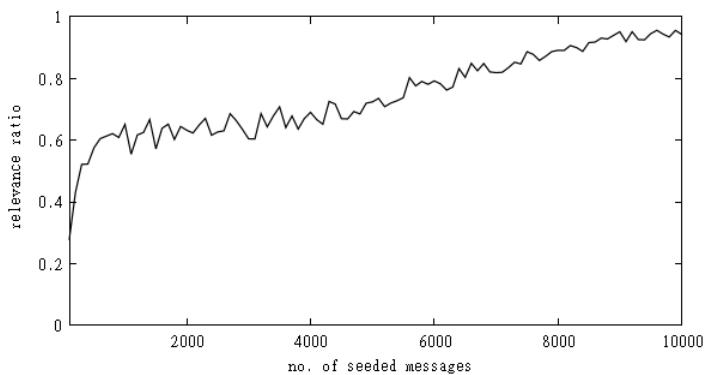
### 3.2 Results

The result of the first experiment is shown in Fig. 1. It shows the correlation between the number of seeded messages and the number of nodes these messages reached. There are two graphs depicting the number of nodes for which these messages are irrelevant and the number of those nodes for which the messages are relevant. The number of nodes receiving irrelevant information is high at the beginning of the experiment (280) but after approximately 1000 seeded messages it drops down to 40 nodes. The system filters out all the irrelevant information at the end of the experiment. While the irrelevant information is reaching a decreasing number of nodes during the experiment, the relevant information reaches 80-160 nodes throughout the experiment. Therefore, the experiment verifies that the approach filters out the irrelevant information while ensuring that relevant information reaches the interested nodes within the social network.

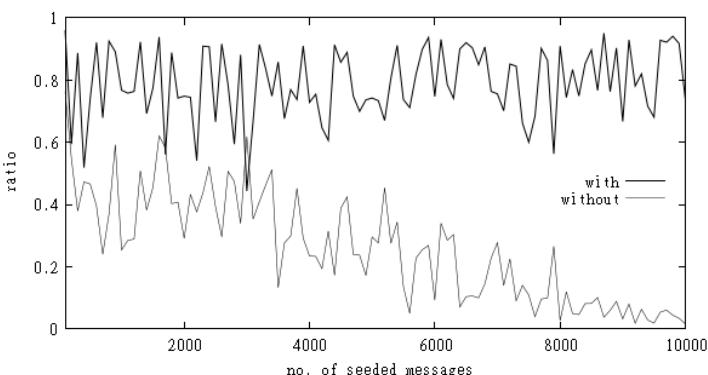
With the same experiment, the relevance ratio is also calculated (see Figure 2). The relevance ratio is the number of nodes to which the message forwarded is relevant out of total number of nodes that received the same message. As illustrated in Figure 2, the relevance ratio increases from 0.3 and reaches around 1.0 at the end of the experiment. There is a steep increase at the beginning of the experiment; this is due to the fact that at the beginning of the experiment all the nodes in the social network have the highest relationship strength with their friend nodes. Therefore, at the beginning of the experiment all messages are shared, also with nodes that have low interest level in the message. But as the system learns through the relationship models, later irrelevant information is filtered out and thus reaches less population in the social network.



**Fig. 1.** Flow of relevant vs. irrelevant information through the network for each seeded messages



**Fig. 2.** Relevance ratios for seeded messages



**Fig. 3.** Flow of irrelevant popular messages with and without consideration of the popularity parameter while filtering irrelevant information

Fig.3 depicts the effect of considering the popularity parameter of the message while filtering irrelevant information. It shows the ratio of irrelevant popular information being displayed to nodes out of the total population. Without considering the popularity parameter, the flow of irrelevant popular information dies out. However, if the popularity parameter is considered, the flow of irrelevant popular information does not diminish and thus the nodes experience a steady flow of serendipitous messages throughout the run.

## 4 Discussion and Related Work

While the evaluation is just a proof of concept, showing that the approach is feasible and leads to the desired results, it also shows that even in a fairly sparse social network, the filtering mechanism does not disconnect the network and social data is propagated to the nodes that are interested.

In comparison with other content-based filtering approaches, e.g. [1], [7], no pure interest model is created, but the interest model is an overlay over the social relationships between users. This allows more flexibility, since filtering differentiate both on the source of information and on the interest level in a category, and in addition, the relationships do not need to be symmetrical. For example, a user A may be interested in receiving news in particular area from user B, but not the reverse. And user A may not be interested in receiving news in this particular area from user C. This allows capturing the trust that a user may have in other users' criterions for recommending / forwarding social updates in a particular area of interest. Trust-based approaches for recommending services, e.g. [9], [5] share some similarities to this approach.

The simulation uses some assumptions which put limitations on the validity of results. In the experiment, the number of categories of interest is fixed to 10, while in reality it can be unlimited (e.g. if tags are used to define the areas). Another assumption is that each shared social update belongs to a single interest area, which is done for the simplicity of calculation in the simulation. In the real world, the same update can be classified into several categories of interest at the same time. A more complex way of deciding whether to present an update to the user and forward it to her friends, based on all the relationship-strengths along the involved categories, would require a bit more computation to be done by each node. The interest value of each node in a category remains static during the single run of the experiment. However, in reality users build or lose interest in different areas over time. This evolution of interest will be accounted naturally in a real world implementation of the approach, since the user's feedback on the received updates will be reflected by the corresponding relationship strengths for these categories. Though the approach has not been implemented and evaluated in a real decentralized OSN, the experiments demonstrated that the relationship model reduces noise over signals and consideration of popularity in information can ensure serendipity for the nodes in the social graph.

## 5 Conclusion and Future Works

The paper presents an approach to filtering social data in decentralized social networks according to the interests of users and their social relationships. The approach addresses the personalization bubble problem by allowing serendipitous messages that are popular to pass through the filter. A model of interest in categories of social data is overlayed over a model of the strength of user interpersonal relationships to achieve filtering of information. The model is updated based on implicit and explicit feedback based on what the user does with the social data. The evaluation of the approach is presented through simulation of the mechanism in a realistic social graph. Future work directions include implementing the mechanism in a real decentralized OSN and evaluating it in a small enterprise social network.

## References

1. Abel, F., Gao, Q., Houben, G.-J., Tao, K.: Analyzing User Modeling on Twitter for Personalized News Recommendations. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 1–12. Springer, Heidelberg (2011)
2. Albert, R., Barabasi, A.-L.: Statistical mechanics of complex networks. *Reviews of Modern Physics* 74(1), 78 (2001)
3. Datta, A., Sharma, R.: GoDisco: Selective Gossip Based Dissemination of Information in Social Community Based Overlays. In: Aguilera, M.K., Yu, H., Vaidya, N.H., Srinivasan, V., Choudhury, R.R. (eds.) ICDCN 2011. LNCS, vol. 6522, pp. 227–238. Springer, Heidelberg (2011)
4. Heckmann, D., Schwartz, T., Brandherm, B.: Decentralized user modeling with UserML and GUMO. In: Workshop on Decentralized, Agent Based and Social Approaches to User Modelling (DASUM), 9th Intl Conference on User Modeling, Edinburgh, Scotland, pp. 61–64 (2005)
5. Nusrat, S., Vassileva, J.: Recommending Services in a Trust-Based Decentralized User Modeling System. In: Ardissono, L., Kuflik, T. (eds.) UMAP 2011 Workshops. LNCS, vol. 7138, pp. 230–242. Springer, Heidelberg (2012)
6. Pariser, E.: *The Filter Bubble: What the Internet is Hiding From You*. Penguin Press HC, New York (2011)
7. Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
8. Tandukar, U., Vassileva, J.: Selective Propagation of Social Data in Decentralized Online Social Network. In: Ardissono, L., Kuflik, T. (eds.) UMAP 2011 Workshops. LNCS, vol. 7138, pp. 213–224. Springer, Heidelberg (2012)
9. Wang, Y., Vassileva, J.: Trust and reputation model in peer-to-peer networks. In: Proceedings Third International Conference on Peer-to-Peer Computing (P2P 2003), pp. 150–157 (September 2003)
10. Wilson, C., Boe, B., Sala, A., Puttaswamy, K.P.N., Zhao, B.Y.: User interactions in social networks and their implications. In: Proceedings of the 4th ACM European Conference on Computer Systems, pp. 205–218. ACM (2009)
11. Yeung, C.A., Liccardi, I., Lu, K., Seneviratne, O., Berners-Lee, T.: Decentralization: The future of online social networking. In: W3C Workshop on the Future of Social Networking Position Papers (2009)

# Ontology-Based Information and Event Extraction for Business Intelligence

Ernest Arendarenko and Tuomo Kakkonen

School of Computing, University of Eastern Finland, Joensuu, Finland  
`{Ernest.Arendarenko,Tuomo.Kakkonen}@uef.fi`

**Abstract.** We would like to introduce BEECON, an information and event extraction system for business intelligence. This is the first ontology-based system for business documents analysis that is able to detect 41 different types of business events from unstructured sources of information. The described system is intended to enhance business intelligence efficiency by automatically extracting relevant content such as business entities and events. In order to achieve it, we use natural language processing techniques, pattern recognition algorithms and hand-written detection rules. In our test set consisting of 190 documents with 550 events, the system achieved 95% precision and 67% recall in detecting all supported business event types from newspaper texts.

**Keywords:** event extraction, ontology-based information extraction, business intelligence.

## 1 Introduction

*Business intelligence* (BI) refers to techniques and tools that aim to provide businesses with support for decision-making by collecting and analyzing information relevant to business leaders. The enormous volume of textual information that is available in online sources and in the internal storage of organizations makes it nearly impossible to follow and analyze all the relevant sources manually or to apply traditional storage and retrieval methods. Useful business information is hidden in a mass of data that is constantly increasing and evolving.

*Information extraction* (IE) deals with extracting relevant and high-quality information from unstructured sources, such as texts and document collections. The extraction process involves representing information contained in textual data in a structured and normalized way. BI offers an interesting field for applying IE techniques. The relevant information in the business context is represented by *named entities* (NE) and the relationships between them. We use the term IE to refer also to the extraction of events that are relevant to the entities that are being tracked.

The purpose of the *Business Events Extractor Component, based on the ONtology* (BEECON) system introduced in this paper, is the automatic extraction of business-related entities (in particular company and product information) and events, by applying text processing methods. BEECON is based on an approach we refer to as

*Ontology-Based Information Extraction* (OBIE). OBIE combines the processes of extracting and storing information and uses an ontology as the basis for a domain knowledge base that stores both verified facts and newly discovered information.

In an OBIE system, the domain knowledge base stored in an ontology and the IE are strongly connected. The domain knowledge base is used to extract relevant information, while the domain knowledge base represented by an ontology is enriched with the results from the IE component. This is a continuous process of enhancing the ontology and knowledge base with entities, facts and their relationships.

The paper is organized as follows: In Section 2 we introduce the background of this work and the most relevant previous research work. Section 3 describes BEECON in detail and explains how the system works. In Section 4, we describe the test settings, evaluation measures and data used for evaluating BEECON. In Section 5, we report on the outcomes of the evaluation, analyze the results and compare them to other related systems. Finally, in the Section 6 we draw conclusions and suggest new directions for future work.

## 2 Background

### 2.1 DAVID

The work described in the current paper is a part of a larger effort to build the *Data Analysis and Visualization aid for Decision-making* (DAVID) system for detecting and analyzing BI by using NLP and text mining methods as well as information visualization techniques. The aim of DAVID is to derive information for business decision-making from written texts by using methods such as entity and event extraction, categorization, clustering and sentiment analysis. DAVID utilizes an ontology for processing and storing information that is relevant to business decision-makers.

The BEECON component presented in this paper is one of the core components of the DAVID system. It is used for detecting relevant parts of the automatically collected source documents (e.g. noise removal and summarization) and for extracting business events from the source documents (e.g. OBIE). Extracted information regarding business events (such as a competitor of a company releasing a new product or a stock market recommendation by an analyst) is transferred into a *Resource Description Framework* (RDF) triple and stored in an ontology that represents newly discovered facts. Later it can be accessed to be used by IE or a knowledge discovery component.

### 2.2 Ontologies and Information Extraction

Ontologies allow for representing domain knowledge in an information system. The meaning of an ontology, in the sense that computer scientists see it, can be defined as “an explicit and formal specification of a conceptualization” (Antoniou & Harmelen, 2008). There are various ways in which ontologies have been applied to assist IE. For

example, Popov (2003) introduced a platform for automatic annotation called Knowledge and Information Management (KIM). Among its features are NE recognition, usage of an ontology and a knowledge base. The idea of KIM is to use metadata to link entities present in a text with their semantic descriptions. The user activates the KIM browser plug-in that highlights recognized entities on the page and provides links to the additional knowledge related to the entity.

IE is used for automatic and semi-automatic ontology population (Vargas-Vera et al., 2007; Maynard et al., 2009) and for enhancing search engines (Hahn et al., 2010). Hahn introduced an alternative approach to searching for information on Wikipedia pages called *Faceted Wikipedia Search*. Knowledge is extracted from an electronic snapshot of millions of Wikipedia pages. After processing it with *DBpedia Information Extraction Framework* (Bizer et al., 2009), the knowledge is stored in an ontology. Currently the system contains 4.7 billion entities that cover a wide set of domains. The authors point out several advantages over existing knowledge bases. Along with its wide coverage, the DBpedia knowledge base automatically updates as Wikipedia evolves. It is multilingual, open and accessible from the Web.

Ontology learning from relevant text collections is a principle that can be applied to many systems, e.g. *TextToOnto* (Cimiano and Völker, 2005) and *OntoLearn* (Navigli et al., 2004). They combine NLP and ML to transfer processing results into an ontology. Buitelaar et al. (2004) described *OntoLT* as a tool used for ontology development that supports interactive extraction of ontologies from texts by using a set of mapping rules. The rules map entities in relation to annotated text collections and concepts in ontology.

### 2.3 Information Extraction and Ontologies for Business Intelligence

There are very few IE systems that are aimed at the BI domain which use ontologies. Maynard et al. (2007) described an IE system for BI that was developed as part of an EU-funded *MUSING* project. The system uses *General Architecture for Text Engineering* (GATE) (Cunningham et al., 2002) for NLP and a domain ontology to aid businesses in areas such as company intelligence and financial risk management. Authors concentrate on extracting information from company profiles, such as the names and address of a company, analyst ratings, shareholders and other details. This data is then utilized for internationalization applications and for merging information from different sources.

Another GATE-based IE system for BI is the *h-TechSight Knowledge Management Portal* (KMP) (Maynard et al., 2005). The purpose of KMP is to aid users in information tracking from online resources by using a domain ontology. The ontology provided by the user contains the required concepts which follow. The system then automatically searches the Web or the selected locations and detects trends and changes in the specified domain. KMP achieved recall and precision scores of 92 and 97 respectively in an experiment by Mikroyannidis et al. (2007). However, these figures refer to the accuracy of entity rather than the event detection. The scope of the experiment, moreover, was rather narrow. Only 13 concepts related to chemical

technologies were considered and the test was carried out by using a test set consisting of 38 documents.

*JV-FASTUS* (Appelt et al., 1993) is an IE system for the business domain that is able to recognize events related to joint ventures from texts written in English and Japanese. It is implemented by using a general architecture of NLP systems: preprocessing, analysis and post processing. In order to find joint ventures from texts, JV-FASTUS uses a so-called “domain pattern recognizer”, that consists of patterns made by domain experts.

*Paramendies* is an ontology-based business decision making support system (Mikroyannidis et al., 2007). It uses a knowledge base that consists of a set of ontologies for the analysis of documents and knowledge extraction. The documents are collected from online resources and the system provides users with tools for the customized analysis of selected market sectors.

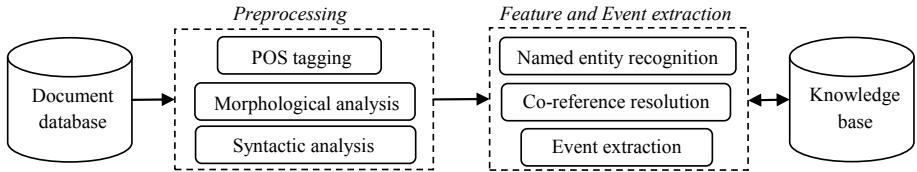
Another notable business-related relation extractor is *PROSPERA*, a pattern-based relation extraction system (Nakashole et al., 2010). Although the authors used academic relation types (such as *hasAcademicAdvisor* and *GraduatedFrom*) as the main evaluation domains, they also compared its performance with another BI-oriented system, *SNOWBALL* (Agichtein & Gravano, 2000). The test set that was used in this comparison was related to extracting the locations of company headquarters.

Our system differs from the systems described above because we are not only extracting specific pieces of information from texts, but also detecting complex patterns and relationships between them by discovering and extracting respective business events. Nevertheless, BEECON also shares common characteristics with the systems described above. Like MUSING and KMP, BEECON is built on top of GATE. PROSPERA resembles BEECON in the sense that it uses pattern-matching algorithms and manually written detection rules. Both BEECON and OntoLT outputs are integrated into an ontology using a dedicated API. The main difference between these two systems is in the detection process: the domain ontology is created before the rules. The mapping rules in BEECON are implemented according to existing ontology classes. This means that only relevant concepts are searched in the input documents, and there is no need for expert validation of each extracted piece of information.

## 3 BEECON

### 3.1 Introduction

BEECON takes inputs from DAVID’s document database that contains automatically fetched relevant documents. Documents go through a processing workflow that is divided into two logical phases: *preprocessing* and *feature & event extraction*. The results of the IE process are finally stored in an ontology. As shown in Figure 1, the link between feature extraction and knowledge base is bidirectional (OBIE).



**Fig. 1.** BEECON workflow

The processing results of the component are transferred into the knowledge base, while the knowledge base is used in the opposite direction: BEECON retrieves known entities (companies, products, etc) which the knowledge base retains and discovers new information about them from input texts. In other words, the component detects instances of ontological classes in annotated texts.

BEECON is built on top of the GATE environment. GATE is a multi-stage NLP engine which provides a set of robust processing resources. BEECON uses a pipeline consisting of various GATE processors described below.

The first phase is the preprocessing of the input documents. A document goes through a cascade of standard processing resources: a sentence splitter, a tokenizer, a part-of-speech tagger and a morphological analyzer. The second phase, feature extraction, consists of NE extraction and co-reference resolution. While preprocessing uses standard GATE resources, this phase uses components that have been tailored to suit the business domain and some new components described later were implemented.

### 3.2 Entity Detection

**Named Entity Recognition.** NEs are recognized by using a rule-based processing resource with a gazetteer to find general types of entities (names, locations) and an ontology to find domain-specific entities such as companies of interest and their products. The gazetteer lists are extended from the standard GATE gazetteers by including more entities relevant to the business domain. New lists representing entities in the business domain are also created such as analyst and rating agencies, stock exchanges and finance metric.

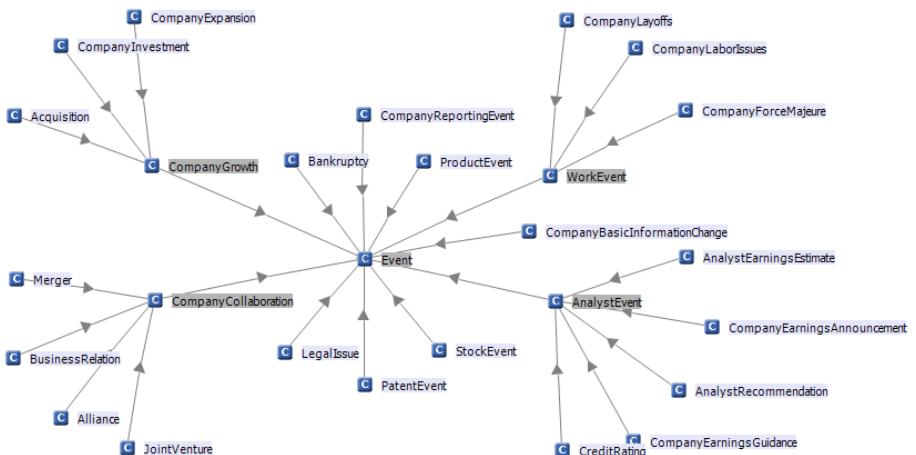
The ontology is used for discovering NEs by using an Onto Root Gazetteer (Damljanovic et al., 2008). The Onto Root Gazetteer component is a GATE processing resource that dynamically constructs an ordinary gazetteer from an ontology. The Onto Root Gazetteer preprocesses all the items in the ontology and annotates them with POS tags and lemmas. The resulting tokens are applied to create, in combination with other GATE components, ontology-based annotations for the given text document. The combination of domain-specific gazetteers and rules, as well as ontological information, allows BEECON to extract entities automatically and without any user input.

**Co-reference Resolution.** It is common in news articles that NEs (such as companies or persons) are referred to by using different linguistic expressions, such as “Standard & Poor’s” and “The financial services company”. In order to extract information correctly, BEECON needs to be able to distinguish which expressions refer to the same entity. This is the task of the co-reference resolution processor. It maps discovered entities that match and resolves the pronouns “he”, “she”, “it”, “its”. The co-referencer is able to handle information about dates, persons, organizations and locations.

The system employs a novel co-reference resolution algorithm for detecting mentions of the company, which is entirely built by the authors for this specific purpose. This algorithm is effective in recognizing references such as “the company”, “the Group”, “the largest carmaker” or “the US-based vendor”. The algorithm first finds entities corresponding to company references. Then, it proceeds to resolve cases where references occur at the start of a sentence. Finally, the remaining references are resolved. Currently, the most of company references on our data sets (Section 4) are detected correctly and successfully connected to the detected references to the company.

### 3.3 Event Detection

**Event Types.** Currently BEECON is able to recognize 41 event types specified in the purpose-built domain ontology called CoProE.



**Fig. 2.** Event types supported by BEECON with four expanded branches

Figure 2 shows all 11 *event categories* detected by BEECON and some of the *event types* from the most important event categories: *Company Growth*, *Company Collaboration*, *Work and Stock Event*, *Bankruptcy* and *Legal Issue*. The *Company Reporting* event category consists of event types including company meetings,

conference calls and debt financing. The *Product Event* category includes event types related to products: releases, recalls and issues. The category *Company Basic Information Change* contains events regarding the company changing its stock listing, name or accounting procedures, as well as reorganizations and changes in employment. The *Stock Event* category describes events such as the company paying dividends, performing stock buybacks, stock splits and other events related to stocks. The category *Patent Event* consists of patent filing and issuance types of event.

**Detection Rules.** The core of events detection is a hand-crafted collection of around 200 grammar rules for *Java Annotation Patterns Engine* (JAPE), which provides the means for defining and processing patterns (Cunningham et al., 2000). A JAPE pattern is a kind of advanced context-aware regular expression consisting of lexical features. Each pattern is a result of text analysis and represents the unique language constructions which are used to describe a particular event. These patterns are then matched with texts to discover and extract events and their features. An example of the BEECON JAPE rule for detecting Company Growth / Acquisition event type is shown in Figure 3:

“IBM said on Monday it has agreed to **pay \$1.7 billion for Netezza.**”

---

Rule: Acquisition\_A\_pay\_money\_for\_B

```
({Organization}):acquirer
({Token, !Split})*
{Token.root == "pay"}
{Money}
{Token.category == "IN"}
({Token, !Split, !Organization})*
({Organization}):acquired
```

**Fig. 3.** Pattern matching rule in action

In the above example rule, the input sentence has been preprocessed, and each token has a POS (“Token.category”) and a morphological annotation including an affix and a root (“Token.root”). For instance, the token word “for” is marked with the following annotation:

```
{category = IN, kind = word, length = 3, orth = lowercase, root = for, string = for}
```

Let us assume that the companies IBM and Netezza are known to BEECON (i.e. are stored in the domain ontology). They are annotated as entities of the type Organization in the NE recognition step. The currency amount “\$1.7 billion” is classified on the same step and is given the Money annotation. When the event detection phase starts, the rule “Acquisition\_A\_pay\_money\_for\_B” is fired, as its pattern is a perfect match for the input sentence, as seen in Figure 3. The time of the event is “Monday”, which will be normalized to Day-Month-Year format on later stages, considering the time of publishing of the news.

The initial set of event detection rules was written in JAPE, based on an analysis of common linguistic patterns present in news articles describing various types of relevant business events. When rules were able to cover most, if not all, of events in a certain event category, we proceeded to update or create rules for other events.

### 3.4 Output

BEECON provides its output in the form of semantic triplets. These triplets are the smallest pieces of ontological information (for example, *[Microsoft, hasStockExchangeTicker, MSFT]*). The output of feature extraction is then formatted into triplets surrounded by event separating tags. The output of BEECON, according to the acquisition example shown in Figure 3, would thus be:

```
[Acquisition, hasAcquiringCompany, IBM]
[Acquisition, hasAcquiredCompany, Netezza]
[Acquisition, hasMoney, $1.7 billion]
```

Each triplet is converted to an RDF syntax that is compliant with our purpose-built API. In addition to storing triplets in the ontology, the API ensures that the information carried by the triplet is a new piece of information, rather than a known fact. The newly discovered information stored in the knowledge base can then be analyzed by using various text mining and knowledge discovery techniques.

## 4 System Development and Evaluation

In order to iteratively develop the system and conduct rigorous evaluations, sentences containing relevant business events data were collected from various online news sources. The data was divided into a development (Section 4.2) and a test set (4.3). Same test settings and evaluation methods were used for evaluations of both data sets.

### 4.1 Data Collection

The data preparation and system improvement started by manually collecting a development set that consisted of hundreds of texts in the business domain. The Google and Bing search engines were used to target relevant news stories from well known online newswire channels such as Reuters, the Wall Street Journal and the New York Times, as well as corporate home pages. Search strings were built from names of event types in the CoProE ontology. Moreover, we used related keywords and combinations of event names and keywords. For example, the retrieval of news that describes a Stock Event / IPO event type was performed using keywords ranging from the obvious “IPO”, “Initial Public Offering” to “stock market debut”.

While one might assert that such a method of collecting data might lead to the creation of a biased dataset (finding only those sentences that contain one of the targeted keywords), the fact that each news article contained dozens of sentences makes such an argument invalid. While the sentence that was found by using the

search engine query would contain the keyword we were expecting, that would most likely not be the case with the rest of the sentences in the news article. In order to make the data set even less prone to bias, we also collected news by browsing news outlets and manually selecting relevant articles.

We selected the newest available stories for the development data set. In order to avoid having exact or near duplicates in the test data, news articles referring to the same event were only taken into consideration if they described the event in a different way and were from different news sources. We also skipped simple cases of events (such as very short sentences like “Apple buys Emagic”) in order to make the process of evaluation more challenging. The simplest test cases had, moreover, been tested informally during the initial system development phase.

Next, each news article was saved in a separate text file. The manually annotated sentences that contained events were stored in a separate file. Each item in the dataset thus consisted of a full original news article with sentences that either did or did not contain an event and the manual event markup for the sentences that contained events.

## 4.2 Test Settings and Evaluation Methods

The evaluation of the accuracy of BEECON was performed as follows: The test files were processed with BEECON. The system processed the text automatically, without any human intervention. Each argument for each triplet in BEECON output was compared against the manually annotated gold standard. An event was considered as correctly extracted only if all of its arguments were correct.

We used the three standard evaluation metrics to measure the accuracy: precision, recall, and F-score. In the context of event extraction, recall indicates the percentage of events which were extracted compared to all the relevant events. Precision defines the proportion of discovered events that were correctly classified. The F-score is a harmonic mean of precision and recall. The higher it is, the better the system performs.

**Development Round.** The development data contained around 250 sentences with events, with an average of 6 sentences per event type. The first evaluation round was carried out by using the initial rule set. The development set had not been used for writing this set. The results of the initial evaluation round were as follows: 95% precision, 81% recall and 88% F-score.

After the first test round was conducted, BEECON was improved in order to achieve a performance as close to 100% precision and recall as possible. This was achieved by improving the event detection rules and adding new ones as well as updating the gazetteers. When the updated set of rules allowed accurate detection of all the event types on the development data, we proceeded to collect texts for the test set.

**Test Round.** The test set consisted of a total of 190 documents with around 6000 sentences. Each document had sentences containing one or more relevant business

events, amounting in total to 550 events. The set was extracted using exactly the same procedure that was used for collecting the development data. The frequency of event types in the test data was not the same for all the event types, because some events occurred more often in news articles (such as a company announcing its earnings) than others (for example, a company expanding or a company declaring a product recall).

## 5 Results

The evaluation results on the test set of 190 documents are: 0.95 precision, 0.67 recall and 0.79 F-score. No improvements were implemented to the system or the detection rules before reporting the evaluation results.

The experimental results showed that the F-score on the test set decreased compared to the development set test by 9 percentage points. The recall rate was affected by false negatives that were caused by sentence patterns which were not covered by the extraction rules (i.e. they were not covered by the initial rule set and did not occur in the development data). The proportion of false positives, or events which were not classified correctly, remained the same as on the development set. Hence, there was no change in precision. Table 1 shows the performance data and number of test cases (TC) for all event categories:

**Table 1.** Performance indicators for event categories

Event category	TC	P	R	F
Analyst Event	141	0.94	0.79	0.86
Bankruptcy	15	0.92	1.00	0.96
Company Basic Information Change	29	0.93	0.72	0.77
Company Collaboration	51	0.97	0.67	0.77
Company Growth	94	0.93	0.72	0.81
Company Reporting Event	40	1.00	0.80	0.87
Legal Issue	19	1.00	0.17	0.29
Patent Event	9	1.00	0.88	0.93
Product Event	30	0.95	0.28	0.43
Stock Event	70	0.97	0.77	0.85
Work Event	52	1.00	0.69	0.80
<b>Total</b>	<b>550</b>	<b>0.95</b>	<b>0.67</b>	<b>0.79</b>

The performance of BEECON was relatively high on some event categories, which is indicated by strong F-scores in Table 1. The precision on all the event categories was adequate, ranging from 0.92 to 1.00. In most categories, the precision was also on an acceptable level. The performance on the Product and Legal Issue categories was quite unimpressive, as the recall rates were low (0.28 and 0.17 respectively). These results indicated that more detection rules were also required to improve recall for Company Collaboration and Work event categories as well, although almost all of the

detected events of these categories were correctly classified. To summarize, the performance data shows that precision rates were evenly distributed between the event categories, reaching more than 92%, while the recall seemed to be dispersed across the table.

The performance data for the six most accurate and the four least accurate event types is shown in Table 2. A group of event types that showed the worst performance had a common characteristic: these events are described by using more complicated sentence structures, and there also are many ways of saying the same thing.

**Table 2.** The six most and the four least accurate event types

Event type	P	R	F
Company Basic Information Change / Accounting Change	1.00	1.00	1.00
Company Reporting Event / Conference Call	1.00	1.00	1.00
Stock Event / Stock Split	1.00	1.00	1.00
Bankruptcy	0.92	1.00	0.96
Analyst Event / Company	0.92	0.86	0.89
Earnings Announcement			
Company Collaboration / Alliance	1.00	0.80	0.89
Analyst Event /			
Analyst Recommendation	1.00	0.15	0.27
Legal Issue / Company			
Legal Issues	1.00	0.17	0.29
Company Basic Information Change / Employment Change	1.00	0.18	0.31
Product Event / Product Recall	1.00	0.20	0.33

The test set also included sentence patterns that had not been encountered before for these event types, causing an increase in false negatives. Such difficulties did not occur in the case of the six event types which showed the best levels of detection accuracy.

## 5.1 Comparison to Existing Systems

Since none of the related systems are capable of detecting such a wide range of event types as BEECON, we are not able to provide a direct comparison between these systems and BEECON. JV-FASTUS achieved 34% of recall, 56% of precision and 43% of the F-score for the recognition of events related to joint ventures in texts written in English (Appelt et al., 1993). Our system supports the detection of such events (Company Collaboration / Joint Venture event type). In our test data BEECON was able to find 50% of Joint venture events (recall), of which 100% had been correctly extracted (precision). The F-score was 74%. BEECON, hence, compares very favorably to JV-FASTUS in terms of the capability of detecting joint venture events.

PROSPERA was reported to achieve 85% recall and 91% precision in the task of extracting the locations of company headquarters (Nakashole et al., 2010). BEECON does not extract such information, so these figures cannot be directly compared to any event types supported by our system. The event types that perhaps match the best of this type of extraction are the ones belonging to the Company Basic Information Change category, on which the performance of BEECON was 93% for precision and 72% for recall over all the 5 event types in that category.

Evaluation of the MUSING system on a company profiles application showed 86% precision, 94% recall and an 84% F-score (Maynard et al., 2007). While the F-score is five percentage units higher than that achieved by BEECON on our test data, it is not possible to make a direct comparison between our system and MUSING, due to the differences in the type of information that was extracted. It is also worth noting that we deliberately avoided including the simple test cases consisting of short sentences in our test set. No measure of event detection accuracy has been released for KMP and Paramendies. Hence, we were not able to compare our results for these systems.

## 6 Conclusions

In this paper, we have presented BEECON, an IE system for BI that uses an ontology for storing and processing domain knowledge. Due to its domain-specific language resources, the system has a wide coverage of the business events domain. The system also includes a novel co-reference resolution algorithm for company names. Being built on top of well-known Java tools, it introduces new components as well as improvements to the original components.

During the first iteration of system development and evaluation, we used a data set consisting of around 250 sentences to test the accuracy of the system with an initial set of 112 detection rules and to improve the coverage of the rules. This resulted in an improved set of 160 rules and precision and recall figures of 95% and 81% respectively.

On the evaluation on the test set consisting of 550 sentences, the system achieved 67% recall, 95% precision and a 79% F-score. These results compared favorably to those reported for other business-related IE systems. The precision level was high and comparable to that of the existing system. Recall was 29 percentage units higher than the related system, JV-FASTUS, achieved in detecting joint venture events. However, there was still room for improvement on the recall rate of BEECON. The system was revised based on the test set, was provided with more rules and received updates to the existing ones. The next system development and evaluation round that is currently in progress consists of 2000 sentences with events. The preliminary results show a similar level of performance.

There are several directions for future work in developing BEECON. The first issue we will work on is the coverage of the rules. The rules are improved and added via an iterative process. We first collect a new set of data, test the performance of the current version of the system and finally improve the rules, until nearly 100% precision and recall is achieved.

The detection of events in the current version of the BEECON system is based on manually written rules. While manually coded “strict” rules may deliver higher precision and require a smaller training set than Machine Learning techniques, ML methods often provide a higher recall rate. In an events detection task, this approach may help to discover events covered by previously unseen patterns. The mentioned approaches have pros and cons, and we are looking forward to combining these to achieve a balanced performance. ML is planned to be implemented in the following releases of the component.

## References

- Agichtein, E., Gravano, L.: Snowball: extracting relations from large plain-text collections. In: Proceedings of the Fifth ACM Conference on Digital libraries (DL 2000), pp. 85–94. ACM, New York (2000)
- Antoniou, G., van Harmelen, F.: A Semantic Web Primer, 2nd edn. The MIT Press, Cambridge (2008)
- Appelt, D., Hobbs, J., Bear, J., Israel, D., Kamayama, M., Tyson, M.: SRI: description of the JVFASTUS system used for MUC-5. In: Proceedings of MUC 1993, Baltimore, Maryland, USA (1993)
- Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web* 7(3), 155–165 (2009)
- Buitelaar, P., Olejnik, D., Sintek, M.: A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) *ESWS 2004. LNCS*, vol. 3053, pp. 31–44. Springer, Heidelberg (2004)
- Cimiano, P., Völker, J.: Text2Onto -A Framework for Ontology Learning and Data-driven Change Discovery. In: Montoyo, A., Muñoz, R., Métais, E. (eds.) *NLDB 2005. LNCS*, vol. 3513, pp. 227–238. Springer, Heidelberg (2005)
- Cunningham, H., Maynard, D., Tablan, V.: JAPE: a Java Annotation Patterns Engine, 2nd edn. Technical Report, CS-00-10, University of Sheffield, Department of Computer Science (2000)
- Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA (2002)
- Damljanovic, D., Tablan, V., Bontcheva, K.: A Text-based Query Interface to OWL Ontologies. In: Proceedings of the 6th Language Resources and Evaluation Conference, Marrakech, Morocco (2008)
- Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., Düwiger, H., Scheel, U.: Faceted Wikipedia Search. In: Abramowicz, W., Tolksdorf, R. (eds.) *BIS 2010. LNBP*, vol. 47, pp. 1–11. Springer, Heidelberg (2010)
- Maynard, D., Funk, A., Peters, W.: SPRAT: a tool for automatic semantic pattern based ontology population. In: International Conference for Digital Libraries and the Semantic Web, Trento, Italy (2009)
- Maynard, D., Saggion, H., Yankova, M., Bontcheva, K., Peters, W.: Natural Language Technology for Information Integration in Business Intelligence. In: Abramowicz, W. (ed.) *BIS 2007. LNCS*, vol. 4439, pp. 366–380. Springer, Heidelberg (2007)

- Maynard, D., Yankova, M., Kourakis, A., Kokossis, A.: Ontology-based information extraction for market monitoring and technology watch. In: Proceedings of the European Semantic Web Conference, End User Aspects of the Semantic Web Workshop, Heraklion, Crete, Greece (2005)
- Mikroyannidis, A., Theodoulidis, B., Persidis, A.: PARMENIDES: towards business intelligence discovery from web data. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA (2007)
- Nakashole, N., Theobald, M., Weikum, G.: Find your advisor: robust knowledge gathering from the web. In: Proceedings of the 13th International Workshop on the Web and Databases. ACM, New York (2010)
- Navigli, R., Velardi, P., Cucchiarelli, A., Neri, F., Cucchiarelli, R.: Extending and Enriching WordNet with OntoLearn. In: Proceedings of The Second Global World Net Conference, Brno, Czech Republic (2004)
- OWL Web Ontology Language (2004), <http://www.w3.org/TR/owl2-overview> (retrieved October 10, 2011)
- Vargas-Vera, M., Moreale, E., Stutt, A., Motta, E., Ciravegna, F.: MnM: semi-automatic ontology population from text. In: Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems. Integrated Series in Information Systems, vol. (14), pp. 373–402. Springer Science + Business Media, LLC, New York (2007)

# Modelling Highly Symmetrical Molecules: Linking Ontologies and Graphs

Oliver Kutz<sup>1</sup>, Janna Hastings<sup>2</sup>, and Till Mossakowski<sup>1,3</sup>

<sup>1</sup> Research Center on Spatial Cognition, University of Bremen, Germany

<sup>2</sup> European Bioinformatics Institute, Cambridge, UK

<sup>3</sup> DFKI GmbH Bremen

okutz@informatik.uni-bremen.de, hastings@ebi.ac.uk,

Till.Mossakowski@dfki.de

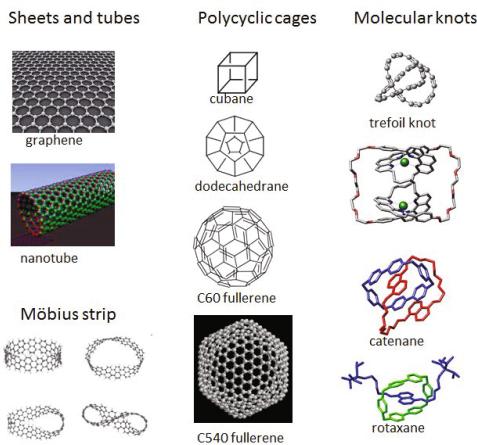
**Abstract.** Methods for automated classification of chemical data depend on identifying interesting parts and properties. However, classes of chemical entities which are highly symmetrical and contain large numbers of homogeneous parts (such as carbon atoms) are not straightforwardly classified in this fashion. One such class of molecules is the fullerene family, which shows potential for many novel applications including in biomedicine. The Web Ontology Language OWL cannot be used to represent the structure of fullerenes, as their structure is not tree-shaped. While individual members of the fullerene class can be modelled in standard FOL, expressing the properties of the class as a whole (independent of the count of atoms of the members) requires second-order quantification. Given the size of chemical ontologies such as ChEBI, using second-order expressivity in the general case is prohibitively expensive to practical applications. To address these conflicting requirements, we introduce a novel framework in which we heterogeneously integrate standard ontological modelling with monadic second-order reasoning over chemical graphs, enabling various kinds of information flow between the distinct representational layers.

## 1 Introduction and Motivation

Organic chemistry has seen a dramatic increase in available data in recent years, tracking progress in the search for novel therapeutics.<sup>1</sup> However, large-scale data that are not appropriately organised can be more of a burden than a benefit. Ontologies and knowledge-based methods for automated classification are increasingly harnessed to address this challenge. ChEBI – Chemical Entities of Biological Interest – is a chemical ontology that is widely used to organise and classify chemical data [4]. However, ChEBI is manually maintained, reducing its scalability. Methods for automated classification of chemical entities depend on algorithms which reduce complex molecular graphs to lists of interesting parts and properties, such as atomic constituents and groups, charges and overall molecular weight. Knowledge representation and reasoning for chemistry has also largely been dominated by this paradigm [13][12][14].

<sup>1</sup> The basic ideas formulated in this paper were previously presented at the Deep Knowledge Representation Workshop DKR-11, Banff, Canada, 2011 (2nd prize in the DKR competition).

In recent years there has been a progression in capacity for the synthesis of highly symmetrical, polycyclic chemical entities, which are made up of a very small number of part *sorts* (e.g. mainly carbon atoms) with a very large number of actual parts. Polycyclic carbon molecules show incredible topological versatility, not only forming spheres, tubes and sheets, but also molecular Möbius strips [8][11] and knots [7][17], as illustrated in Figure 1. These molecules elicit increased interest following advances in synthesis methods towards a nanoscale molecular ‘machinery’ with carefully designed shapes that are able to rival the power and scale of biological machinery [19].



**Fig. 1.** Some examples of highly symmetric molecules, constituted almost entirely by carbon atoms. The overall arrangements of atoms in the molecules, rather than the nature of functional groups, characterise their types.

to represent deep knowledge about this class of molecules. We will therefore complement OWL with the more expressive formalism of monadic second-order logic (MSOL).

## 2 Background

**OWL Representation.** Chemical entities can be represented and exchanged in the form of chemical graphs, in which atoms form the vertices and covalent chemical bonds the edges. However, complex graphs that contain cycles cannot be faithfully modelled at the class level in OWL due to the requirement that all axioms in OWL have models shaped like a tree [12].

Since their parts are homogeneous, listing part types cannot distinguish distinct such classes. Rather, such molecules must be characterised by their *shape* or *topology*. For example, fullerene molecules form spherical or ellipsoidal cages, Möbius molecules display classical Möbius topologies, while molecular knots and interlinked chains display the topological and shape properties of macromolecular knots and chains. To adequately represent knowledge about these molecules requires the ability to describe and reason over features which apply to the entire molecular graph (i.e. the connection of atoms via bonds).

In what follows, we will argue that formalisms with limited expressivity such as OWL are not sufficient

Can we recognise members of the fullerene class of molecules based on the structure of their chemical graphs? A formulation of C<sub>60</sub> fullerene and a graphene of 60 atoms might refer explicitly to their differing shapes in order to allow automated reasoning to distinguish them, using axioms such as *hasPart* only CarbonAtom and *hasShape* some Sphere or *hasShape* some Flat.

However, this approach clearly does not allow automated reasoning to deduce the class of the molecule based on the properties of the molecular graph, since a human has to specify the shape of the molecule. Following this pattern, a different shape has to be defined for every differently shaped molecule, with no means of automatically discerning relationships or similarity between the stated shapes. Furthermore, those properties of the molecules that depend on their shapes are not explained by the information contained in the ontology. Many of the properties of fullerenes stem from the fact that they can enclose other molecules inside their cage structure, a property not shared by graphene. The properties of molecular knots stem from the fact that they are mechanically interlocked. What is required is a framework that is able to *define* classes of molecules based on properties of the graphs of their members, and then *deduce* which molecular graphs belong to these classes.

**Description Graphs, Rules and FOL.** Cycles can be represented adequately in rules, which are combined with OWL for ontology engineering in the DL-safe rule extension [16]. The DL-safe rules extension, however, is applicable only to explicitly named objects in the ontology (individuals), to ensure decidability of the resulting knowledge base. This means that it is not possible to reason at the class level about highly symmetric molecules using this formalism.

This shortcoming motivated the introduction of description graphs [15], an OWL extension for expressing the structure of complex objects at the class level. However, the knowledge base is still constrained in that the OWL axioms and the edge properties in the description graphs must be kept separate. The reasoning capability of the framework is limited to what can be expressed in “graph-safe” rules: rules which do not mix graph edge properties and OWL object properties. Furthermore, there are inherent limitations in the use of rules for reasoning, since there is no  $\forall$  quantification in the rules formalism, which means that properties of all atoms in a given graph cannot be used for reasoning [12]. In an effort to relax these limitations, a radically different semantics has recently been proposed, based on logic programming: description graph logic programs [14]. Since the semantics from logic programming ensures decidability in a different way to the OWL model-theoretic semantics, there is no need for property separation, thus the ontology designer may interchangeably use OWL and description graph properties in creating the knowledge base. This formalism allows representation and reasoning with cyclic chemical structures at the class level. It is possible, for example, to define a particular member of the fullerenes class, such as dodecahedrane, and to use reasoning for detection of cycles of fixed lengths. However, it is not possible to express the properties of fullerenes as a whole. Using full FOL it is possible to get very close to a definition for the fullerenes, including axioms that every atom must have 3 or 4 bonds, every atom must belong to a cycle, and every cycle (face) must

have 5 or 6 members. However, such constraints (“local perspective”) cannot allow correct classification in all cases. For example, fullerenes of different sizes (for example, C540, C240 and C60) can be *nested* inside one another. The local perspective at each atom and at each face correctly matches the best definition that is possible to specify in FOL. Yet, this should be classified as a *complex* consisting of multiple fullerenes, rather than as itself a (single) fullerene molecule. To distinguish the complex from a single molecule, the second-order construct of *graph connectedness* is needed. However, it is well-known that connectedness is not first-order definable [5].

### 3 Properties of Graphs for Chemical Classes

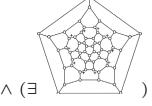
In order to distinguish between fullerenes, graphenes, strips and Möbius strips, we need to define some properties of graphs based on chemical graph theory [18]. For simplification we will assume that all graphs are *finite*, which is true of all graphs corresponding to real chemical entities.

**Planar Polyhedral Graphs.** A chemical graph is **planar** if it can be drawn on a flat plane without any edges crossing. Overwhelmingly, most chemical entities can be described by planar graphs. The only exception found in a recent analysis of public compound databases were Möbius-like molecules [8]. A graph is **cubic** if all vertices have degree three, i.e. are connected to three other vertices. It is **connected**, if any two vertices are connected by a path, and it is **3-connected** if it is connected and remains so after removal of any two vertices. A graph is **polyhedral** iff it is the graph of some convex polyhedron. By Steinitz’ theorem (1922), this is equivalent to being 3-connected and planar (see [20]). Indeed, polyhedral graphs, while being planar (2D), are typically represented as convex polyhedra (3D). A **polycyclic cage** is any polyhedral graph. Chemical examples include cubane, tetrahedrane, and of course all fullerenes. A **fullerene** is a cubic polyhedral graph consisting of hexagons and pentagons only. By the Euler formula for polyhedra, one can show that the number of pentagons must always be 12. A **closed nanotube** is a fullerene which is extended into a tube shape with a circular extension consisting only of hexagons between the two ends, the latter consisting of two hemispheres of the buckyball structure. An **open nanotube** is a cubic polyhedral graph consisting of hexagons and two non-hexagons (the two non-hexagons are the outer boundaries).

**Planar Non-polyhedral Graphs.** A **graphene** is a planar graph consisting of hexagons and one face (the outer boundary) not necessarily being a hexagon, where all vertices involved in the outer boundary have degree two or three, while the remaining vertices have degree three.

**Non-planar Graphs.** A **Möbius strip** is non-planar graph, consisting of hexagons and one non-hexagon (the outer boundary).

## 4 Describing Molecule Graph Classes in MSOL

$Cubic \Leftrightarrow \forall x. \exists! 3y. edge(x, y)$   
 $degree_n(x) \Leftrightarrow \exists! ny. edge(x, y)$   
  
 $Planar \Leftrightarrow \neg(\exists \dots \neg(\exists \dots))$   
 $Connected\_Subgraph(C) \Leftrightarrow \forall D \subseteq C, E \subseteq C. C = D \cup E \Rightarrow \exists u \in D, v \in E. edge(u, v)$   
 $Connected \Leftrightarrow \exists C. \forall x. x \in C \wedge Connected\_Subgraph(C)$   
 $Cycle(C) \Leftrightarrow Connected\_Subgraph(C) \wedge \forall x \in C \exists y \in C. edge(x, y)$   
 $Three\_Connected \Leftrightarrow \forall x, y. Connected\_Subgraph(V \setminus \{x, y\})$   
 $Polyhedron \Leftrightarrow Planar \wedge Three\_Connected$   
 $Polycyclic\_Cage \Leftrightarrow Polyhedron$   
 $Face(C) \Leftrightarrow Cycle(C) \wedge Connected\_Subgraph(V \setminus C) \wedge \forall u, x, y, z \in C. edge(u, x) \wedge edge(u, y) \wedge edge(u, z) \rightarrow (x = y \vee x = z \vee y = z)$   
 $Pent(C) \Leftrightarrow Cycle(C) \wedge \exists! 5x. x \in C$   
 $Hex(C) \Leftrightarrow Cycle(C) \wedge \exists! 6x. x \in C$   
 $Carbon\_Allotrope \Leftrightarrow \forall x. Carbon(x)$   
 $Fullerene \Leftrightarrow Carbon\_Allotrope \wedge Polycyclic\_Cage \wedge Cubic \wedge \forall C. Face(C) \rightarrow Pent(C) \vee Hex(C)$   
 $Closed\_Nanotube \Leftrightarrow Fullerene \wedge$   


$Open\_Nanotube \Leftrightarrow Carbon\_Allotrope \wedge Polyhedron \wedge Cubic \wedge \exists B, C. Face(B) \wedge Face(C) \wedge B \neq C \wedge \forall D. Face(D) \rightarrow B = D \vee C = D \vee Hex(D)$

$Graphene \Leftrightarrow Carbon\_Allotrope \wedge Planar \wedge \exists B. Face(B) \wedge (\forall x \in B. degree_2(x) \vee degree_3(x)) \wedge \forall C. Face(C) \rightarrow B = C \vee (Hex(C) \wedge \forall x \in C. degree_3(x))$

$Moebius\_Strip \Leftrightarrow \neg Planar \wedge \exists B. Face(B) \wedge \forall C. Face(C) \rightarrow B = C \vee Hex(C)$

**Fig. 2.** MSOL formalisation of molecule classes

most graph classes that we are interested in, even with the above syntactic sugar, often the axioms can become cumbersome and large. We therefore additionally use the *nested conditions* of [9,10]. The simplest and most prominent formulas here are of form  $(\exists G)$ , where  $G$  is a graph with some edges annotated with  $\oplus$ . The semantics is that  $G$  can be injectively embedded into the given graph, where each edge labelled with  $\oplus$  may be mapped to a finite path (this may be used to express that a certain  $G$  is a minor of the given graph). The MSOL formalisation of the above notions is shown in Fig. 2. Note that graph classes are represented as MSOL model classes; this means that e.g. a graph is cubic if and only if it (when seen as a MSOL model) satisfies the formula  $\forall x. \exists! 3y. edge(x, y)$ . The correctness of the definition of polyhedral graph follows from Steinitz' theorem discussed above, and the correctness of the definition of planarity follows from Kuratowski's characterisation in terms of forbidden minors.

We want to formalise the definitions of graph classes such that membership in a graph class can be machine-checked. It has been noted (see [3]) that the role finite automata play for the specification of word languages is played by *monadic second-order logic* (MSOL) for expressing graph properties and defining graph classes. Although the general problem is NP-complete, monadic second-order logic for graphs can be model-checked quite efficiently; indeed, for graphs with bounded tree-width, model checking can be done in linear time. MSOL for graphs consists of untyped first-order logic, extended with quantification over sets (and membership in such sets). We assume binary predicates  $edge$ ,  $edge_2$  and  $edge_3$  for all bonds, double bonds and triple bonds, respectively. We also assume unary predicates like  $Carbon$  for the atoms (and suitable atom classes) in the periodic table. When writing MSOL formulas, we use syntactic sugar like unique-existential quantifiers and number quantifiers, which can easily be coded out even in first-order logic. We also will freely use standard set-theoretic notation where it can easily be coded out into MSOL.  $V$  denotes the set of all vertices. While the expressive power of MSOL suffices to axiomatise

## 5 Connecting Ontology and Graph Layers

We have seen that monadic second-order logic combined with nested conditions provides a convenient formalism for adequate formalisation of graph-conditions relevant for the modelling of chemicals. However, how can such specifications of graph classes be related to existing ontologies of molecules such as ChEBI [4] that are formulated in a light-weight ontology language like OWL-EL? Clearly, one cannot expect to be able to formalise deeper graph-theoretical properties in OWL. However, using the MSOL formalisation, we can build what we call a **grounded ontology**: class names such as fullerene are equipped with a (or several) formal MSOL specification(s), and specific instances, i.e. object names are equipped with concrete graphs. Such an association, if done systematically, will give rise to a number of automated reasoning problems such as model and subclass checking, deduction in MSOL, and abduction, here put into a new context. To motivate the following definition, note the following considerations. Two (different) ontology classes may be equipped with the same MSOL theories. This reflects an *intensionality* in the definition of the ontological classes which, although having different ontological definitions, denote the same structural class of molecules. Conversely, one and the same ontology class may be equipped with different MSOL theories. This corresponds to an intensionality in the realm of graph classes, where different descriptions of a graph class may be found in the literature (e.g. if the molecule has different structural variants). Therefore, we express soundness of the relation between ontology classes and monadic second-order theories as functionality modulo logical equivalence.

We consider the ontology as the primary, and the graph-based formalisation as a secondary source of information. This is reflected in the following formal definition for ontologies expressed in  $\mathcal{ALC}$ .<sup>2</sup>

**Definition 1.** Fix an  $\mathcal{ALC}$  ontology  $O = \langle T, A \rangle$ , where  $T$  is a TBox, and  $A$  is an ABox. Let  $\mathbf{C}$  be the set of  $\mathcal{ALC}$  (sub)concept descriptions (atomic or complex) and  $\mathbf{I}$  the set of object names appearing in  $O$ ,  $\mathfrak{T}$  a set of finite MSOL theories,<sup>3</sup> and  $\mathfrak{G}$  a set of MSOL finite undirected graphs. An **ontology-graph association** (**oga** for short) is a pair of relations  $\rightsquigarrow = \langle \rightsquigarrow_T, \rightsquigarrow_A \rangle$ , where

$$\rightsquigarrow_T \subseteq \mathbf{C} \times \mathfrak{T} \quad \text{and} \quad \rightsquigarrow_A \subseteq \mathbf{I} \times \mathfrak{G}$$

$\rightsquigarrow$  is **total** if for any concept  $C \in \mathbf{C}$  and object  $a \in \mathbf{I}$  there exist  $T \in \mathfrak{T}, G \in \mathfrak{G}$  such that  $C \rightsquigarrow_T T$  and  $a \rightsquigarrow_A G$ .

$\rightsquigarrow$  is **sound** if for all  $O \models C \sqsubseteq D$  and  $a : C$  and  $b : D$  we have:  $C \rightsquigarrow_T S, D \rightsquigarrow_T T$  implies  $S \models_{\text{MSOL}} T$  and  $a \rightsquigarrow_A G$  implies  $G \models_{\text{MSOL}} S$  and  $b \rightsquigarrow_A H$  implies  $H \models_{\text{MSOL}} T$ .

$\rightsquigarrow$  is **complete** if for all  $C \rightsquigarrow_T S, D \rightsquigarrow_T T$  we have:  $S \models_{\text{MSOL}} T$  implies  $O \models C \sqsubseteq D$  and for all  $a \rightsquigarrow_A G$  and  $b \rightsquigarrow_A H$  with  $G \models_{\text{MSOL}} S$  and  $H \models_{\text{MSOL}} T$  we have  $a : C$  and  $b : D$ .

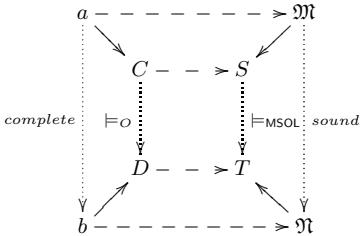
$\rightsquigarrow$  is **graph-extensional** if  $C \rightsquigarrow_T S, C \rightsquigarrow_T T \implies \models_{\text{MSOL}} S \leftrightarrow T$ .

$\rightsquigarrow$  is **class-extensional** if  $C \rightsquigarrow_T S, D \rightsquigarrow_T S \implies O \models C \equiv D$ .

---

<sup>2</sup> We focus here on OWL ontologies with at most  $\mathcal{ALC}$  expressivity. However, all definitions carry over to FOL ontologies mutatis mutandis.

<sup>3</sup> We can therefore meaningfully use Boolean combinations of such theories.



**Fig. 3.** Ontology-graph association: OWL and MSOL

MSOL term	chemical notion	OWL term
MSOL theory	molecule class	OWL class
graph	molecule	OWL individual
model checking	instance checking	OWL ABox checking
logical entailment	subclass relation	OWL subclass (TBox)
consistent theory	nonempty class	satisfiable OWL class

**Fig. 4.** Grounded ontology: correspondences between MSOL, OWL, and chemical notions

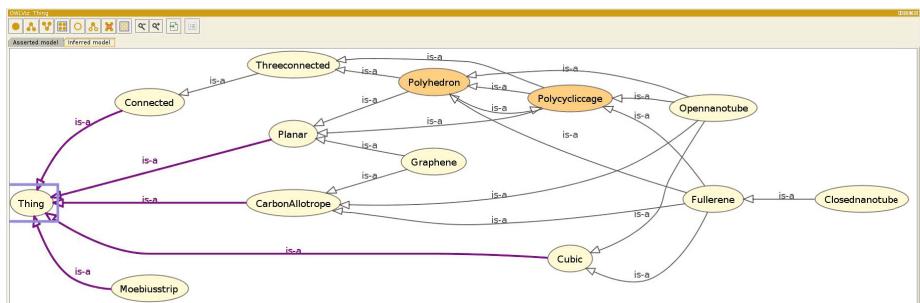
**Proposition 1.** *Completeness implies class-extensionality (not vice versa), and soundness implies graph-extensionality (not vice versa).*

The logical structure of ontology-graph associations is illustrated in Figure 3. Note that the new aspect here is that there is a *shift of levels*, namely graphs are models in MSOL, while they are individuals in OWL. The correspondences (including reasoning) are as follows: Rather than *integrating* or *combining* different logics and running the risk of losing any of the desirable properties of the special purpose formalisms, our approach realises an interlinked formalisation of different aspects of the domain of chemical molecules that relies on a mapping between the different layers. Whilst (lightweight) ontology languages are used to cope with the rather large chemical ontologies, MSOL is used to adequately capture some of the ontologically relevant spatial structure of molecule classes. Obviously, to make this approach worthwhile, we need to establish systematic ways of exchanging information between these two layers of different abstraction and expressiveness.

**Deduction.** Proven entailment between MSOL theories may be used to assert subsumption between the corresponding classes in the chemical ontology (e.g. the ontology in Fig. 5 has been obtained in this way). This corresponds to ensuring completeness as defined above.

**Abduction.** Abduction [6] can be used to hypothesise new correspondences. For example, given  $A \sqcap B \rightsquigarrow_T T_1$ ,  $A \rightsquigarrow_T T_2$  and  $\models_{MSOL} T_1 \leftrightarrow T_1 \wedge T_2$ , this may have the explanation  $B \rightsquigarrow_T T_3$ .

Other related reasoning problems are **induction**, i.e. given a set of example molecules (e.g. MOL datafiles), learn a corresponding graph class specified in MSOL, and **model checking**, i.e. given a Möbius strip, check (using a tool such as [1]) that it is non-planar. Moreover, to show that it is additionally a loop is an interesting and non-trivial subsumption check. Although MSOL entailment is in general undecidable, logical entailment in second-order logic can be approximated with automated theorem provers like LEO-II (<http://www.ags.uni-sb.de/~leo/>). An initial OWL ontology computed from the logical implications among the MSOL axiomatisations given in Sec. 4 is shown in Fig. 5 (available at [ontohub.org](http://ontohub.org)). Here, the implications between the graph classes are mostly definitional and therefore easy to check automatically, and trigger the creation of corresponding subsumptions in the ontology.



**Fig. 5.** Class hierarchy computed from MSOL implications

## 6 Conclusions

Representation and reasoning with structured objects such as molecules is still an area of active research and development for ontologists and chemoinformaticians. Chemical ontologies such as ChEBI [4] provide one solution to this problem through careful manual classification. Formal ontology aims to supplement such manual efforts with explicit computable knowledge representation and accompanying automated reasoning. We here focus on a particularly interesting and challenging class of molecules for such formalisation, and examine an approach which uses the expressive power of monadic second-order logic (MSOL) to formalise properties that cannot be defined in OWL, proposing to systematically link the two layers.

Compared to algorithmic approaches of molecule classification, we can offer a language for a *declarative* description of molecules and molecule classes, which offers a path to not only instance checking (as in the algorithmic case), but also to subclass checking, through MSOL theorem proving. We propose to combine this with OWL ontologies such as ChEBI, thus obtaining a “grounded ontology”, where OWL subclass relations can be verified or inferred by looking at the corresponding graph properties in MSOL.

In the semi-automatic generation of MSOL theories chemical graphs datasets via inductive reasoning, a problem that has to be considered is that abstracting a graph class from a finite number of sample molecules can sometimes produce ambiguous results. Importantly, classes of molecules conforming to particular graph theories may have characteristic emergent properties in terms of chemical and biochemical reactivity and activity profile that none of the superclasses (with less restrictive accompanying graph theories) display. The activity and reactivity properties of molecules would need to be included in a separate ontological layer within the framework we describe. Note that the MSOL approach can only classify molecules based on properties of their graphs. However, from a graph-theoretic point of view, the molecular trefoil is equivalent to a simple loop. In order to distinguish it from the loop, one has to consider its embedding into Euclidean space, and use knot theory. Future work should consider invariants from knot theory (such as genus, polynomials and groups) in a similar role as that in which we presently propose to use MSOL. Also, the results of computational graph theory will be useful, e.g. for optimising parts of the model checking for graphs.

**Acknowledgements.** Work on this paper has been supported by the DFG-funded collaborative research centre SFB/TR 8 ‘Spatial Cognition’. The authors are grateful to Berthold Hoffmann, Hans-Jörg Kreowski, Ian Pratt-Hartmann, Michael Grüninger and Despoina Magka for valuable discussions and comments.

## References

1. Arnborg, S.: A general purpose MSOL model checker and optimizer based on Boolean function representation. Technical report, KTH, Stockholm, Sweden (1994)
2. Caetano, E.W.S., Freire, V.N., dos Santos, S.G., Galvao, D.S., Sato, F.: Möbius and twisted graphene nanoribbons: stability, geometry and electronic properties. *The Journal of Chemical Physics* 128(164719) (2008)
3. Courcelle, B., Engelfriet, J.: Graph structure and monadic second-order logic—A language theoretic approach. Cambridge University Press (2011)
4. de Matos, P., Alcántara, R., Dekker, A., Ennis, M., Hastings, J., Haug, K., Spiteri, I., Turner, S., Steinbeck, C.: Chemical Entities of Biological Interest: an update. *Nucl. Acids Res.* 38, D249–D254 (2010)
5. Ebbinghaus, H.-D., Flum, J.: Finite Model Theory. Springer (2005)
6. Elsenbroich, C., Kutz, O., Sattler, U.: A Case for Abductive Reasoning over Ontologies. In: Proc. of OWLED 2006 (2006)
7. Cancill, J., et al.: From classical chirality to topologically chiral catenands and knots. In: Supramolecular Chemistry I Directed Synthesis and Molecular Recognition. Topics in Current Chemistry, vol. 165, pp. 131–162. Springer, Heidelberg (1993)
8. Wester, M.J., et al.: Scaffold Topologies. 2. Analysis of Chemical Databases. *Journal of Chemical Information and Modeling* 48(7), 1311–1324 (2008)
9. Habel, A., Pennemann, K.-H.: Correctness of high-level transformation systems relative to nested conditions. *Mathematical Structures in Computer Science* 19(2), 245–296 (2009)
10. Habel, A., Radke, H.: Expressiveness of graph conditions with variables. *ECEASST* 30 (2010)
11. Han, D., Pal, S., Liu, Y., Yan, H.: Folding and cutting DNA into reconfigurable topological nanostructures. *Nature Nanotechnology* 5, 712–717 (2010)
12. Hastings, J., Magka, D., Batchelor, C., Duan, L., Stevens, R., Ennis, M., Steinbeck, C.: Structure-based classification and ontology in chemistry. *Journal of Cheminformatics* 4(1), 8 (2012)
13. Konyk, M., De Leon, A., Dumontier, M.: Chemical Knowledge for the Semantic Web. In: Bairoch, A., Cohen-Boulakia, S., Froidevaux, C. (eds.) DILS 2008. LNCS (LNBI), vol. 5109, pp. 169–176. Springer, Heidelberg (2008)
14. Magka, D., Motik, B., Horrocks, I.: Modelling structured domains using description graphs and logic programming, Technical report, Dept. of Computer Science, U. of Oxford (2011)
15. Motik, B., Cuenca Grau, B., Horrocks, I., Sattler, U.: Representing Ontologies Using Description Logics, Description Graphs, and Rules. *Artificial Intelligence* 173(14), 1275–1309 (2009)
16. Motik, B., Sattler, U., Studer, R.: Query Answering for OWL-DL with Rules. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 3(1), 41–60 (2005)
17. Rzepa, H.: Molecular möbius strips and trefoil knots (2003), <http://www.ch.ic.ac.uk/motm/trefoil/> (last accessed December 2011)
18. Trinajstić, N.: Chemical graph theory. CRC Press, Florida (1992)
19. Tseng, H.-R., Vignon, S.A., Stoddart, J.F.: Toward chemically controlled nanoscale molecular machinery. *Angewandte Chemie International Edition* 42, 1491–1495 (2003)
20. Weisstein, E.W.: Polyhedral graph (2011), <http://mathworld.wolfram.com/PolyhedralGraph.html>

# Personalizing and Improving Tag-Based Search in Folksonomies

Samia Beldjoudi<sup>1</sup>, Hassina Seridi-Bouchelaghem<sup>1</sup>, and Catherine Faron-Zucker<sup>2</sup>

<sup>1</sup> Laboratory of Electronic Document Management LabGED,

Badji Mokhtar University Annaba, Algeria

{beldjoudi, seridi}@labged.net

<sup>2</sup> I3S, Université Nice - Sophia Antipolis, CNRS 930 route des Colles, BP 145,

06930 Sophia Antipolis Cedex, France

catherine.faron-zucker@unice.fr

**Abstract.** Recently, the approaches that combine semantic web ontologies and web 2.0 technologies have constituted a significant research field. We present in this paper an original approach concerning a technology that has recognized a great popularity in these recent years, we talk about folksonomies. Our aim in this contribution is propose new technique for the Social Semantic Web technologies in order to see how we can overcome the problem of tags' ambiguity automatically in folksonomies even when we choose representing these latter with ontologies. We'll also illustrate how we can enrich any folksonomy by a set of pertinent data to improve and facilitate the resources' retrieval in these systems; all this with tackling another problem, we speak about spelling variations.

**Keywords:** Folksonomies, Web 2.0, Semantic Web, Tags Ambiguity, Spelling Variations.

## 1 Introduction

Among the powerful technologies of Web 2.0, we find folksonomies, this term has recently appeared on the net to describe a system of classification derived from the practice and method of collaboratively creating and managing tags to annotate and categorize content. Ontologies which constitute the backbone of semantic web contribute significantly in solving the problems of semantics during the definition and the search of information. However even with the strong points of folksonomies and ontologies; their combination together still suffers from some problems. As examples we can cite the problem of tags' ambiguity and spelling variations (or Synonymy) in folksonomies. Our goal in this contribution is to show how we can exploit the power of social interactions between the folksonomy's members in order to extract the meaning of terms and overcome the problems of tags' ambiguity and spelling variations. Also we will try to show how we can use the principle of rules-based systems with ontologies for helping our system to enhance automatically the folksonomy by relevant facts can increase the data available within our system with

relevant information for facilitating the resources retrieval and optimizing the time expended during this process. Our paper is organized as follows: Section 2 presents a quick overview about the main contributions attached to our search field; in Section 3 we will detail the design of our approach. After in Section 4 we move to the experimental phase in order to measure the performance of our approach and discuss the obtained results. Conclusion and future works are discussed in Section 5.

## 2 Related Work

In this section, we will put the point on the famous works which try to reduce the tags' ambiguity problem and especially those aimed to extract the semantic links between folksonomy's terms using ontologies. Mika [7] proposed to extend the traditional bipartite model of ontologies to a tripartite one: that of folksonomies. In another work, Gruber [5] argued that there is no contrast between ontologies and folksonomies, and therefore recommended to build an "ontology of folksonomy". According to Gruber, the problem of the lack of semantic links between terms in folksonomies can be easily resolved by representing folksonomies by ontologies. Specia and Motta [9] in their turn have preferred the use of ontologies to extract the semantics of tags. Their approach consists in building tags clusters, and then trying to identify the possible relationships between tags in each cluster. The niceTag project of Limpens et al. [6] is focused on this same principle: the use of ontologies to extract semantic links existing between tags in a system. In addition, this project has introduced the idea of exploiting interactions between users and the system. Pan et al. [8] aimed at reducing the problem of ambiguity in tagging. They proposed to extend the search of tags in a folksonomy by using ontologies. They defended this principle of extension of the search in order to avoid bothering the users with the rigidity of ontologies. Beldjoudi et al. [1] proposed a technique specially designed to show the social interactions' usefulness in folksonomies for reducing tags' ambiguity problem. In another contribution the one of Beldjoudi el al. [2], the authors propose a method to analyze user profiles according to their tags in order to personalize the recommendation of resources. To sum up, most of the works relative to folksonomies aim to bring together ontologies and folksonomies as a solution to the tags' ambiguity problem and that of the lack of semantic links between tags. In this context, we started our trial to improve a little this technology and give a new view concerning the combination between folksonomies and ontologies.

## 3 Semantic Social Folksonomy with Ontology (SSFO)

Our aim in this contribution is to introduce both the semantics and the social aspects in folksonomies in order to let any user in the system retrieving relevant web resources close to his preferences. In this paper, we aim to show how we can produce a technique for helping any ontology already used for representing a folksonomy to overcome the problem of tags' ambiguity automatically without the need of an expert who must control and organize links between terms. In addition we want show how

we can enrich our folksonomy (without human intervention) with relevant data in order to help optimizing the time of search and enormously reduce the problems of spelling variations and the lack of semantics within folksonomies focusing on the rules-based systems.

### 3.1 Formal Description

Formally, a folksonomy is a tuple  $F = \langle U, T, R, A \rangle$  where  $U$ ,  $T$  and  $R$  represent respectively the set of users, tags and resources, and  $A$  represents the relationship between the three preceding elements i.e.  $A \subseteq U \times T \times R$ . Because this approach is intended to present a technique that can help any folksonomy represented by an ontology to overcome the problems of tags' ambiguity and spelling variations based on the preferences and the interests of each user, and also enrich automatically the system by new relevant data, we suggest here to represent our folksonomy with a simple ontology defined by primitives relations such as "tagged by" and "used by" ... etc.

### 3.2 Resolving Tags' Ambiguity in Folksonomies

Our technique to overcome the problem of tags' ambiguity is not based on ontologies. The idea is to study the profile of each member in the system and then compare the preferences of this one with other users in order to extract those who are similar to him.

It should be noted that: To make the system flexible, we propose to make it interact with the user to accept or reject the retrieved resources. And to avoid the "cold start" problem which is generally occur from a lack of the required data by the system in order to make an excellent recommendation; it's proposed to measure the similarity between resources when the users are not similar. So we can summarize our methodology as follow:

**Similarities between Users.** To calculate this similarity we suggest to use a measure that allows representing each user by a vector  $v_i$  designates a series of binary numbers defined the set of his tags or his resources. Thus, to calculate the similarity between two users, for example  $U_1$  and  $U_2$ , this measure proposes to calculate the cosines of the angle between their associated vectors  $v_1$  and  $v_2$  as shown in the formula (1):

$$\cos(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\|^2 \|v_2\|^2} \quad (1)$$

**Similarities between Resources.** When the users are not similar we suggest measuring the degree of similarity between resources in order to avoid "cold start" problem which is generally resulted from a lack of the data required by the system in order to make an excellent recommendation.

**Recommendation Levels.** We propose here assigning to each resource recommended by the system a factor that indicates the percentage of its recommendation. To achieve this classification, we propose to calculate the ratio between the number of resources

used by the user himself (i.e. the one who does the search) and the number of the resources shared between him and the other users. Above a threshold fixed in [0..1], we qualify the resource as *highly recommended*; under this threshold, it is simply *recommended* or *weakly recommended* if the similarity is close to zero.

### 3.3 Rules-Based Systems in Folksonomies

The purpose of using rules-based systems can be summarized as follow: 1) Avoid the existence of an expert who must control and organize links between terms. This let us say that our technique is dynamic and automatic. 2) Optimize and reduce the time required for searching relevant resources for each user by avoiding the recalculation of similarities every time. And 3) enrich the folksonomy by a relevant fact which can help improving the process of search and reducing the problem of spelling variations. In our approach the folksonomies' enrichment is realized by two categories of data as follows:

1. Enrich our fact base by facts extracted from the similarities' calculations that have been made during the step 3.2; and which say that: such resource is similar to such resource. For example; if we have already found that a resource R1 is similar to another resource R2, then we can add the following fact: is-similar-to (R1, R2) which express that "*R1 is similar to R2*". With this method our system does not recalculate the similarity between the users every time when an actor want to search relevant resources, but it will optimize this time and also the memory space that can be lost in each calculation because with this process; before our system begin the calculation of similarity between users or between resources it will firstly see in the fact base if there are resources similar to those already proposed to this user.

2. The second kind of facts has the following form: "A resource R<sub>Z</sub> can have as tags the tag T<sub>Y</sub>" or can-tagged-by (R<sub>Z</sub>, T<sub>Y</sub>). The advantage of such fact is twofold: a) Reduce the problem of tags' ambiguity (because the similarity between resources became more highly). b) Reduce the problem of spelling variations. We can explain this second point (b) by the following example: "cat" and "chat" means both the same concept (animal) in English and in French, but when a user searches resources annotated by the tag "cat", the system will not offer him those tagged by the word "chat" because it can't understand that the tag "cat" is equivalent to the tag "chat". In others words, supposing that the user U<sub>X</sub> tagged a resource R<sub>1</sub> by the tag *cat* and U<sub>w</sub> is the user who tagged the resource R<sub>2</sub> by the tag *chat*. Noting that; the two resources R<sub>1</sub> and R<sub>2</sub> are already considered as similar according to the similarities' calculations that have been made before. Now if the user U<sub>X</sub> wants search resources concerning the animal "cat" by the tag *cat*, the resource R<sub>2</sub> will not be given to him. In order to overcome this problem our approach proposes to add the following facts: can-tagged-by (R<sub>1</sub>, chat), can-tagged-by (R<sub>2</sub>, cat). And now any user can benefit from the resources of the other and so we have overcame the problem of spelling variation in folksonomies.

Finally, the relationship between our solution and the above problems can be appeared behind the choice of the rules' language RIF (Rule Interchange Format), which became recently a W3C Recommendation. The choice of this language is motivated by the fact that it can support the import of RDF data and RDFS/OWL ontologies [3]. Also a mapping to RIF from ontologies and the vice versa is possible, and thus we can easily treat our dataset and enrich the folksonomy. Furthermore the strength of this language is appear from the fact that it can support many kinds of dialects; among them we find the RIF-PRD (the Production Rule Dialect of the W3C Rule Interchange Format) [4] which allows adding, deleting and modifying facts in the fact base. And so can modify, assert and also retract a set of facts in our data base according to our needs.

## 4 Experimentation

### 4.1 Dataset and Data Treatment

In order to validate our approach, we have conducted an experiment with del.icio.us database. Our test base comprises 1605 tag assignments involving 55 users, 526 tags some of which are ambiguous or have many spelling variations, 950 resources each having possibly several tags and several users. To demonstrate the validity of our approach, we have distinguished two classes of users: the first one contains the users who have employed ambiguous tags and the other one those who did not use those tags. This ambiguity of tags has been subjectively decided: for instance *apple* is ambiguous and *software* is not.

First of all, we have constituted a simple ontology from this dataset in order to represent the folksonomy by ontology. It should be noted that we have used a simple properties for describing this ontology in order to avoid losing the meaning and the objective of our approach, where we have suggested representing our folksonomy by a simple ontology defined by primitives relations such as "tagged by" and "used by"...etc. After that we have used a tool for social network analysis called "Pajek<sup>1</sup>", in order to extract the three networks 'Users-Tags', 'Users-Resources' and 'Tags-Resources'. The results of this step are used in our methodology to calculate the similarities between users and between resources in order to detect the pertinent resources for each user. Now, once we have extracted the three social networks and calculate the different similarities, we have choosen to represent these data with the language RIF because it allow us representing and manipulating our data easily since it can manage RDF data and RDFS /OWL ontologies.

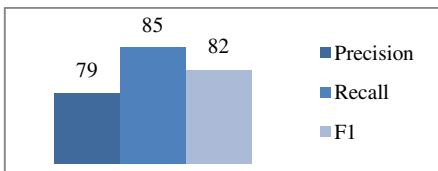
### 4.2 Results

Three metrics are used for evaluating our approach: Precision: It measures the system's ability to reject all not relevant resources to a query. It is given by the ratio of all relevant selected resources and the set of all selected resources. Recall: It measures

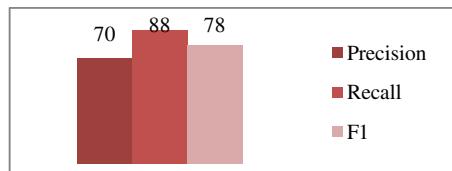
---

<sup>1</sup> It's an analytical tool of social networks, used in [7].

the ability of the system to retrieve all relevant resources to a query. It is given by the ratio of relevant retrieved resources and all relevant resources in the database. And the metric F1: Which is a combination of the two previous metrics and is defined by the formula (2):  $F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$  (2). The three metrics listed above are calculated for each user, and then the average of each metric in the system is calculated. The results are shown in Figures 1 and 2.



**Fig. 1.** The average of the three metrics concerning the problem of tags' ambiguity



**Fig. 2.** The average of the three metrics concerning the problem of spelling variations

#### 4.3 Discussion

The approach presented in this work has tried to extract the semantics in folksonomies in order to allow users capturing the social dimension of their tagging activity. Indeed the obtained results show that the technique SSFO succeeded in distinguishing between ambiguous tags and also them which have many spelling variations. Comparing now our approach with other ones trying to discuss the problem of tags' ambiguity; for example the Pan's and al work [8], we can conclude that our results are very optimistic especially when we know that the proposed approach is flexible i.e. the result of the search's procedure can be changed according to the profile and the interests of each user in contrary to the other approach. In addition the work presented in [8] doesn't tackle the problem of spelling variations. In comparison with [5], we find that our approach doesn't need an expert who must control and organize links between terms. Also the expertise of users which was introduced in [6] is characterized by the complexity of its exploitation when we try as much as possible to avoid a cognitive overload, to limit the necessary effort for the formalization of this expertise which is achieved by our approach.

### 5 Conclusion and Future Work

We have proposed in this paper a new technique based on the force of social interactions between the different actors in folksonomies in order to create a consensus among the users and so increase the semantics in these systems. We have tested this approach on a small amount of data and we have obtained good results. In order to expand and improve this work we propose to validate our approach on a larger dataset and also enrich our database by other relevant facts.

## References

1. Beldjoudi, S., Seridi, H., Faron-Zucker, C.: Ambiguity in Tagging and the Community Effect in Researching Relevant Resources in Folksonomies. In: Proc. of ESWC Workshop User Profile Data on the Social Semantic Web (2011)
2. Beldjoudi, S., Seridi, H., Faron-Zucker, C.: Improving Tag-based Resource Recommendation with Association Rules on Folksonomies. In: Proc. of ISWC Workshop on Semantic Personalized Information Management: Retrieval and Recommendation (2011)
3. de Bruijn, J. (ed.): RIF RDF and OWL Compatibility. W3C Recommendation (June 22, 2010), <http://www.w3.org/TR/2010/REC-rif-rdf-owl-20100622/>
4. de Sainte Marie, C., Hallmark, G., Paschke, A. (eds.): RIF Production Rule Dialect. W3C Recommendation (June 22, 2010), <http://www.w3.org/TR/2010/REC-rif-prd-20100622/>
5. Gruber, T.: Tag Ontology-a way to agree on the semantics of tagging data (2005)
6. Limpens, F., Gandon, F., Buffa, M.: Sémantique des folksonomies: structuration collaborative et assistée, IC (2009)
7. Mika, P.: Ontologies Are Us: A Unified Model of Social Networks and Semantics. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 522–536. Springer, Heidelberg (2005)
8. Pan, J., Taylor, S., Thomas, E.: Reducing Ambiguity in Tagging Systems with Folksonomy Search Expansion. In: Proc. of the 17th International World Wide Web Conference (2009)
9. Specia, L., Motta, E.: Integrating Folksonomies with the Semantic Web. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 624–639. Springer, Heidelberg (2007)

# Views and Synthesis of Cognitive Maps

Aymeric Le Dorze<sup>1</sup>, Lionel Chauvin<sup>2</sup>, Laurent Garcia<sup>1</sup>, David Genest<sup>1</sup>,  
and Stéphane Loiseau<sup>1</sup>

<sup>1</sup> LERIA - Univ. of Angers - 2, Bd Lavoisier 49045 Angers, France  
`{ledorze,garcia,genest,loiseau}@info.univ-angers.fr`

<sup>2</sup> Polytech Nantes - Univ. of Nantes - 2 rue de la Houssinière - 44322 Nantes, France  
`Lionel.Chauvin@univ-nantes.fr`

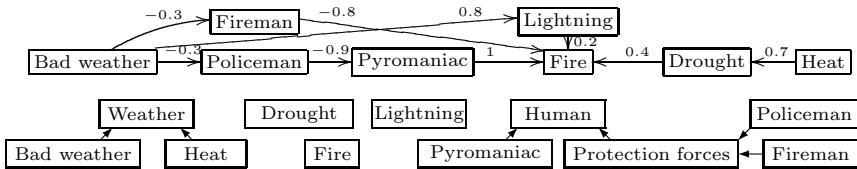
**Abstract.** A cognitive map represent concepts linked by influences. This paper introduces a taxonomic cognitive map model that allows to define a cognitive map on a taxonomy. This taxonomy is used to provide simplified views of a map to help a user to understand it. The model we present also provides a mechanism to help a group of people to build a map. Our proposal consists of a method for building a synthesized map from individual cognitive maps provided by each participant, called designer, by exploiting the taxonomy and preferences over these designers.

**Keywords:** Cognitive map, Influence, Taxonomy, View, Fusion.

## 1 Introduction

*Cognitive maps* [1] represent a set of concepts linked by influences. A concept is a semantic text and an influence is associated to a value; most common *sets of values* are  $\{+, -\}$  [2], the interval  $[-1; 1]$  [3] or  $\{\text{none}; \text{some}; \text{much}; \text{a lot}\}$  [4]. Many cognitive map systems associate a causal algebra which allows to compute the *propagated influence* from a concept to another, according to the *paths* between them. They are used in many fields: biology, sociology, politics... Cognitive maps have the drawback to be not easy to exploit due to the important number of concepts that makes them difficult to construct and apprehend. Indeed, it is both difficult to have a synthetic view of what the map is compound of and to find in the map the concepts the user is interested in.

The first contribution of this paper is to provide views of a cognitive map adapted to the user. A model called *taxonomic cognitive map* (called ontological cognitive map in [5]) is defined by associating a cognitive map with a taxonomy that organizes the concepts. The taxonomy can be seen as a *symbolic scale* to see views of the cognitive map. In this scale, a *degree scale* is a subset of concepts of the taxonomy used to provide a view of the map adapted to the user. This *view* is a cognitive map constructed on the only concepts of the degree scale. Two characteristics must be underlined: first, a view of a cognitive map has fewer concepts thus is more concise; secondly, each concept of the map or a concept generalizing it is in the view, so the view is a generalization of the cognitive map.



**Fig. 1.** A cognitive map ( $M_1$ ) and a taxonomy ( $O$ )

The second contribution of this article is about the collaborative building of a cognitive map. This paper proposes a new approach to collaborate by merging the maps: a single map, called *synthesized map*, merges maps individually produced by *designers*. There are several interests to merge maps: first, the synthesized map can be seen as an agreement between designers and the building of this map may help the process of reaching an agreement. Second, the synthesized map may be used to present a readable aggregation of information.

The second section of this paper describes the taxonomic cognitive map model and its basic inference mechanisms. The third section presents a degree scale and views. The fourth section presents how to compute a synthesized map.

## 2 Taxonomic Cognitive Map

The taxonomic cognitive map model associates a taxonomy with a cognitive map. A concept is defined by its name expressed in a natural language, and a cognitive map is a graph defined on a set of concepts and a set of values. The nodes represent concepts, and they are linked by arcs that represent influences.

**Definition 1 (Cognitive map).** Let  $C$  be a set of concepts. Let  $I$  be a set of values. A cognitive map (CM) defined on  $C$  and  $I$  is a directed labelled graph  $(V, \text{lab}_C, A, \text{lab}_I)$  where:  $V$  is a set of nodes;  $\text{lab}_C : V \rightarrow C$  is a bijective function labelling a node of  $V$  with a concept of  $C$ ;  $A \subseteq V \times V$  is a set of arcs, called influences;  $\text{lab}_I : A \rightarrow I$  is a function labelling each influence with a value of  $I$ .

$M_1$  (fig. 1) is a cognitive map defined on  $[-1,1]$  where 1 (resp. -1) is the total influence (resp. total opposite influence). If we consider the concepts *Drought* and *Fire*, a drought influences the risks of fire at 0.4. This CM also represents the fact that the presence of policemen decreases the presence of pyromaniacs (-0.9).

A taxonomy  $O = (C, \leq)$  is a set of trees of concepts  $C$  ordered by a relation of specialization  $\leq$ . For a subset  $C'$  of concepts of  $O$ , the set of *minimum concepts* are the concepts for which there are no lesser concepts:  $\text{min}(C') = \{c \in C' | \forall c' \in C', c \leq c'\}$ . The concepts of  $\text{min}(C)$  are called the *elementary concepts* of  $C$ .

**Definition 2 (Taxonomic cognitive map (TCM)).** A TCM defined on  $(C, \leq)$  and a set of values  $I$  is a cognitive map defined on  $\text{min}(C)$  and  $I$ .

Since the taxonomy is built for a given CM in order to provide operations, it has to contain concepts of the map, and nothing else. More precisely, concepts

that are incomparable to concepts of the map and specializations of concepts of the map are useless. For example,  $M_1$  is a TCM defined on  $O$  and  $[-1,1]$ .

Propagated influence of a concept on another is computed according to the influence paths between the concepts in the CM. In the following, we consider  $M_1 = (V, \text{lab}_C, A, \text{lab}_I)$  as a TCM defined on a taxonomy  $O = (C, \leq)$  and a set of values  $[-1,1]$ . Let  $c_1, c_2$  be two concepts of  $C$ . An *influence path* from  $c_1$  to  $c_2$  is a sequence (of length  $k$ ) of influences  $(u_i, v_i) \in A$  with  $u_1 = \text{lab}_C^{-1}(c_1)$ ,  $v_k = \text{lab}_C^{-1}(c_2)$  and  $\forall i \in [1, k-1], v_i = u_{i+1}$ . An influence path  $P$  from the concept  $c_1$  to  $c_2$  is *minimal* iff there exists no influence path  $P'$  from  $c_1$  to  $c_2$  such that  $P'$  is a subsequence of  $P$ . The *propagated influence for an influence path*  $P$  is the product of all values labelling the influences of this path. The *propagated influence* from  $c_1$  to  $c_2$ , denoted by  $\mathcal{I}(c_1, c_2)$  is the average of the propagated influences for every minimal path between  $c_1$  and  $c_2$ . In  $M_1$ , there are three minimal influence paths between *Bad weather* and *Fire*. So,  $\mathcal{I}(\text{Bad weather}, \text{Fire}) = ((-0.3 \times -0.8) + (0.8 \times 0.2) + (-0.3 \times -0.9 \times 1)) / 3 \approx 0.22$ .

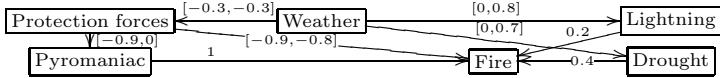
Taxonomic influence enables the user to interrogate a TCM so as to determine the influence between any pair of concepts of the taxonomy. To do that, the two subsets of elementary concepts which specialize the two concepts are computed. We name the set of *elementary concepts for a concept*  $c \in C$  the subset of  $C$  defined as:  $\text{elem}(c) = \{c' \in \min(C) | c' \leq c\}$ . The *taxonomic influence between*  $c_1$  and  $c_2$  is the interval between the minimum and the maximum of the propagated influences between the elementary concepts for  $c_1$  and the ones for  $c_2$ . For example,  $\text{elem}(\text{Fire}) = \{\text{Fire}\}$ ;  $\text{elem}(\text{Protection forces}) = \{\text{Policeman}, \text{Fireman}\}$ . The taxonomic influence between *Protection forces* and *Fire* is  $\mathcal{I}_T$  (*Protection forces, Fire*) =  $[\mathcal{I}(\text{Policeman}, \text{Fire}), \mathcal{I}(\text{Fireman}, \text{Fire})] = [-0.9, -0.8]$ .

### 3 View of a Map

A cognitive map is not always easy to understand, because a map may contain irrelevant information for a given use. Views are thus defined to provide simplified and adapted maps. The taxonomy can be seen as a symbolic scale in which a user chooses a degree scale in order to obtain a view of the TCM. A degree scale is a subset of concepts of the taxonomy and respects some particular properties. Two concepts  $c$  and  $c'$  of  $C$  are said *comparable*, denoted by  $c_1 \perp c_2$ , iff  $c \leq c'$  or  $c' \leq c$ . They are said *incomparable*, denoted by  $c_1 \parallel c_2$ , otherwise. Let  $C' \subseteq C$ .  $C'$  is a *set of incomparable concepts* iff  $\forall c, c' \in C', c \neq c' \Rightarrow c \parallel c'$ . The *set of elementary concepts for*  $C'$  is defined as  $\text{elemSet}(C') = \bigcup_{c \in C'} \text{elem}(c)$ . Let  $C_1, C_2 \subseteq C$ .  $C_1$  is a *representative set of*  $C_2$  iff  $\text{elemSet}(C_2) \subseteq \min(C_1)$ . Note that  $C' \subseteq C$  is a representative set of  $C$  iff  $\text{elemSet}(C') = \min(C)$ .

**Definition 3 (Degree scale).** Let  $C' \subseteq C$ .  $C'$  is a *degree scale* iff  $C'$  is a set of incomparable concepts and is representative of  $C$ .

The use of incomparable concepts avoids redundancies. The use of representative concepts of the taxonomy ensures that all concepts are represented. For example,  $\{\text{Protection forces}, \text{Weather}, \text{Pyromaniac}, \text{Fire}, \text{Lightning}, \text{Drought}\}$  is a degree scale. A view of a TCM is a map in which the concepts are those of a



**Fig. 2.** View of  $M$  for  $\{ \text{Protection forces}, \text{Weather}, \text{Pyromaniac}, \text{Fire}, \text{Lightning}, \text{Drought} \}$

degree scale. Two concepts of a view are connected if there is one elementary concept for each of them so that those elementary concepts are connected in the TCM. An arc between two elementary concepts of the view is labelled in the same way as the corresponding arc of the TCM. In other cases, its label is computed using the taxonomic influence:  $\text{Value}(c_1, c_2) = \text{lab}_I(\text{lab}_{C'}^{-1}(c_1), \text{lab}_{C'}^{-1}(c_2))$  if  $c_1$  and  $c_2$  are elementary; otherwise,  $\text{Value}(c_1, c_2) = \mathcal{I}_T(c_1, c_2)$ .

**Definition 4 (View).** Let  $C'$  be a degree scale. A view for  $C'$  of  $M_1$  is a CM  $(V_s, \text{lab}_{C_s}, A_s, \text{lab}_{I_s})$  defined on  $C'$  and  $I$  such that:  $V_s$  is a set of nodes such that  $\text{card}(V_s) = \text{card}(C')$ .  $\text{lab}_{C_s} : V_s \rightarrow C'$  labels each node of  $V_s$  with a concept of  $C'$ .  $A_s \subseteq V_s \times V_s$  is the set of influences  $(\text{lab}_{C_s}^{-1}(c_1), \text{lab}_{C_s}^{-1}(c_2))$  such that  $c_1$  and  $c_2$  are connected.  $\text{lab}_{I_s} : A_s \rightarrow I \cup \text{Int}(I)$ ;  $(v_1, v_2) \mapsto \text{Value}(\text{lab}_{C_s}(v_1), \text{lab}_{C_s}(v_2))$

Fig. 2 is a view of  $M_1$ . Moreover, other views are possible. For example, a view for an influence may help the user to understand the taxonomic influence between two concepts: the elementary concepts involved in the taxonomic influence are determined and presented in the view in order to see the propagation in detail.

## 4 Synthesized Cognitive Map

To help a group of people to take a decision, a way to proceed is to ask each designer to express his point of view with an individual CM and then to merge these maps in a synthesized map. Our proposal consists of a method for building this map by exploiting a taxonomy describing the concepts and preferences on designers. Each designer creates his own map called *attributed map* using a previously defined taxonomy. This taxonomy is shared between designers and could be more exhaustive than a taxonomy dedicated to a unique map. An attributed map is not a TCM because concepts of this map are not necessarily elementary. An *attributed map* defined on  $O$  and  $I$  is a pair  $(M_1, d)$  where  $d$  is the designer of the CM  $M_1$  defined on  $C' \subseteq C$  and  $I$  such that concepts of  $C'$  are pairwise incomparable. A set  $E$  of attributed maps is said to be *univocal* iff no designer of maps of  $E$  is associated with more than one map. The set of cognitive maps  $E = \{AM_M, AM_F, AM_T, AM_Y\}$  (fig. 3) is a univocal set of attributed maps, defined on the same taxonomy  $O$  and the same set of values  $[-1;1]$ .

Preferences on designers allow to express that the information provided by a given designer has more weight than those provided by another. The *preferences*  $P = (D, \succeq)$  associates a total preorder relation  $\succeq$  to a set of designers  $D$ . Let  $d_1$  and  $d_2$  be two designers.  $d_1 \succ d_2$  means that  $d_1$  is *preferred to*  $d_2$ .  $d_1 \approx d_2$  means that they are *equally preferred*. A total preorder relation can be represented as

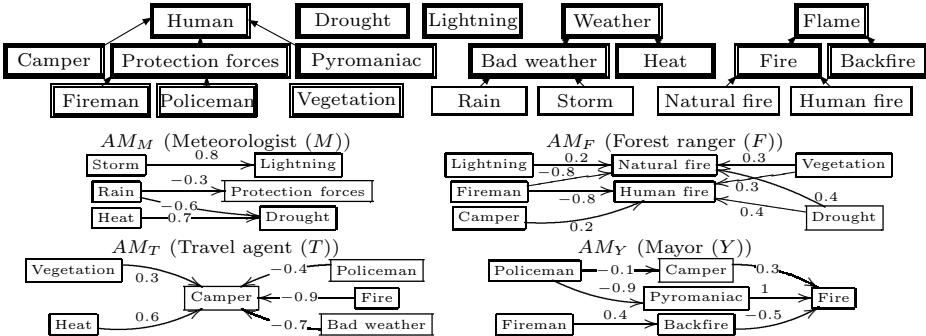


Fig. 3. A taxonomy  $O$  and a univocal set of attributed maps  $E$

a stratification of a set. A stratification  $\mathcal{S}_D$  of  $P$  is a tuple  $\langle \mathcal{S}_1, \dots, \mathcal{S}_m \rangle$  such that  $\forall i \in [1; m] : \mathcal{S}_i \neq \emptyset ; \mathcal{S}_i = \{d \in (D \setminus \bigcup_{k=1}^{i-1} \mathcal{S}_k) / \forall d' \in (D \setminus \bigcup_{k=1}^{i-1} \mathcal{S}_k), d \succeq d'\}$ . Thus, we define the weight of a designer as:  $\mathcal{W}_P(d) = m - i + 1$ . For  $P = Y \succ F \succ T \succ M$ , the stratification is  $\langle \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3 \rangle = \langle \{Y\}, \{F, T\}, \{M\} \rangle$  and  $\mathcal{W}_P(Y) = 3 - 1 + 1 = 3$ .

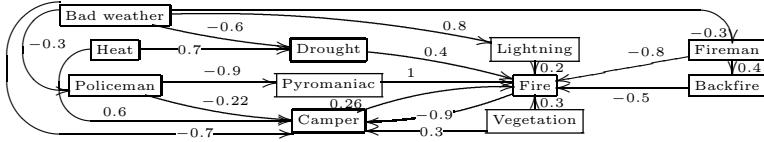
The first step of building the synthesized map is to select concepts in order to obtain a reasonably-sized set of concepts, called synthesized concepts. The second step is to determine the influences, called synthesized influences.

A concept used by a designer is synthesized if there is no comparable concept used by preferred designers.  $\text{usedC}_E(D)$ , the *concepts used by a set of designers*  $D$  are the concepts used by at least one designer of  $D$ . The synthesis of concepts is made in two steps: first, we construct, for each stratum of the preferences, a set of incomparable concepts corresponding to the concepts used by the designers of this stratum; then, we compute the union of these sets. The concepts of each of these sets have to be incomparable with those of the sets corresponding to preferred strata. Formally, the *set of synthesized concepts* of  $E$  is defined as:  $\text{syntC}_E(P) = \bigcup_{i=1}^m \mathcal{CS}(\mathcal{S}_i)$  with:  $\mathcal{CS}(\mathcal{S}_i) = \min(\{c \in \text{usedC}_E(\mathcal{S}_i) / \forall c' \in \bigcup_{k=1}^{i-1} \mathcal{CS}(\mathcal{S}_k), c \parallel c'\})$ . *Fire* is selected because it is used by the favourite designer. *Bad weather* is selected it is used by  $T$  and is incomparable with any concept used by  $Y$ . *Natural Fire* is not selected because *Fire* has already been selected.

The idea of a synthesis of influences is to represent every influence of the set of attributed maps. However, the ending nodes of these original influences are not necessarily in the set of synthesized concepts. Therefore, these influences are modified to be defined between synthesized concepts that are comparable to their original ending nodes.  $\text{syntI}_E(P) = \{(c_1, c_2) / c_1, c_2 \in \text{syntC}_E(P) \wedge \exists((V, \text{lab}_C, A, \text{lab}_I), d) \in E, \exists(v_1, v_2) \in A, \text{lab}_C(v_1) \perp c_1 \wedge \text{lab}_C(v_2) \perp c_2\}$ .

Synthesized concepts and synthesized influences define a synthesized map. Such a map is a first result which highlights influences between concepts without evaluating their strength. This result can be offered as a basis for discussions, but a map that contains values for influences may be automatically built.

The *synthesized values* are the values labelling the influences of the synthesized map. To compute these values, we first build, from two concepts, the set



**Fig. 4.** Valued synthesized map ( $SM_P$ )

of values of influences such that their end nodes are comparable with these two concepts. *Extracted influence values* of two concepts  $c_1, c_2$  of  $C$  are a set of 4-tuples such that :  $\text{extract}_E(c_1, c_2) = \{(l_{\text{abC}}(v_1), l_{\text{abC}}(v_2), d, l_{\text{abI}}((v_1, v_2)) / ((V, l_{\text{abC}}, A, l_{\text{abI}}), d) \in E \wedge (v_1, v_2) \in A \wedge l_{\text{abC}}(v_1) \perp c_1 \wedge l_{\text{abC}}(v_2) \perp c_2\}$ . The extracted influence values between Camper and Fire are  $\{(Camper, Fire, Y, 0.3); (Camper, Human fire, F, 0.2)\}$ . If we define the synthesized value as a weighted average, we obtain  $(0.3 \times 3 + 0.2 \times 2) / (3 + 2) = 0.26$ . A valued synthesized map is a synthesized map where influences are labelled by synthesized values. Fig. 4 shows the valued synthesized map of  $E$  for  $P$ .

Once the synthesized map is built, views may help a user to analyze it. However a synthesized map is not a TCM: its concepts are not necessarily elementary. We build so a *simplified taxonomy* based on the previous taxonomy by removing the concepts that specialize any concept of the synthesized map. Then a synthesized map can be seen as a TCM on this simplified taxonomy and views can be defined. We define it as  $O_s = (\{c \in C / \exists c' \in \text{syntCE}(P), c \leq c'\}, \leq)$ . The double-boxed concepts in fig. 3 is the simplified taxonomy of  $O$  for  $SM_P$ .

## 5 Conclusion

The contribution of this paper is the use of a taxonomy within the cognitive map model to help users to exploit a CM, but also to build it. A degree scale provides a way to navigate in a map using views. The visual simplicity of the taxonomy makes the use of the degree scale coherent with the cognitive map model: the taxonomic cognitive map model is indeed easily understandable. This paper also proposes a method to merge several CMs into a unique map using preferences on their respective designers.

## References

1. Tolman, E.C.: Cognitive maps in rats and men. *The Psychological Review* 55(4), 189–208 (1948)
2. Axelrod, R.M.: Structure of decision: the cognitive maps of political elites, Princeton, N.J. (1976)
3. Kosko, B.: Fuzzy cognitive maps. *Int. J. of Man-Machines Studies* 24, 65–75 (1986)
4. Zhou, S., Zhang, J.Y., Liu, Z.Q.: Quotient fcms – a decomposition theory for fuzzy cognitive maps. *IEEE Transactions on Fuzzy Systems* 11(5), 593–604 (2003)
5. Chauvin, L., Genest, D., Loiseau, S.: Ontological cognitive map. *International Journal on Artificial Intelligence Tools* 18(5), 697–716 (2009)

# Identification of the Compound Subjective Rule Interestingness Measure for Rule-Based Functional Description of Genes

Aleksandra Gruca<sup>1</sup> and Marek Sikora<sup>1,2</sup>

<sup>1</sup> Silesian University of Technology, Gliwice, Poland

<sup>2</sup> Institute of Innovative Technologies EMAG, Katowice, Poland

**Abstract.** Methods for automatic functional description of gene groups are useful tools supporting the interpretation of biological experiments. The RuleGO algorithm provides functional interpretation of gene groups in a form of logical rules including combinations of Gene Ontology terms in their premises. The number of rules generated by the algorithm is usually huge and additional methods of rule quality evaluation and filtration are required in order to select the most interesting ones. In the paper, we apply the multicriteria decision making UTA method to obtain a ranking of rules based on subjective expert opinion which is provided in a form of an ordered list of several rules. The presented approach is applied to the well known data set from microarray experiment and the results are compared with the standard RuleGO compound rule quality measure.

**Keywords:** rule quality, rule interestingness, multicriteria decision making, functional annotations, Gene Ontology, bioinformatics.

## 1 Introduction

Over the past 10 years, high-throughput experimental techniques in molecular biology have become one of the most important tools that help to discover and better understand complexity of biological systems. However, understanding of the biological meaning from the raw outputs of DNA microarray experiments is becoming the bottleneck in the application of sophisticated high-throughput techniques for genomic or proteomic data analysis.

DNA microarray experiment allows measuring expression profiles of a thousands of genes simultaneously during a single biological experiment, generating thousands of data points. The result needs biological interpretation and this aspect of data analysis is often done by an expert in the field of the experimental design, frequently manually, which is time consuming for large datasets. Therefore, there is a strong need for development of methods and algorithms that support the expert work during the process of gene groups interpretation.

In this paper we apply the multicriteria decision making UTA method [1][2] to obtain a ranking of generated rules based on subjective expert opinion which is provided in a form of an ordered list consisting of several selected rules. The rules are generated using RuleGO method [3] and provide functional interpretation of gene groups in a form of Gene Ontology terms combinations.

## 2 Description of Gene Groups Using Decision Rules

### 2.1 Functional Annotations of Genes

Application of annotations for functional description of groups of genes can be very efficient method supporting the process of interpretation of biological experiment results. One of the well-known and good-quality functional annotation database is the Gene Ontology (GO) database [4] which has hierarchical form of directed acyclic graph (DAG) and provides controlled and structured vocabulary that describe genes and their products. Each node is called GO term and a particular GO term can be regarded as a keyword providing functional description of gene or gene product that is annotated by such GO term.

The information included in the GO database is provided on different levels of details. The GO terms that are close to the root describe more general concepts, and as a DAG is traversed down from the root to its leaves, the GO terms are more specific. For each GO term we can also compute its level in the GO graph hierarchy by counting the maximal number of edges along the longest possible path from that term to the root. The annotations derived from the GO database can be used by automated methods or tools to support the process of functional interpretation of genes function.

### 2.2 Induction of Rules for Gene Groups Description

We denote two disjoint sets of genes by  $G_1$  and  $G_2$ . Group  $G_1$  represents analysed set of genes and group  $G_2$  is the reference set. The functional characterisation of gene group  $G_1$  is given by a set of decision rules  $RUL$ . Each rule  $r \in RUL$  include combination of GO terms in its premise and described group of genes  $G_1$  in its conclusion and has the following form:

$$\text{IF } t_1 \wedge t_2 \wedge \dots \wedge t_k \text{ THEN } G_1 \quad . \quad (1)$$

When specified to a particular gene, the rule  $r$  has the following interpretation: *if a gene is described by the GO terms that compose the rule r, then it belongs to the group presented in the rule conclusion*. For each rule we can determine the number of genes belonging to the group  $G_1$  that are described by GO terms from rule premise and denote this number as  $supp(r)$ . We can also determine the number of genes from the set  $G = G_1 \cup G_2$  that are described by GO terms from rule premise and denote this number as  $rec(r)$ .

Several tools providing methods of induction of rules that include combinations of GO terms in their premises has been developed [3][5]. The rules are generated and presented to a human expert, and as the number of generated rules is typically huge, the reduction of their amount is required. RuleGO method of rule filtration and evaluation allows selecting the most interesting rules according to user preferences [6]. However, the proposed method does not guarantee that during filtration process some of interesting rules will not be removed. Also, the final ranking provided by the RuleGO method is not always consistent with expert subjective opinion. To address these problems we developed solution based on multicriteria decision making method UTA.

### 3 RuleGO Rule Evaluation and Filtration

The rule evaluation and filtration method implemented in the RuleGO service is based on the compound quality measure  $Q$ . During rule generation process, each rule is evaluated with three different partial quality measures:  $normLength(r)$  - normalized number of GO terms appearing in rule premise,  $depth(r)$  - normalized sum of the levels of GO terms appearing in rule premise,  $modY(r)$  - modified YAILS measure [7]. The measure  $Q$  is a product of the three above components:

$$Q(r) = normLength(r) * depth(r) * modY(r) . \quad (2)$$

The compound measure  $Q$  is used to create rule ranking on which the filtration method is based. Detailed description of the algorithm and its comparison with other methods can be found in article [3] and [6].

Description of the components of the compound quality measure are given below, together with other criteria that are partial quality measures used to identify the utility function. Proposed indices allow evaluating different features of obtained rules. Some of them consider objective criteria such as rule precision and coverage and other are domain-specific.

First criterion is represented by modified YAILS measure [2] which is a weighted measure that evaluates rule precision and coverage:

$$modY(r) = (0.5 + 0.25prec(r))prec(r) + (0.5 - 0.25cov(r))cov(r) , \quad (3)$$

where  $prec(r) = supp(r)/rec(r)$ ,  $cov(r) = supp(r)/N$  and  $N$  denotes number of genes in the group described by the rule.

Another measure that evaluates simultaneously both the rule precision and coverage is RSS measure [4] [8][9] which considers example distribution among the gene group indicated by the rule and the other genes:

$$RSS(r) = \frac{supp(r)}{N} - \frac{rec(r) - supp(r)}{M - N} , \quad (4)$$

where  $N$  is the number of genes in the group described by the rule and  $M$  is the number of genes in the whole analysed data set. The measure proposes the method of rule evaluation analogous to the method of classifier sensitivity and specificity evaluation [8][10][11].

Another criterion can be regarded as a subjective, as it evaluates specificity of information included in analysed rule. From the description point of view we are interested in rules that describe specific interactions among genes, thus we prefer rules including GO terms from as low level of GO graph as possible (5):

$$depth(r) = \frac{\sum_{i=1}^{NOGOTerms(r)} level(t_i)}{\sum_{i=1}^{NOGOTerms(r)} max\_path(t_i)} , \quad (5)$$

where  $level(t_i)$  is a level of a GO term  $t_i$  that occurs in the rule premise and  $max\_path(t_i)$  is the longest path leading from the root to a leaf of GO graph that passes through the node  $t_i$ .

Another component of the utility function is represented by a rule statistical significance. A hypergeometric test is used to verify the following null hypothesis: *assignment of genes recognized by the rule to the group described by the rule is equivalent to a random assignment of the genes to that group*. The p-value of the right-side hypergeometric test is computed as follows (6):

$$p\text{-value}(r_{(x,K,N,M)}) = \sum_{x_i \geq x} p(r_{x_i,K,N,M}) , \quad (6)$$

where  $p(r_{x,K,N,M})$  denotes probability based on hypergeometric test,  $x = supp(r)$ ,  $K = rec(r)$ ,  $N$  is the number of genes in the group described by the rule and  $M$  is the number of genes in the whole analysed data set.

Next component measure prefers rules that include GO terms that differs the most from the other GO terms composing analysed rule set. The  $avgSim(r)$  measure allows introducing into the output rule set the diverse biological knowledge. The similarity of two rules  $r_i$  and  $r_j$  is computed using the following formula:

$$Sim(r_i, r_j) = 1 - \frac{\#GOterms(r_i, r_j) + \#GOterms(r_j, r_i)}{\#GOterms(r_i) + \#GOterms(r_j)}, \quad (7)$$

where:  $\#GOterms(r_i, r_j)$  is a number of unique GO-terms occurring in the rule  $r_i$  and not occurring in the rule  $r_j$ . The GO-term  $t$  from the rule  $r_i$  is recognized as the unique if it does not occur directly in the rule  $r_j$  and there is no path in GO graph that includes both term  $t$  and any term  $k$  from rule  $r_j$  premise;  $\#GOterms(r)$  is the number of GO-terms in the rule  $r$  premise. The measure  $avgSim(r)$  is based on the above equation (7) and it is computed as the mean similarity of the rule  $r$  to the other rules from result set.

The last criterion  $length(r)$  is simply the number of GO terms that compose the rule  $r$  premise. We assume that the bigger number of terms is in the rule premise, the more information is represented by a rule.

## 4 Multicriteria Rules Evaluation Based on Utility Function

In this paper we propose a multicriteria evaluation of rules using so-called utility function. In this method we use a set of criteria  $K = \{q_1, q_2, \dots, q_l\}$  and the order of user-defined rules. The criteria may be rule interestingness (quality) measures or any other index useful for the user.

To identify additive compound measure the UTA method will be used [1][2]. In this approach a compound rule evaluation measure is estimated. This measure aggregates the values of all component measures  $q_i, i \in \{1, 2, \dots, l\}$  for a single global value assessment. The basis for estimating the utility function is a set of rules ordered according to the subjective user preferences. This set is a subset of the one we want to order. Evaluation performed by the user need to be associated with the rule assessment done by component measures. It is assumed that the assessment satisfies the conditions of monotonicity.

The purpose of estimating the utility function is to restore with its use the rule order defined by an expert. The resulting function can then be used to order the remaining rules. For simplicity, we assume that a set of criteria  $K$  will be identified with a set of rule evaluation measures. We also suppose that all measures are gain-type and satisfy the condition (8):

$$\begin{cases} q_i(r_1) > q_i(r_2) \Leftrightarrow r_1 \succ r_2 \\ q_i(r_1) = q_i(r_2) \Leftrightarrow r_1 \cong r_2 \end{cases} . \quad (8)$$

The notation  $r_1 \succ r_2$  means that the rule  $r_1$  is better than the rule  $r_2$ . The notation  $r_1 \cong r_2$  means that the rule  $r_1$  is no better or worse than  $r_2$ . The range of values for each measure  $q_i$  is defined as  $[vq_{i*}, vq_i^*]$ . Within a set of rules the reference set is selected  $RUL_R \subset RUL$  and arranged by the user. The resulting arrangement is called the reference order.

With each rule  $r \in RUL$ , the measure values vector  $\mathbf{q}^r = (vq_1^r, vq_2^r, \dots, vq_l^r)$  can be associated, where  $vq_i^r = q_i(r)$ ,  $i \in \{1, 2, \dots, l\}$ . The UTA method is searching for additive utility function  $u(q^r)$  composed of normalized partial utility functions  $u_i$ :

$$u(\mathbf{q}^r) = \sum_{i=1}^l u_i(vq_i^r) , \quad (9)$$

$$\begin{cases} \sum_{i=1}^l u_i(vq_i^*) = 1 \\ u_i(vq_*) = 0 \quad \forall i \in \{1, 2, \dots, l\} \end{cases} . \quad (10)$$

Global and partial utility functions satisfy the monotonicity condition (11):

$$\begin{cases} u(\mathbf{q}^{r1}) > u(\mathbf{q}^{r2}) \Leftrightarrow r_1 \succ r_2 \\ u(\mathbf{q}^{r1}) = u(\mathbf{q}^{r2}) \Leftrightarrow r_1 \cong r_2 \end{cases} . \quad (11)$$

Considering (9) and (10), and the monotonicity constraints (11), the global assessment of each rule from the rule set  $RUL_R$  can be expressed as (12):

$$u'(\mathbf{q}^r) = \sum_{i=1}^l u_i(vq_i^r) + \sigma(r) , \quad (12)$$

where  $\sigma(r)$  is a potentially committed error in terms of real value  $u(\mathbf{q}^r)$ .

The UTA method requires that the partial utility functions are piecewise linear. To obtain linearity of each  $q_i$  function, the range of values for each measure  $q_i$  is divided into  $t_i - 1$  equal intervals (values  $t_i$  are set by the user). After linearization, the value of the partial utility function  $u_i$  for each  $q_i(r) = vq_i^r \in [vq_i^j, vq_i^{j+1}]$ ,  $\forall j \in \{1, 2, \dots, t_i\}$  is calculated according to (13):

$$u_i(q_i(r)) = u_i(vq_i^r) = u_i(vq_i^j) + \frac{vq_i^r - vq_i^j}{vq_i^{j+1} - vq_i^j} [u_i(vq_i^{j+1}) - u_i(vq_i^j)] . \quad (13)$$

We assume that in the set  $RUL_R$  rules are numbered such that  $r_1$  is the best rule, and  $r_m$  ( $m = |RUL_R|$ ) is the worst one. This means that for each arranged pair

$\langle r_k, r_{k+1} \rangle$  it is either satisfied  $r_k \succ r_{k+1}$  or  $r_k \cong r_{k+1}$ . Based on this assumptions, for each pair of rules  $\langle r_k, r_{k+1} \rangle$ , the value  $\Delta(r_k, r_{k+1}) = u'(\mathbf{q}^{\mathbf{r}^k}) - u'(\mathbf{q}^{\mathbf{r}^{(k+1)}})$  satisfies one of the following conditions (14):

$$\begin{cases} \Delta(r_k, r_{k+1}) \geq \delta \Leftrightarrow r_k \succ r_{k+1} \\ \Delta(r_k, r_{k+1}) = 0 \Leftrightarrow r_k \cong r_{k+1} \end{cases} . \quad (14)$$

Partial utility functions also satisfy the condition:  $u_i(vq_i^{j+1}) - u_i(vq_i^j) \geq s_i, \forall j \in \{1, 2, \dots, t_i\}, \forall i \in \{1, 2, \dots, l\}$ , where  $s_i \geq 0$  is threshold of neutrality.

Based on the above conditions, the partial utility functions are estimated by solving the following linear programming problem with constraints (15):

$$\begin{cases} [\min] F = \sum_{r \in RUL_R} \sigma(r) \\ \text{with constrains} \\ \Delta(r_k, r_{k+1}) \geq \delta \Leftrightarrow r_k \succ r_{k+1} \quad \forall k \\ \Delta(r_k, r_{k+1}) = 0 \Leftrightarrow r_k \cong r_{k+1} \quad \forall k \\ u_i(vq_i^{j+1}) - u_i(vq_i^j) \geq 0 \quad \forall i, j \\ \sum_{i=1}^l u_i(vq_i^*) = 1 \\ u_i(vq_{i*}) = 0 \quad u_i(vq_i^j) \geq 0 \quad \sigma(r) \geq 0 \quad \forall r \in RUL_r, \forall i, j \end{cases} . \quad (15)$$

The quality of the obtained solution is verified by comparing the rule order established by an expert in  $RUL_R$  with the order based on the estimated utility function, using  $\tau$  Kendall correlation coefficient. If orders are identical, we may use the utility function to order all available rules. If a solution is not satisfactory, we can try to: change the reference rule set, increase the values of the  $t_i$  parameters, change the value of  $\delta$  parameter.

A detailed description of the UTA method can be found in [1][2].

## 5 Experimental Analysis

For the experimental analysis we used well-known, benchmark data set from the DNA microarray experiment described in [12]. From 10 functional clusters discovered by Eisen in [12] we selected group which was described as a group including genes involved in *glycolysis process*. The 17 genes that composed *glycolysis* group were annotated by GO terms from at least 3rd and at most 10th level in GO graph that described at least 3 genes from our analysed dataset.

The rules were generated by RulGO rule induction algorithm. Maximal number of GO terms in a rule premise was set to five. The unfiltered set of rules included 297,040 rules. Then, all induced rules were evaluated using standard RuleGO compound quality measure and filtered (similarity threshold was set to 0.5). Finally, after filtration process, we obtained set of 42 rules covering all genes from *glycolysis* group.

The obtained set of 42 rules was presented to the expert who randomly selected six rules and ordered them according to his subjective preferences. For each selected rule we computed six component partial quality measures. Computed values of the individual component measures are presented in the Table I which also includes original RuleGO ranking based on compound  $Q$  measure.

**Table 1.** Component measure values for six rules selected by the expert

Expert ranking	RuleGO ranking	modY	RSS	GOLevel	p-value	avgSim	length
1	22	0.99	0.88	0.46	7.13e-23	0.16	3
2	24	0.91	0.41	0.56	9.13e-10	0.04	2
3	1	0.83	0.18	0.97	0.0002	0.15	5
3	3	0.83	0.18	0.87	0.0002	0.14	5
3	20	0.83	0.18	0.64	0.0002	0.09	3
6	37	0.61	0.29	1.00	9.71e-06	0.04	1

Each of the six component measures was then used to estimate the partial utility functions according to the formula (13). Measures *modY*, *RSS*, *GOLevel*, *p-value* and *avgSim* were divided into three linear intervals. For measure *length* we assumed values from the set {1, 2, 3, 4, 5}. For a pair of rules  $\langle r_k, r_{k+1} \rangle$  satisfying the relation  $r_k \succ r_{k+1}$  we assumed threshold  $\delta = 0.05$ .

We compared the UTA rule ranking with Expert's ranking and two rankings generated by RuleGO method, that is, ranking based on compound *Q* measure and ranking based on *p-value*. The results show that the rule ranking generated with the UTA method is consistent with the expert opinion in contrast to the ranking based on compound *Q* measure. As the ranking provided by an expert on six selected rules was consistent with the ranking defined by the *modY* measure, we also analysed the correlation between UTA and *modY* rules. The results of comparison of different rankings are presented in the Table 2.

**Table 2.** Comparison of rule ranking obtained for different rule quality measures

Rankings compared	$\tau$ Kendall correlation coefficient
UTA vs. Expert's ranking	1
UTA vs. Compound <i>Q</i>	0.4472
UTA vs. <i>p-value</i>	0.158
UTA vs. <i>modY</i>	0.2834
Compound <i>Q</i> vs. <i>p-value</i>	-0.0906

Analysis of the results presented in the Table 2 shows that the rule ranking generated with the use of the utility function differs from the RuleGO, *modY* and *p-value* rankings. Selected rules from RuleGO compound *Q* measure ranking and values of their quality indices are provided in the Table 3.

Based on the fact that we were able to restore the ranking given by the expert, we assume that proposed method allows obtaining rule set that is more consistent with the expert preferences than rules generated by standard approach.

**Table 3.** RuleGO and UTA rankings of filtered rules with rules selected by the expert

RuleGO ranking	UTA ranking	supp	rec	modY	RSS	GOLevel	p-value	avgSim	length
<b>1</b>	<b>6</b>	<b>3</b>	<b>3</b>	<b>0.83</b>	<b>0.18</b>	<b>0.97</b>	<b>0.0002</b>	<b>0.15</b>	<b>5</b>
2	14	3	3	0.83	0.18	0.88	0.0002	0.15	5
<b>3</b>	<b>6</b>	<b>3</b>	<b>3</b>	<b>0.83</b>	<b>0.18</b>	<b>0.87</b>	<b>0.0002</b>	<b>0.14</b>	<b>5</b>
4	12	3	3	0.83	0.18	0.81	0.0002	0.14	5
5	11	3	3	0.83	0.18	0.73	0.0002	0.13	5
				...					
18	9	3	3	0.83	0.18	0.52	0.0002	0.07	4
19	24	3	3	0.83	0.18	0.5	0.0002	0.16	4
<b>20</b>	<b>6</b>	<b>3</b>	<b>3</b>	<b>0.83</b>	<b>0.18</b>	<b>0.64</b>	<b>0.0002</b>	<b>0.09</b>	<b>3</b>
21	2	14	14	0.99	0.82	0.49	6.18e-21	0.15	3
<b>22</b>	<b>1</b>	<b>15</b>	<b>15</b>	<b>0.99</b>	<b>0.88</b>	<b>0.46</b>	<b>7.13e-23</b>	<b>0.16</b>	<b>3</b>
23	5	3	3	0.83	0.18	0.77	0.0002	0.04	2
<b>24</b>	<b>4</b>	<b>7</b>	<b>7</b>	<b>0.91</b>	<b>0.41</b>	<b>0.56</b>	<b>9.13e-10</b>	<b>0.04</b>	<b>2</b>
25	20	3	3	0.83	0.18	0.52	0.0002	0.08	2
				...					
31	31	3	3	0.83	0.18	0.89	0.0002	0.03	1
32	40	4	7	0.47	0.22	0.78	0.0003	0.1	2
				...					
36	32	4	6	0.55	0.23	0.57	0.0001	0.02	2
<b>37</b>	<b>42</b>	<b>5</b>	<b>7</b>	<b>0.61</b>	<b>0.29</b>	<b>1.00</b>	<b>9.71e-06</b>	<b>0.04</b>	<b>1</b>
38	28	3	3	0.83	0.18	0.7	0.0002	0.01	1
39	37	3	3	0.83	0.18	0.67	0.0002	0.03	1
40	38	3	3	0.83	0.18	0.63	0.0002	0.03	1
41	3	8	8	0.93	0.47	0.25	3.4e-11	0.04	2
42	39	3	3	0.83	0.18	0.56	0.0002	0.05	1

## 6 Related Work

The problem with a complex, multicriteria rule evaluation due to chosen interestingness measures was investigated in [13]. The article shows that support–precision Pareto-optimal border (i.e. the set of non-dominated rules with respect to support and precision) includes rules optimal according to several rule attractiveness (quality) measures, such as *Laplace*, *lift*, *rule interest* and others.

In the work presented in [14] for multicriteria rule evaluation, support and so called confirmation measure was used [15] as semantically more useful than precision. The results presented in [13] show that if the most precise rule is selected from the set of rules with identical support then this rule also has the highest value of other interestingness measures. These measures are monotonic relative to the support and precision. However results presented in articles [13] and [14] do not allow ordering a rule set.

For multicriteria rule evaluation it is possible to apply machine learning methods [16] as well as multicriteria decision making methods [17]. These methods use information obtained from an expert. Lenca [17] apply the PROMETHEE method [18] to select interestingness measure which is able to order a rule set in a manner most similar to the order provided by an expert.

Another machine learning method is proposed in [16]. In this approach an expert classifies rules to one of three categories: interesting, uninteresting, incomprehensible. For classified rules, the broad spectrum of interestingness measures is calculated. This allows obtaining a set of meta-data where examples (rules) are described by a vector of interestingness measures values and decision attribute that indicates the category to which the rule belongs. The learning algorithm is started on the meta-data set, and the result of learning is the classifier that is used to determine whether new analysed rules are interesting or not. However, the presented method does not allow creating rule ranking.

## 7 Conclusions

The UTA is well known method used to support decision-making processes in multicriteria decision making. As shown in the paper, this method is suitable for estimating a compound rule interestingness measure, which supports expert-guided rule selection process. The number of rules that are generated by the RuleGO method is typically huge, and the compound  $Q$  measure does not always allow ordering the rules as an expert expects.

In application domain we considered rules that are characterized by various indices (interestingness measures). The advantage of the UTA method is the possibility of combining indices of different types (gain- or cost-type) which are expressed in various scales.

The application of the UTA method has proved to be effective, as it allowed restoring of the rule ranking defined by the expert and ordering the rest of the induced rules. This new ranking differs from the orders obtained on basis of the rule statistical significance and on the basis of the compound  $Q$  measure.

In most cases, the subjective evaluation of rule interestingness is based on an expert individual preferences and experience, and it is very difficult to define a single measure satisfying needs of different users performing functional interpretation of gene groups. Thus, the proposed method that is strictly focused on the subjective preferences of the individual user may be for him the most attractive way to evaluate the whole set of rules. The advantage of the method is its simplicity (from the user point of view) and flexibility.

The UTA methodology will be another method of rule evaluation for a functional description of gene groups that will be implemented in the RuleGO service.

**Acknowledgements.** The work was supported by National Science Centre (decision DEC-2011/01/D/ST6/07007).

## References

1. Jacquet-Lagreze, E., Siskos, J.: Assessing a set of additive utility functions for multicriteria decision-making, the UTA method. *European Journal of Operational Research* 10, 151–164 (1982)
2. Siskos, Y., Grigoroudis, E., Matsatsinis, N.: UTA methods. In: *Multiple Criteria Decision Analysis: State of the Art Surveys*, pp. 297–343. Springer (2005)
3. Gruca, A., Sikora, M., Polański, A.: RuleGO: a logical rules-based tool for description of gene groups by means of gene ontology. *Nucleic Acids Res.* 39(suppl. 2), W293–W301 (2011)
4. Ashburner, M., et al.: Gene Ontology: tool for the unification of biology. *Nature Genetics* 25, 25–29 (2000)
5. Nogales-Cadenas, R., et al.: Genecodis: interpreting gene lists through enrichment analysis and integration of diverse biological information. *Nucleic Acids Res.* 37(suppl. 2), W317–W322 (2009)
6. Sikora, M., Gruca, A.: Induction and selection of the most interesting gene ontology based multiattribute rules for descriptions of gene groups. *Pattern Recognition Letters* 32(2), 258–269 (2011)
7. Gruca, A.: Characterization of gene groups using decision rules (in Polish). PhD thesis, Silesian University of Technology, Gliwice, Poland (2009)
8. Fürnkranz, J., Flach, P.: Roc 'n' rule learning - towards a better understanding of covering algorithms. *Machine Learning* 58(1), 39–77 (2005)
9. Sikora, M., Gruca, A.: Quality improvement of rules based gene groups descriptions using information about GO terms importance occurring in premises of determined rules. *Int. J. of Appl. Mathematics and Computer Science* 20(3), 555–570 (2010)
10. Bairagi, R., Suchindran, C.: An estimation of the cutoff point maximizing sum of sensitivity and specificity. *Indian Journal of Statistics* 51, 263–269 (1989)
11. Lavrač, N., Flach, P.A., Zupan, B.: Rule Evaluation Measures: A Unifying View. In: Džeroski, S., Flach, P.A. (eds.) *ILP 1999. LNCS (LNAI)*, vol. 1634, pp. 174–185. Springer, Heidelberg (1999)
12. Eisen, M., et al.: Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* 95(25), 14863–14868 (1998)
13. Bayardo, R.J., Agrawal, R.: Mining the most interesting rules. In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, California, United States, pp. 145–154. ACM (1999)
14. Brzezinska, I., Greco, S., Slowinski, R.: Mining pareto-optimal rules with respect to support and ant-support. *Engineering Applications of Artificial Intelligence* 20(5), 587–600 (2007)
15. Christensen, D.: Measuring confirmation. *Journal of Philosophy XCVI*, 437–461 (1999)
16. Abe, H., Tsumoto, S.: Comparing accuracies of rule evaluation models to determine human criteria on evaluated rule sets. In: *Proc. of the 2008 IEEE International Conference on Data Mining Workshops*, Italy, pp. 1–7. IEEE Computer Society (2008)
17. Lenca, P., et al.: A multicriteria decision aid for interestingness measure selection. Lussi-tr-2004-01-em, LUSSI Department, GET/ENST, Bretagne, France (2004)
18. Brans, J., Mareschal, B.: PROMETHEE Methods. In: *Multiple Criteria Decision Analysis: State of the Art Surveys*, pp. 163–195. Springer, New York (2005)

# Automatic Generation and Learning of Finite-State Controllers

Matteo Leonetti<sup>1</sup>, Luca Iocchi<sup>2</sup>, and Fabio Patrizi<sup>2</sup>

<sup>1</sup> Department of Advanced Robotics, Istituto Italiano di Tecnologia,  
via Morego, 30, 16163 Genova

<sup>2</sup> Department of Computer, Control, and Management Engineering,  
Sapienza University of Rome, via Ariosto 25, 00185 Rome

**Abstract.** We propose a method for generating and learning agent controllers, which combines techniques from automated planning and reinforcement learning. An incomplete description of the domain is first used to generate a non-deterministic automaton able to act (sub-optimally) in the given environment. Such a controller is then refined through experience, by learning choices at non-deterministic points. On the one hand, the incompleteness of the model, which would make a pure-planning approach ineffective, is overcome through learning. On the other hand, the portion of the domain available drives the learning process, that otherwise would be excessively expensive. Our method allows to adapt the behavior of a given planner to the environment, facing the unavoidable discrepancies between the model and the environment. We provide quantitative experiments with a simulator of a mobile robot to assess the performance of the proposed method.

**Keywords:** Agent Programming, Planning, Reinforcement Learning, Controller Induction.

## 1 Introduction

Decision making in domains for which no detailed model description of the underlying system is available is one of the most challenging problems in Artificial Intelligence. This condition often characterizes applications related to cognitive robots acting in a real environment. Two different perspectives have been applied to this problem. On the one hand, the use of automatic planning techniques assume the presence of a suitable model of the domain. Solutions can be found, by planning, only if they are represented or characterized by this model. On the other hand, the use of model-free machine learning techniques prove to be too sample inefficient (in the number of trials) to converge in real environments, even for simple tasks.

The integration of the two perspectives allows to overcome these main difficulties. Planning-based methods provide an effective way to properly reduce the solution space, while learning-based methods allow for exploring such a solution space. The resulting behavior improves from experience, and can take into account non-modeled aspects of the task.

More specifically, in this paper, we consider the following problem: an agent has to optimize its performance over a complex task. A model of the task and the environment is given with limited accuracy (i.e., not all the relevant properties may be considered in the model). The setting is such that: 1) a pure planning method would not be effective, because: i) it may not be able to provide an effective solution since the model is not sufficiently accurate; ii) it returns a strategy with constant average performance (i.e., it will not adapt to the real environment and improve over time); 2) a pure learning method will not be effective, because the number of experiments needed for convergence to either a “good” or optimal solution is not feasible for the system.

In this paper we propose a method that allows for improving the performance over time with limited sample complexity, so to be suitable also for real systems (like robots) acting in a real environment.

Although many different representations can be used for the behavior of an intelligent agent, in this paper we focus on finite state machines, because they can realize compact controllers and have been largely used for this purpose. As already mentioned, however, we consider a scenario in which both automatic generation from a model, and learning from data separately present non-optimal performance due to the complexity of the task. We present a combined method that is able to generate and learn a finite state machine representing the controller of a complex system. It exploits both an approximated and incomplete model of the domain and experience on the field. In particular, a planning-based technique is used to generate a set of possible solutions to the problem given the approximated model, while learning is applied to refine and adapt these solutions to the actual environment.

## 2 Background and Related Work

In this section, we introduce the notation and provide the background, behind planning and reinforcement learning, required in the this paper. We also describe the methods of Hierarchical Reinforcement Learning closer to our own, to clarify the novel aspects.

### 2.1 Planning Problems

A *non-deterministic planning domain* is a tuple  $\mathcal{D} = \langle A, Prop, S, s_0, T \rangle$ , where: (a)  $A$  is the finite set of actions; (b)  $Prop$  is the finite set of propositions; (c)  $S \subseteq 2^{Prop}$  is the finite set of states; (d)  $s_0 \in S$  is the initial state; (e)  $T : S \times A \mapsto 2^S$  is the transition function. Each state  $s$  implicitly defines a propositional interpretation  $\nu_s : Prop \mapsto \{\top, \perp\}$  s.t.  $\nu_s(p) = \top$  iff  $p \in s$ . Given a propositional formula  $\varphi$  (over  $Prop$ ), a state  $s$  is said to *satisfy*  $\varphi$ , written  $s \models \varphi$ , iff  $\nu_s$  is a model of  $\varphi$ . A *goal*  $G$  is a set of states, i.e.,  $G \subseteq S$ , possibly represented as a propositional (goal) formula  $\varphi$ , in which case  $G \doteq \{s \in S \mid s \models \varphi\}$ . When  $T(s, a) \neq \emptyset$  we say that action  $a$  is *executable* in  $s$ . If  $s' \in T(s, a)$ ,  $s'$  is called a *successor* of  $s$  on  $a$ . A  $\mathcal{D}$ -*trace* is a sequence  $\tau = s_0, s_1, \dots, s_n$  s.t.  $s_{i+1} \in T(s_i, a)$ ,

for some  $a \in A$ . A *policy* for  $\mathcal{D}$  is a function  $\pi : S \mapsto A$ . A  $\mathcal{D}$ -trace  $\tau$  is induced by a policy  $\pi$  on  $\mathcal{D}$  iff  $s_{i+1} \in T(s_i, \pi(s_i))$ .

A planning domain and a goal define a *planning problem*. We say that a policy  $\pi$  is a *strong solution* [3] to a problem  $Prob = \langle \mathcal{D}, G \rangle$ , if every  $\mathcal{D}$ -trace  $\tau$  induced by  $\pi$  contains a state  $s' \in G$ . Intuitively, strong solutions are those which guarantee the achievement of a goal despite the non-determinism of all the executed actions.

The planning setting we consider here is that of non-deterministic planning under full observability. Model Based Planner (MBP) [3] is a very natural choice to compute strong solutions. Nonetheless, we chose a tool for verification and synthesis, Temporal Logic Verifier (TLV) [7], as it offers the possibility of easily building custom algorithms over transition systems. We plan to take advantage of such a flexibility in future extensions of this work. It is worth noting that, similarly to MBP, TLV exploits ordered binary decision diagrams (OBDDs) [2] to limit the space required by the search.

## 2.2 Markov Decision Processes

A Markov Decision Process is a tuple  $MDP = \langle S, A, T, \rho \rangle$  where:  $S$  is a set of *states*,  $A$  is a set of *actions*,  $T : S \times A \times S \rightarrow [0, 1]$  is the transition function.  $T(s, a, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$  is the probability that the current state changes from  $s$  to  $s'$  by executing action  $a$ .  $\rho : S \times \mathbb{R} \rightarrow [0, 1]$  is the reward function.  $\rho(s, r) = Pr(r_{t+1} = r | s_t = s)$  is the probability to get a reward  $r$  from being in state  $s$ . MDPs are the model of choice for domains in Reinforcement Learning (RL). Our method takes a planning problem as input, and produces an MDP suitably constrained to learn more quickly than in the original domain. The behavior of the agent is represented as a function  $\pi_t : S \times A \rightarrow [0, 1]$  called a *policy*, where  $\pi_t(s, a)$  is the probability of selecting action  $a$  in state  $s$  at time  $t$ . The expected cumulative discounted reward, that is the reward expected from each state  $s$  executing action  $a$ , and following a policy  $\pi$ , is defined as:  $Q : S \times A \rightarrow \mathbb{R}$ . Its value is given by:  $Q^\pi(s, a) = \mathbb{E} \left[ \sum_{t \geq 0} \gamma^t \rho(s_{t+1}, \cdot) | s_0 = s, a_0 = a \right]$  where  $0 < \gamma \leq 1$  is a discount factor. The discount factor is allowed to be 1 only in *episodic* tasks, where the MDP has *absorbing* states which once entered cannot be left. The reward is accumulated by executing  $a$  in  $s$  and following  $\pi$  thereafter.

## 2.3 Hierarchies of Abstract Machines

From the learning perspective, the problem of acting optimally in a fully observable environment has been studied under Reinforcement Learning. In particular, Hierarchical Reinforcement Learning (HRL) [1] is the field of RL most related to our work. In HRL, the designer provides structure to the policies searched, constraining the exploration in fully observable domains. Such structure can be provided in different forms, among which state machines.

A Hierarchy of Abstract Machines (HAM) [6] is a set of state machines that encodes a sketch of the solution for a given problem. The environment is described as a Markov Decision Process. The hierarchy constraints the policies

allowed, effectively reducing possible exploration. At any given time, a machine either specifies an action to be executed or is in a *choice* state. When in a choice state, the next action is determined by the underlying MDP, performing learning only in those states. HAMs are hierarchical machines, but in this paper we only make use of one-level machines, therefore we report a simplified description. The complete definition of HAMs is available in the original paper [6].

The composition of a machine  $H$  and the MDP  $M$ ,  $H \circ M$ , is obtained as described in the following. First, the Cartesian product of the state spaces of  $H$  and  $M$  is computed. Then, a new MDP over such a state space is defined by picking transitions either from the machine, or the MDP. An action is borrowed from  $H$  if the state is an action state, and from  $M$  if it is a choice state. Thus, only choice states have more than one action available. The reward is taken from the MDP if the transition is an action, while it is zero otherwise. The resulting process is still an MDP [6], and will be leveraged to learn in the representation we create through the planner. It is proved that there is no need to store the entire state space, since there exists always a Semi-Markov Decision Process (SMDP) such that its optimal policy is the same as  $H \circ M$ . Such an SMDP has only the states  $(q, s) \in H \times M$  in which  $q$  (the machine state) is a choice state.

The definition of the machine  $H$  is still an open problem, and is usually carried out by hand. To the best of our knowledge this is the first attempt to generate such a structure automatically. Our work differs from automated option discovery [8], as this usually consists in the identification of sub-goals, and learning of a procedure to reach them in the framework of Semi-Markov Decision Processes. Our work does not concern discovering a suitable goal, nor a representation of the sub-task, as both are part of the initial planning problem. Our method produces a structure that allows subsequent learning for a given task, and differently from the main focus of option learning, does not create new tasks.

### 3 Controller Generation and Learning

Our method for the specification of a controller is divided into two phases: generation and learning. At planning time the available model is used to produce a finite-state automaton as described in Section 3.1. Then, the learning phase begins by generating from such an automaton a HAM, as described in Section 3.2. Finally, at run time, the learning phase is carried out in the real environment.

#### 3.1 Finite-State Controller Generation

We start from a formal representation of the environment, encoded as a non-deterministic planning domain. The domain is given as input to the TLV system, which we exploit to compute a set of policies guaranteed to achieve the goal.

Our aim is to build, for a given domain  $\mathcal{D}$ , a set of policies that are strong solutions. To do so, we created a variant of the algorithm for strong planning described by Cimatti et al. [3]. Intuitively, the algorithm computes the set of all *minimal-cost* strong solutions for *Prob*, where by *cost* of a policy  $\pi$  we

---

**Algorithm 1.** Computes minimal-cost strong solutions

---

**Input:** a planning domain problem  $Prob = \langle \mathcal{D}, G \rangle$

**Output:** a set of minimal strong solutions

**Let**  $Z := G$ ,  $Z' := \emptyset$ ,  $T_G := \emptyset$ ;

**While**( $s_0 \notin Z \wedge Z \neq Z'$ ) **do**

**Let**  $N := \{(s, a) \in (S \setminus Z) \times A \mid T(s, a) \neq \emptyset \wedge T(s, a) \subseteq Z\}$ ; **Let**  $T_G := T_G \cup \{(s, a, s') \in S \times A \times S \mid (s, a) \in N \wedge s' \in T(s, a)\}$ ;

**Let**  $Z' := Z$ ;

**Let**  $Z := \{s \in S \mid \exists a. (s, a) \in N\} \cup Z$ ;

**End**

**if**( $s_0 \in Z$ ) **return**  $T_G$ ; **else return**  $\emptyset$ ;

---

mean the length of the longest trace induced by  $\pi$ , containing a goal state only as its last state. To do so, the algorithm iteratively computes the set  $Z$  of all the states from which a policy that achieves a goal state exists. At the end of the  $i$ -th iteration,  $Z$  contains the set of states from which a goal state can be achieved in at most (depending on the outcome of non-deterministic actions)  $i$  steps. The relation  $N$  associates a state  $s$  with those actions whose successors are all members of  $Z$ . The relation  $T_G$  completes  $N$  with the information on the transitions that take place as actions are executed.

Although, in this paper, we focus on strong solutions, weaker notions of solutions can be in principle considered. For instance, one could use weak solutions, i.e., those that guarantee at least a possibility of reaching a goal, or strongly cyclic ones, i.e., those that guarantee goal achievement under a fairness assumption on the action effects. Also mixing these notions is possible.

The shortest solutions are not necessarily the optimal ones, as the true cost depends on the environment and cannot in general be modeled. Therefore, while policies that extend the minimal cost (in terms of number of actions) may be considered of lower quality by the planner, they may yield better performance in practice. This is the subject of the following learning phase.

### 3.2 Finite-State Controller Learning

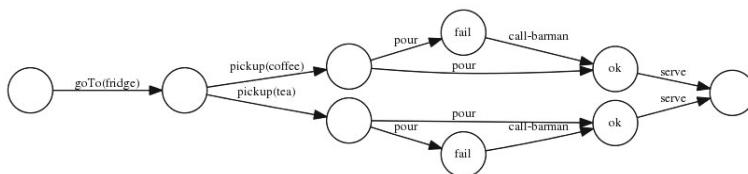
The controller generated by the previous phase (cf. Section 3.1) can be translated into a one-level HAM as follows.

The output of the controller generator is a graph  $G = \langle S, T_G \rangle$  where  $S$  is the set of states of the original domain  $\mathcal{D}$ . Essentially, when the domain is in state  $s$ , the agent is guaranteed that any action  $a$ , s.t. for some  $s'$   $T_G(s, a, s')$  holds, is a valid option to move closer to the goal.

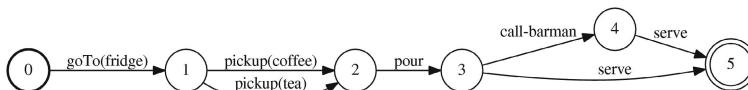
Considering  $G$  as an acceptor, with the actions on the transitions as its symbols, a HAM  $H$  is obtained as the minimal automaton [5] that accepts such a language. The minimization process removes the environment state information, and the one-to-one correspondence between the states of the automaton and the states of the domain is not maintained in the new representation.

During learning, this state machine is used in conjunction with the observations (the states of the domain's MDP) as described in Section 2.3. The dynamics of the MDP needs not to be known, as learning at choice points can be model-free. If the state space is fully observable the system falls in the category of Semi-Markov Decision Processes, and the related theory guarantees convergence.

As an example, consider the simple planning problem described in the following. Bar-bot has to help a customer, by serving either coffee or tea. The actions available are *goto(location)*, *pickup(object)*, *pour*, *call-barman*, and *serve*. *Pour* is a non-deterministic action, that can either succeed or fail. If it fails bar-bot can recover by calling the barman. The customer likes one between coffee and tea more than the other, but which one is unknown to the agent. A pure learning approach would start trying actions, initially at random, to discover that bar-bot has to go to the fridge before picking up coffee; or that it has to have picked up coffee, before being able to pour it. The model is fed to the planner, together with a goal description in Linear Temporal Logic. TLV produces the graph in Figure 1. This graph represents two different strong plans (as they recover from non-determinism): one to serve coffee and one to serve tea. There is still a one-to-one correspondence between states of the graph and states of the environment at this stage. Once the graph is minimized, such a correspondence is lost, together with the information of whether an action is actually permitted in a given state. In Figure 2, the minimized automaton is shown, which constitutes our automatically generated HAM. In this automaton you can see that *call-barman* and *serve* are both available in state 3. Call-barman, however, makes sense only if the previous *pour* action has failed. The HAM will introduce the state, if necessary, through the Cartesian product, as explained in Section 2.3. State 1 is a non-deterministic choice point, as both actions will be simultaneously available for execution. This decision will be learned, by getting reward,



**Fig. 1.** The automaton created by the generation phase



**Fig. 2.** The HAM obtained by minimizing the previous automaton

and discovering whether the client actually prefers coffee or tea. Such information was not available at planning time, and a pure planning agent cannot do better than choosing randomly among the options available.

## 4 Experimental Evaluation

We begin our experimental section by introducing the printing domain, where we performed experiments on a realistic simulator. The task considered in this paper consists in receiving printing requests, sending the papers to an appropriate printer, picking up papers, delivering them to the corresponding person and returning to a home location. In this setting we modeled several kinds of uncertainties: robot sensor noise affecting localization and navigation, probability of paper jams in the printers, variable time to fix the printer by a human operator.

The state space is composed by the following variables:  $location \in \{\text{home}, \text{at\_printer\_1}, \text{at\_printer\_2}\}$ , denotes the current position of the robot;  $docsReady \in \{0,1\}$ , indicates whether there are documents ready to be printed. For each printer:  $docsOnPrinter \in \mathbb{N}^+$  is the number of papers printed ready for pick-up;  $isStuck \in \{0,1\}$ , indicates whether this printer is stuck and must be fixed;  $helpRequested \in \{0,1\}$ , indicates whether this printer is stuck and help has been requested. Moreover, the robot has a fixed number of trays, 3 in our experiments, that can accommodate different papers. For each tray, a positive integer represents the number of papers in the tray.

The actions available to the robot are the following: `retrieveDocs` that loads papers to be printed, affecting `numDocsReady`; `print_n` that sends the next document to be processed to the printer  $n$ ; `goTo_l` that moves the robot to location  $l$ ; `collect` that, executed from the location of a printer, picks up all the papers from a printer and puts them on the first empty tray; `askForHelp` that, executed from the location of a stuck printer, signals that the printer is stuck, and gets the printer fixed by a human operator; `deliver` that delivers all the papers in the trays to the appropriate people, terminating the episode.

The performance of the task is measured in terms of a score that considers both the number of papers printed and delivered, and the time to terminate each episode. For each paper successfully delivered the agent obtains a reward of 500. If two or more documents are mixed in a tray it receives 300 per document. Each action other than `deliver`, returns a negative reward equal to the opposite of the time, in seconds, taken by the action. Note how this information, as much as the probability of each printer to be jammed, is not known in advance. Therefore, it cannot be part of the model and exploited during the planning phase, but the agent must adapt during the execution.

### 4.1 Experimental Setting

The main goal of the presented experiments is to show the capabilities of the proposed method to quickly learn an optimal strategy, providing a comparison with the execution of the controller generated in the planning phase without learning, and with pure learning.

Given the domain introduced above, a domain description is encoded in the language SMV, the input language for TLV, using a compilation scheme similar to the one described by Giunchiglia and Traverso [4]. The algorithm described in Section 3.2 returns a finite-state controller that is given in input to the learning phase. In this example, the returned controller has over 100 states and over 200 transitions.

The environment setting during the experiments (not known to the robot) is the following: the printer closest to the robot home position is faulty (0.8 probability of paper jam) and there is a quick response for fixing the printer (about a third of the time needed to go to the other printer); the other printer is faulty with a probability 0.2.

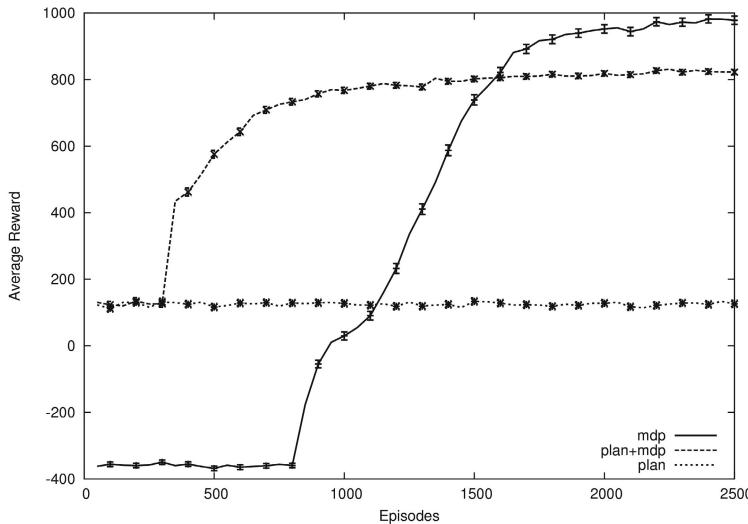
We perform three series of experiments: 1) using the controller generated by the planning phase with a random choice of actions in case of multiple possible actions in a state; 2) using a pure learning approach on the MDP built for this scenario; 3) using the combined planning+learning method described in this paper, where the controller generated by the planning step is refined through reinforcement learning.

We used Sarsa( $\lambda$ ) for both the learning cases, optimizing the parameters for the different representations. Common to both,  $\lambda = 0.9$ ,  $\gamma = 0.999$ ,  $\alpha = 0.1$ . The value function is initialized optimistically, and the exploration is performed following a  $\epsilon$ -greedy strategy, where  $\epsilon$  decays at different rates, depending on the representation. On the full MDP,  $\epsilon$  going from 1 to 0.05 in 800 episodes performed best. With our method, the best decaying rate proved to be from 1 to 0.02 in 300 episodes. Each run has been repeated 100 times to average over the stochasticity of the domain.

## 4.2 Results

The results are shown in Figure 3. At the beginning of the learning phase, and for the first episodes, the 'plan' and the 'plan+mdp' approaches have the same values since they both act randomly over the generated controllers. However, after the initial exploratory phase of 300 episodes, learning allows to significantly improve on the pure planning approach, which keeps a constant average performance. On the other hand, the pure learning approach starts from a lower solution, since it is exploring a large space that contains many non-optimal behaviors. The figure clearly shows the increased learning rate of the 'plan+mdp' approach that is operating on a much smaller search space, with respect to 'mdp'.

In this case, the globally optimal solution was not part of the generated controller therefore the mdp eventually passes learn+mdp. This is common to all HRL methods, since limiting the search space cannot prevent from excluding the optimal path. Nonetheless, the learned behavior is guaranteed to be an improvement on the pure planning approach, as every solution computed by the planner is always part of the generated controller. Moreover, learning alone reaches the optimal behavior after a number of episodes which is usually not feasible when dealing with real systems. Last, focusing on exploration, note how exploring in the constrained space is always safe, as all the trajectories lead to the goal state.



**Fig. 3.** The average cumulative reward obtained with learning alone, planning alone, and their proposed combination

Unconstrained exploration, on the other hand, is in general dangerous in addition to costly. Therefore, the planning+learning method described in this paper guarantees to reach a “good” (close to optimal) performance with respect to a pure learning approach with significantly fewer experiments to be performed.

## 5 Discussion and Conclusion

We presented a method to generate and learn finite state controllers, combining techniques from both automatic planning and reinforcement learning. This method allows for suitably integrating the advantages of a planning approach in reducing the set of possible trajectories in a transition system (discarding those that do not lead to a solution or that are too expensive, with the advantages of a learning approach that is able to improve performance over time from on-line experiments in the environment.

The application of the proposed method to problems involving real systems in real environments (for example, robotic applications) has the advantage of finding “good” or optimal solutions for a complex task in the presence of an incomplete description of the domain with a small number of experiments. This is not possible when using either pure planning or pure learning methods.

We evaluated our method in a realistic domain, where the uncertainty behind the model constitutes the main difficulty, rather than the size of the domain itself. Such uncertainty can be alleviated by providing structure in the space of the task, and adapting to the environment. As the adaptation is constrained, both tracking non-stationarity in the environment (traditionally a hard problem) and exploring is faster and safer.

While in this paper we focus only on a two-step process, where planning is followed by learning, our approach can be extended to an iterative process where planning and on-line learning are interleaved. Specifically, at every iteration the learning phase returns a refined version of the controller that is used to refine the planning domain, thus allowing for the generation of a new controller, provided again as input to the learning phase.

## References

1. Barto, A.G., Mahadevan, S.: Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems* 13(1-2), 41–77 (2003)
2. Bryant, R.E.: Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.* 24(3), 293–318 (1992)
3. Cimatti, A., Pistore, M., Roveri, M., Traverso, P.: Weak, Strong, and Strong Cyclic Planning via Symbolic Model Checking. *Artif. Intell.* 147(1-2), 35–84 (2003)
4. Giunchiglia, F., Traverso, P.: Planning as Model Checking. In: Biundo, S., Fox, M. (eds.) ECP 1999. LNCS, vol. 1809, pp. 1–20. Springer, Heidelberg (2000)
5. Mohri, M.: Minimization algorithms for sequential transducers. *Theoretical Computer Science* 234(1-2), 177–201 (2000)
6. Parr, R., Russell, S.: Reinforcement learning with hierarchies of machines. In: Advances in Neural Information Processing Systems, pp. 1043–1049 (1998)
7. Pnueli, A., Shahar, E.: A Platform for Combining Deductive with Algorithmic Verification. In: Alur, R., Henzinger, T.A. (eds.) CAV 1996. LNCS, vol. 1102, pp. 184–195. Springer, Heidelberg (1996)
8. Stolle, M.: Automated discovery of options in reinforcement learning. PhD thesis, McGill University (2004)

# FactForge: Data Service or the Diversity of Inferred Knowledge over LOD

Mariana Damova, Kiril Simov, Zdravko Tashev, and Atanas Kiryakov

Ontotext, Tsarigradsko Chosse 47A, Sofia 1504, Bulgaria

{mariana.damova, zdravko.tashev, naso}@ontotext.com,  
kivs@bultreebank.org

**Abstract.** Linked Open Data movement is maturing. LOD cloud increases by billions of triples yearly. Technologies and guidelines about how to produce LOD fast, how to assure their quality, and how to provide vertical oriented data services are being developed (LOD2, LATC, baseKB). Little is said however about how to include reasoning in the LOD framework, and about how to cope with its diversity. This paper deals with this topic. It presents a data service – FactForge – the biggest body of general knowledge from LOD on which inference is performed. It has close to 16B triples available for querying, derived from about 2B explicit triples, after inference and some OWLIM repository specific optimization. We discuss the impacts of the reference layer of FactForge and inference on the diversity of the web of data, and argue for a new paradigm of data services based on linked data verticals, and on inferred knowledge.

**Keywords:** Linked Open Data, Linked Data Management, FactForge, reference layer, upper level ontology, PROTON, Semantic Web.

## 1 Introduction

Linked Open Data movement is maturing. Not only LOD cloud increases by billions of triples yearly, but also technologies and guidelines about how to produce LOD fast, how to assure their quality, and how to provide vertical oriented data services are being developed (LOD2 [21], LATC [19], baseKB [1]). The diversity of LOD makes their use and querying extremely challenging, as one has to be intimately familiar with the schemata underlying each dataset. This calls for approaches for efficiently managing this diversity. Although each LOD dataset has an inherent ontological model, the inference that can produce new knowledge and make the available information pools even richer has not been sufficiently explored and its value proven. This paper is an attempt in this direction. A new version of FactForge data service is presented, section 3, and some effects of the inferred knowledge are being outlined and analyzed from the perspective of the diversity they bring, section 4. Section 5 concludes the paper, whereas next section, Section 2 discusses related work.

## 2 Related Work

The diversity of LOD makes their use and querying extremely challenging, as one has to be intimately familiar with the schema underlying each dataset to make queries across datasets. This problem has been addressed in initiatives like schema.org [3], which involves the creation of unified models for general knowledge, UMBEL [31], [2], which has the ambition to become the general reference layer allowing to access the web, BLOOMS+ [16], which uses automatic ontology alignment to an upper-level ontology to provide access to LOD. All these approaches testify of the effort to manage the heterogeneity of the web of data in a practically feasible way. They all try to involve the notion of a golden standard at schema level to allow better interoperability of LOD and the WWW in general, are indicative for the search of a solution along these lines. None of these approaches however addresses inference. We present a data service, which implements a unique method to manage and query LOD – reason-able view. It includes reasoning and inferred knowledge. The new version of FactForge which is presented in this paper aligns with these views for several years now.

## 3 FactForge

FactForge is a reason-able view [17], [18] to the Web of Linked Data, made up of 8 of the central LOD datasets, which have been selected and refined in order to serve as a useful index and entry point to the LOD cloud and to present a good use-case for large-scale reasoning and data integration. The compound dataset of FactForge is the largest body of heterogeneous general knowledge on which inference has been performed. It counts 1.7 billion explicit statements; 15 billion retrievable statements available after inference and owl:sameAs expansion which include 1.4 billion inferred statements. Full materialization with respect to OWL-Horst optimized [30] is performed at the time of data loading. The OWL-Horst optimized inference forward chaining rules are part of the distribution of OWLIM-SE. They allow for defining constraints, defining rules to handle types, transitive relations. The standard OWLIM-SE OWL-Horst optimized ruleset is being augmented with additional rules enriching the datasets with instances which cover structural differences in their schemata. The additional rules also cover mappings of LOD schemata to FactForge reference layer ontologies. The reasoning over FactForge has been discussed at length in [5]. The datasets combined in FactForge are: DBpedia [9], Freebase [13], Geonames [14], CIA World Factbook [6], Lingvoj [20], MusicBrainz [22], WordNet [32], New York Times [23]. To make the access to these heterogeneous data more efficient, a reference layer is built on top of the datasets. This is an upper-level ontology PROTON, mapped to the schemata of DBpedia, Geonames and Freebase covering diverging conceptualizations, structural differences, missing concepts [7], [8]. This reference layer makes loading of the LOD ontologies unnecessary, optimizing the reasoning processes and allows for quick and seamless data integration of new

datasets with the entire LOD segment of FactForge. It also ensures better interfacing with other components via SPARQL as the queries are more compact and easy to formulate, respond time are faster, because of less joins that are employed, and a wealth of inferred knowledge across the datasets, which allows for real journey of knowledge discovery, and navigation from different stand points. PROTON is a light weight ontology of about 500 classes and 150 properties. The mappings allow to query the entire FactForge by using only the concepts of PROTON. The effect of this is that the SPARQL queries are much easier to formulate, the results are more, compared to queries with LOD dataset predicates, as triples with differently named semantically identical properties are retrieved by means of a single PROTON predicate, the inference brings valuable new insight as new knowledge is derived from all datasets in combination. Except for the datasets and PROTON, FactForge includes several popular schemata, e.g. DCMI Metadata Terms (Dublin Core - DC) [10], SKOS [29] (Simple Knowledge Organization System), RSS [24], FOAF [12]. All those participate in the materialization process during loading and provide additional knowledge and means to query the compound dataset of FactForge, see Section 4.

FactForge provides several methods to explore the combined dataset that exploits some of the advanced features of OWLIM-SE. Firstly, ‘RDF Search and Explore’ keywords with a real-time auto-suggest feature ordered by ‘RDF Rank’ (similar to Google’s Page Rank). Secondly, a SPARQL page allows users to write their own queries with clickable options to add each of the known namespaces. Lastly, a graphical search facility called ‘RelFinder’ [27] that discovers paths between selected nodes. This is a computationally intensive activity and the results are displayed and updated dynamically during each iteration. The resulting graph can be reshaped by the user with simple click and drag operations.

## 4 Diversity through Inferred Knowledge

The exploration mode of FactForge allows to inspect the inferred knowledge in the query results. Figure 1 shows the results of a search about the Copenhagen, Denmark. The explicit statements are given in blue, whereas the implicit statements are given in red. We observe several things with respect to diversity in this example. Firstly, the transitive closure over the Geonames predicate `geo-ont:parentFeature` derive facts that Copenhagen is in Denmark and in Europe, expressed both with DBpedia and Geonames instance. Secondly, the PROTON predicate `p top:locatedIn` however, has about 10 more implicit facts about the location of Copenhagen, e.g. that it is in Northern Europe, in Scandinavia, in a Nordic country, etc., which increases the variety of information available for querying. Thirdly, Dublin Core predicates give information about Geonames feature codes, which provides the freedom to access geographical information via generic predicates.

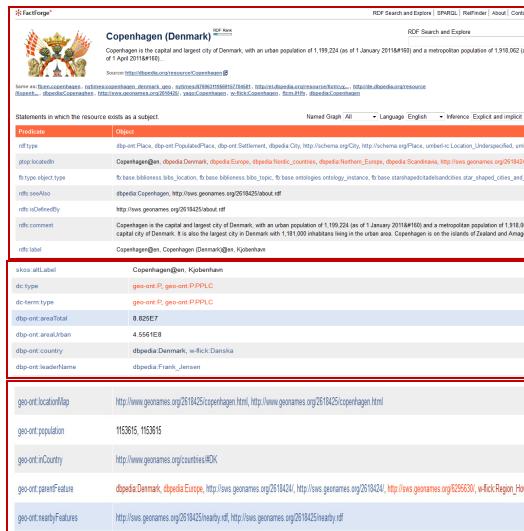


Fig. 1.

Further exploring North Europe, which we pick from the results of Copenhagen above, point us to several inferred facts with relevance to diversity, c.f. Figure 2.

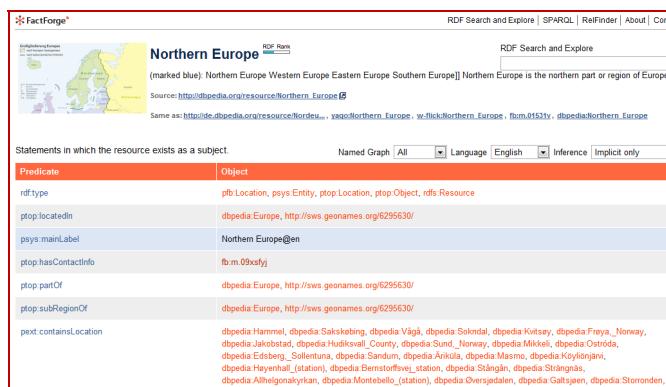


Fig. 2.

Two predicates refer to the location of Northern Europe in Europe, e.g. `ptop:partOf`, and `ptop:subRegionOf`. We observe an inferred reference to a Website of Northern Europe found in Freebase, and more than 50 locations that are inferred to be in Northern Europe.

As pointed out in Section 3, PROTON mappings allow to query FactForge with PROTON predicates. This does not prevent however, to use the LOD datasets original predicates to query FactForge. The example above showed that facts can be inferred, built with predicates from Dublin Core. Querying with PROTON though brings more

results with additional, sometimes very useful information. For example, a query about airports of 50 miles away from London, brings 13 results when formulated with predicates from LOD, and 20 results when formulated with PROTON predicates. Curiously the additional 7 results come from Geonames, and point to airport Terminals, c.f. figure 3.

SPARQL Query	
Results for your query (20) - <a href="#">Edit query</a>	
airport	label
dbpedia_London_Heathrow_Airport	London Heathrow Airport@en
dbpedia_London_City_Airport	London City Airport
dbpedia_RAF_Northolt	Royal Air Force Northolt 90px@en
dbpedia_Antwerp_International_Airport	Antwerp International Airport@en
dbpedia_Croydon_Airport	Croydon Airport@en
dbpedia_London_Biggin_Hill_Airport	London Biggin Hill Airport@en
dbpedia_Elstree_Airfield	Elistree Airfield@en
dbpedia_London_Heliport	London Heliport@en
dbpedia_Heston_Aerodrome	Heston Aerodrome@en
dbpedia_Heston_Aerodrome	Heston Aerodrome@en
dbpedia_Stapleford_Aerodrome	Stapleford Aerodrome@en
dbpedia_North_Weald_Airfield	North Weald Airfield@en
<a href="http://swe.geonames.org/6301524/">http://swe.geonames.org/6301524/</a>	Northolt
dbpedia_Stag_Lane_Aerodrome	Stag Lane Aerodrome@en
<a href="http://swe.geonames.org/6296597/">http://swe.geonames.org/6296597/</a>	Biggin Hill
<a href="http://swe.geonames.org/6691396/">http://swe.geonames.org/6691396/</a>	London Heathrow Terminal 1
<a href="http://swe.geonames.org/6691397/">http://swe.geonames.org/6691397/</a>	London Heathrow Terminal 2
<a href="http://swe.geonames.org/6691395/">http://swe.geonames.org/6691395/</a>	London Heathrow Terminal 3
<a href="http://swe.geonames.org/2647216/">http://swe.geonames.org/2647216/</a>	London Heathrow Airport@en
<a href="http://swe.geonames.org/6691394/">http://swe.geonames.org/6691394/</a>	London Heathrow Terminal 4

Fig. 3.

Finally, inferred knowledge can bring to completely unexpected knowledge. For instance, the city of Missouri is inferred to be the subject in the work of the science fiction writer Joel Rosenberg, cf. figure 4.



The screenshot shows the FactForge RDF search interface. At the top, there's a navigation bar with links for 'RDF Search and Explore', 'SPARQL', 'RefFinder', 'About', and 'Contact'. Below the navigation is a search bar with the placeholder 'RDF Search and Explore'. The main content area displays a query result for 'Missouri'.

Predicate	Object
rdf:type	dbp:ont:AdministrativeRegion, dbp:ont:Place, dbp:ont:PopulatedPlace, dbp:ont:Settlement, dbp:ont:Village, dbpedia:City, http://schema.org/City, http://schema.org/Place
foaf:homepage	<a href="http://state.mo.us">http://state.mo.us</a> , <a href="http://www.mo.gov">http://www.state.mo.us</a>
skos:inScheme	<a href="http://data.nytimes.com/elements/mnyd_geo">http://data.nytimes.com/elements/mnyd_geo</a>
skos:prefLabel	Missouri, Missouri@en
skos:altLabel	Missouri, Missouri@en, Show-Me State@en, State of Missouri@en
skos:isSubjectOf	<a href="#">dbpedia:Joel_Rosenberg_science_fiction_author</a>
dc:type	gn:A, gn:A ADM1
dc-term:type	gn:A, gn:A ADM1
dbp:ont:areaCode	573@en

Fig. 4.

FactForge is accessible at <http://ff-dev.ontotext.com>. These examples present evidence that the inferred knowledge over LOD is a valuable resource of information, which calls for a new paradigm of data services, based on both explicit and implicit knowledge, not only on federating results over linked data physically dispersed over distant servers.

## 5 Conclusion and Future Work

This paper presented FactForge data service, the biggest body of heterogeneous generic LOD knowledge on which inference is performed. We outlined a series of examples showing the ways FactForge knowledge base gets enriched through the OWL-Horst closure, and the diversity of ways of accessing and approaching these data. Analyzing the inferred facts, points to the negative effects of noisy explicit data. Our future plans include the development of methods to identify inconsistencies in the input explicit data based on the ontological models, and the available explicit facts. Additionally, implementation of the loading inference chains logs will make it possible to identify deviations in the application of the inference rules. Finally, the usefulness of inferred knowledge for creating and managing diversity on the web of data will be further argued for.

## References

1. baseKB, <http://basekb.com/>
2. Bergman, M.K.: Seeking a Semantic Web Sweet Spot. Blog post (2011), Retrieved from <http://www.mkbergman.com/946/seeking-a-semantic-web-sweet-spot>
3. Bing Google Yahoo (2011), Retrieved from Bing Google Yahoo unite to build the Web of Objects, [http://www.bing.com/community/site\\_blogs/b/search/archive/2011/06/01/bing-google-and-yahoo-unite-to-build-the-web-of-objects.aspx?form=MFEHPPG&publ=TWITTER&crea=TEXT\\_MFEHPPG\\_SM0602\\_cc0602\\_TW006\\_1x](http://www.bing.com/community/site_blogs/b/search/archive/2011/06/01/bing-google-and-yahoo-unite-to-build-the-web-of-objects.aspx?form=MFEHPPG&publ=TWITTER&crea=TEXT_MFEHPPG_SM0602_cc0602_TW006_1x)
4. Bishop, B., Kiryakov, A., Ognyanoff, D., Peikov, I., Tashev, Z., Velkov, R.: OWLIM: A family of scalable semantic repositories. Semantic Web Journal. Special Issue: real-World Applications of OWL (2011); Hitzler, P. (ed.)
5. Bishop, B., Kiryakov, A., Tashev, Z., Damova, M., Simov, K.: OWLIM Reasoning over FactForge. In: Proceedings of OWL Reasoner Evaluation Workshop (ORE 2012), Collocated with IJCAR 2012, Manchester, UK (2012)
6. CIA The World Factbook (2011), Retrieved 2011 from <https://www.cia.gov/library/publications/the-world-factbook/>
7. Damova, M., Kiryakov, A., Grinberg, M., Bergman, M., Giasson, F., Simov, K.: Creation and Integration of Reference Ontologies for Efficient LOD Management. In: Stellato, A., Pazienza, M.T. (eds.) Semi-Automatic Ontology Development: Processes and Resources. IGI Global, USA (2012)
8. Damova, M., Kiryakov, A., Simov, K., Petrov, S.: Mapping the Central LOD Ontologies to PROTON Upper-Level Ontology. In: Proceedings of Ontology Matching Workshop, International Semantic Web Conference, Shanghai, China (November 2010)
9. DBpedia (2011), Retrieved from Structured information from Wikipedia, <http://DBpedia.org>

10. Dublin Core (2011), Retrieved 2011 from Dublin Core Metadata Element Set,  
<http://dublincore.org/documents/dces/>
11. FactForge (2011), Retrieved from A Reasonable View to the Web of Data,  
<http://factforge.net>, <http://www.ontotext.com/factforge>
12. FOAF, <http://www.foaf-project.org/>
13. Freebase (2011), Retrieved from <http://www.firebaseio.com/>
14. Geonames (2011), Retrieved 2011 from A geographical database,  
<http://www.geonames.org>
15. Heim, P., Hellmann, S., Lehmann, J., Lohmann, S., Stegemann, T.: RelFinder: Revealing Relationships in RDF Knowledge Bases. In: Chua, T.-S., Kompatsiaris, Y., Mérialdo, B., Haas, W., Thallinger, G., Bailer, W. (eds.) SAMT 2009. LNCS, vol. 5887, pp. 182–187. Springer, Heidelberg (2009)
16. Jain, P., Yeh, P.Z., Verma, K., Vasquez, R.G., Damova, M., Hitzler, P., Sheth, A.P.: Contextual Ontology Alignment of LOD with an Upper Ontology: A Case Study with Proton. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 80–92. Springer, Heidelberg (2011)
17. Kiryakov, A., Momtchev, V.: Two Reason-able Views to the Web of Linked Data. Paper Presented at the Semantic Technology Conference, San Jose, USA (June 2009)
18. Kiryakov, A., Ognyanoff, D., Velkov, R., Tashev, Z., Peikov, I.: LDSR: Materialized Reasonable View to the Web of Linked Data. In: Patel-Schneider, R.H. (ed.) Proceedings of OWLED 2009, Chantilly, USA (2009)
19. LATC, <http://latc-project.eu/>
20. Lingvoj, <http://www.lingvoj.org/>
21. LOD2, <http://lod2.eu/Welcome.html>
22. MusicBrainz, community music metadatabase (2011), Retrieved 2011 from  
<http://musicbrainz.org>
23. New York Times Linked Open Data (2011), Retrieved 2011 from  
<http://data.nytimes.com/>
24. OWLIM (2011), Retrieved from <http://www.ontotext.com/owlim>
25. PROTON 3.0, PROTON 3.0 Documentation. Sofia, Bulgaria (2011),  
<http://www.ontotext.com/proton-ontology>
26. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF, W3C Recommendation (January 15, 2008), <http://www.w3.org/TR/rdf-sparql-query/>
27. RelFinder, <http://www.visualdataweb.org/relnfnder.php>
28. RSS, <http://www.rssboard.org/rss-specification>
29. Simple Knowledge Organization System (2011), Retrieved from  
<http://www.w3.org/2004/02/skos/>
30. ter Horst, H.J.: Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 668–684. Springer, Heidelberg (2005)
31. Upper Mapping and Binding Exchange Layer (UMBEL) (2011), Retrieved from  
<http://www.umbel.org/>
32. Wordnet (2011) Retrieved from A lexical database for English,  
<http://wordnet.princeton.edu/>

# From Path-Consistency to Global Consistency in Temporal Qualitative Constraint Networks

Nouhad Amaneddine<sup>1</sup> and Jean-François Condotta<sup>2</sup>

<sup>1</sup> Department of Information Technology  
and Computing, Faculty of Computer Studies

The Arab Open University-Lebanon  
[namaneddine@aou.edu.lb](mailto:namaneddine@aou.edu.lb)

<sup>2</sup> Université Lille-Nord de France, Artois  
CRIL-CNRS, UMR 8188  
rue de l'université, SP 16, F62307 Lens  
[condotta@cril.fr](mailto:condotta@cril.fr)

**Abstract.** We study in this paper the problem of global consistency for qualitative constraints networks (QCNs) of the Point Algebra (PA) and the Interval Algebra (IA). In particular, we consider the subclass  $\mathcal{S}_{\text{PA}}$  corresponding to the set of relations of PA except the relations  $\{<, =\}$  and  $\{>, =\}$ , and the subclass  $\mathcal{S}_{\text{IA}}$  corresponding to pointizable relations of IA one can express by means of relations of  $\mathcal{S}_{\text{PA}}$ . We prove that path-consistency implies global consistency for QCNs defined on these subclasses. Moreover, we show that with the subclasses corresponding to convex relations, there are unique greatest subclasses of PA and IA containing singleton relations satisfying this property.

## 1 Introduction

Qualitative Temporal Reasoning (QTR) is an area of computer science dealing with qualitative information used to configure temporal entities. A calculus in QTR introduces a particular finite set of base relations on a set of elements representing temporal entities. Each base relation is an abstraction of concrete configurations about the relative positions of entities. For applications, a qualitative description can be more appropriate than a numerical description, in particular when information is not accurate or not necessary.

Interval Algebra (IA) is the first and the most known qualitative calculus. It was introduced by Allen [1] aiming at reasoning about temporal information on intervals of the rational number line. For this calculus, intervals denote non punctual activities or events, and thirteen base relations represent all orderings of the four endpoints of two intervals (see Figure 1(a)). Point Algebra (PA) [13] is another formalism of QTR less expressive than Allen's calculus. It considers points to represent punctual temporal entities and three base relations corresponding to all relative position of two points on the rational line. These last three decades, other numerous QTR formalisms have been proposed and studied. In this paper, we will be concerned uniquely with PA and IA widely used in Artificial Intelligence systems.

In QTR, Qualitative Constraint Networks (QCNs) are typically used to express information about temporal situations. A constraint represents a set of acceptable qualitative

configurations between some entities, it is defined by a set of base relations of the qualitative calculus. Given a QCN, several problems can arise. One of these problems is the consistency problem. It consists of finding out whether the information contained in the QCN is consistent or not. For some qualitative formalisms, such as in PA, the consistency problem can be polynomial. In the general case, in IA for example, this problem is NP-hard for the full set of relations. Another fundamental reasoning problem concerning QCNs is the minimal labeling problem. It consists of finding the equivalent minimal QCN, *i.e.* the QCN containing the strongest entailed relations for each pair of variables (for binary QCN).

All subclasses of the Allen's Calculus for which the consistency problem is polynomial have been identified [5]. For some of these tractable subclasses, the path-consistency method that computes the closure under composition of the QCN can be used to solve the consistency problem and sometimes the minimal labeling problem. In [8], Nebel *et al.* have shown that the subclass of ORD-Horn relations, also called preconvex relations [6], is the unique greatest tractable subclass of IA among the subclasses containing singleton relations. For an ORD-Horn QCN, the path-consistency method solves the consistency problem, but still, it cannot obtain the corresponding minimal QCN. The subclass of convex relations [9], also called continuous relations, is a subset of the ORD-Horn subclass for which the path-consistency method solves the minimal labeling problem [10]. Furthermore, for this subclass, path-consistent QCNs are globally consistent QCNs. Global consistency [3] is a property that is stronger than minimality, it provides a solution of a QCN without backtracking. The subclass of pointizable relations of IA [12] corresponds to relations for which constraints on intervals can be translated into constraints of PA. A pointizable relation is an ORD-Horn relation and each convex relation is a pointizable relation. For pointizable QCNs of IA and QCNs of PA, path-consistency is not sufficient to obtain minimality. Indeed, van Beek [10] shows that for these QCNs, the strong 4-consistency is required to establish minimality.

By considering the QCN language, global consistency cannot always be reached. In [4], Koubarakis has studied a temporal constraint language that is more expressive than QCNs and proves that global consistency for QCN of PA translated into this language can be reached by means of strong 5-consistency.

In this paper, we are concerned with identifying all the subclasses of IA containing singleton relations for which path-consistency implies global consistency. We consider the subclass of IA corresponding to pointizable relations one can translate into PA without inequations, *i.e.* without  $\leq$  and  $\geq$ . For this subclass, denoted in this paper by  $S_{IA}$ , we show that path-consistency implies global consistency. In a second step, we prove that the subclasses of convex relations and  $S_{IA}$  are the unique greatest subclasses of IA among the subclasses containing the singleton relations for which global consistency can be established using the path-consistency method.

## 2 Preliminaries

A (binary) temporal qualitative calculus is defined by a finite set  $B$  of base relations on a domain  $D$ . The elements of  $D$  represent temporal entities, and the elements of  $B$  represent all possible configurations between two entities.  $B$  is a set that satisfies the

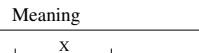
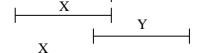
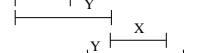
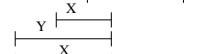
following properties [7]:  $B$  forms a partition of  $D \times D$ ,  $B$  contains the identity relation  $\text{Id}$  on  $D$ , and  $B$  is closed under the converse operation ( $^{-1}$ ). A (complex) relation is the union of some base relations and is represented by the set containing them. Hence, the set  $2^B$  will represent the set of relations of the qualitative calculus. As a set of subsets,  $2^B$  is equipped with the usual set-theoretic operations. As a set of binary relations, it is also equipped with the converse operation and the weak composition operation. The converse of a relation  $r \in 2^B$ , denoted by  $r^{-1}$ , is the union of the converses of the base relations contained in  $r$ . The weak composition operation (also called algebraic composition) that we denote by  $\diamond$  is defined by:  $\forall a, b \in B, a \diamond b = \{c \in B : \exists x, y, z \in D \mid x \, a \, z \wedge z \, b \, y \wedge x \, c \, y\}; \forall r, s \in 2^B, r \diamond s = \bigcup_{a \in r, b \in s} \{a \diamond b\}$ . Note that  $r \diamond s$  is the smallest relation of  $2^B$  including the usual relational composition  $r \circ s = \{(x, y) \in D \times D : \exists z \in D \mid x \, r \, z \wedge z \, s \, y\}$ .

The domain of IA is the set  $D_{IA} = \{(x^-, x^+) \in \mathbb{Q} \times \mathbb{Q} : x^- < x^+\}$  since temporal entities are represented by intervals of the rational line. The set of base relations of this calculus is the set  $B_{IA} = \{eq, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$ , see Figure 1. PA is based on three base relations between points on the line. More precisely, the domain of PA is  $D_{PA} = \mathbb{Q}$  and the set of base relations of this calculus is the set  $B_{PA} = \{<, >, =\}$  representing all orderings of two points on the line. Note that for PA and IA, the weak composition and the usual relational composition are identical.

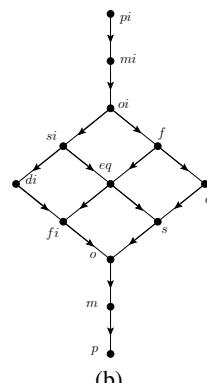
Formally, a Qualitative Constraint Network (QCN) is defined as follows:

**Definition 1.** A QCN is a pair  $\mathcal{N} = (V, C)$  where:  $V$  is a non empty finite set of variables;  $C$  is a mapping that associates a relation  $C(v, v') \in 2^B$  with each pair  $(v, v')$  of  $V \times V$ .  $C$  is such that  $C(v, v) \subseteq \{\text{Id}\}$  and  $C(v, v') = (C(v', v))^{-1}$  for every  $v, v' \in V$ .

For a QCN  $\mathcal{N} = (V, C)$  we have: A *partial solution* of  $\mathcal{N}$  on  $V' \subseteq V$  is a mapping  $\sigma$  defined from  $V'$  to  $D$  such that for every pair  $(v, v')$  of variables in  $V'$ ,  $(\sigma(v), \sigma(v'))$  satisfies  $C(v, v')$ , i.e. there exists a base relation  $b \in C(v, v')$  such that  $(\sigma(v), \sigma(v')) \in b$ . A *solution* of  $\mathcal{N}$  is a partial solution of  $\mathcal{N}$  on  $V$ .  $\mathcal{N}$  is *consistent* iff it admits a solution. Two QCNs are *equivalent* iff they admit the same set of solutions.  $\mathcal{N}$  is  $k$ -consistent,

Relation	Symbol	Inverse	Meaning
precedes	p	pi	
meets	m	mi	
overlaps	o	oi	
starts	s	si	
during	d	di	
finishes	f	fi	
equals	eq	eq	

(a)



(b)

**Fig. 1.** (a) The base relations of IA and (b) the interval lattice

with  $|V| \geq k \geq 1$ , if any partial solution of  $\mathcal{N}$  on  $V' \subset V$  with  $|V'| = |k - 1|$  can be extended to any variable  $v \in V \setminus V'$ .  $\mathcal{N}$  is strongly  $k$ -consistent, with  $|V| \geq k \geq 1$ , iff  $\mathcal{N}$  is  $i$ -consistent for  $i \in \{1, \dots, k\}$ .  $\mathcal{N}$  is globally consistent iff  $\mathcal{N}$  is strongly  $k$ -consistent with  $k = |V|$ . Moreover,  $\mathcal{N}$  is path-consistent iff  $\mathcal{N}$  is 3-consistent.

A QCN  $\mathcal{N} = (V, C)$  is  $\diamond$ -consistent or *closed under weak composition* iff  $\forall v, v', v'' \in V, C(v, v') \subseteq C(v, v'') \diamond C(v'', v')$ . The *closure* under weak composition of  $\mathcal{N}$ , denoted by  $\diamond(\mathcal{N})$ , is the greatest  $\diamond$ -consistent subQCN of  $\mathcal{N}$ ;  $\diamond(\mathcal{N})$  is equivalent to  $\mathcal{N}$ . This QCN can be obtained by iterating the triangulation operation  $C(v, v') \leftarrow C(v, v') \cap (C(v, v'') \diamond C(v'', v'))$  for all  $v, v', v'' \in V$  until a fix point is reached. For PA and IA, a QCN  $\mathcal{N}$  is  $\diamond$ -consistent iff  $\mathcal{N}$  is path-consistent. As we will be concerned with QCN of PA and IA, we will use the property of path-consistency and the property of  $\diamond$ -consistency indifferently. Moreover, the method consisting of computing the closure under weak composition will be called the *path-consistency method*. This method can be implemented by an algorithm running in  $O(n^3)$  time where  $n = |V|$ .

Given a set of base relations  $B$  and a set  $\mathcal{A} \subseteq 2^B$ , we will denote by  $\overline{\mathcal{A}}$  the closure of  $\mathcal{A}$  under converse, intersection and weak composition. In the case where  $\mathcal{A} = \overline{\mathcal{A}}$  we will say that  $\mathcal{A}$  is a subclass of  $2^B$ . Given a relation  $r$  of  $2^B$  and a subclass  $\mathcal{A} \subseteq 2^B$  containing the total relation (the relation containing all base relations),  $\mathcal{A}(r)$  will denote the smallest relation of  $\mathcal{A}$  including the relation  $r$ . Moreover, given a QCN  $\mathcal{N} = (V, C)$ ,  $\mathcal{A}(\mathcal{N})$  will denote the QCN  $\mathcal{N}' = (V, C')$  defined by  $C'(v, v') = \mathcal{A}(C(v, v'))$  for all  $v, v' \in V$ .

The subclass  $\mathcal{P}$  corresponds to the pointizable relations of IA, *i.e.* the relations of IA that can be expressed by constraints of PA.  $\mathcal{P}$  contains 188 relations. Given a QCN  $\mathcal{N} = (V, C)$  of IA defined on  $\mathcal{P}$ , we will denote by  $\text{PA}(\mathcal{N})$  the QCN corresponding to the translation of  $\mathcal{N}$  into PA (see [11] for the translation of each relation of  $\mathcal{P}$ ). To each variable  $v \in V$  is associated two variables  $v^-$  and  $v^+$  of  $\text{PA}(\mathcal{N})$  representing the endpoints of the interval corresponding to  $v$ .  $\mathcal{C}_{\text{PA}}$  and  $\mathcal{C}_{\text{IA}}$  will denote the subclasses of convex relations (also called continuous relations) of PA and IA respectively.  $\mathcal{C}_{\text{PA}}$  does not allow us to express inequations between points, it corresponds to the set  $2^{B_{\text{PA}}} \setminus \{\{\cdot < \cdot, \cdot >\}\}$ .  $\mathcal{C}_{\text{IA}}$  corresponds to those relations of IA belonging to  $\mathcal{P}$  one can translate into PA by relations of  $\mathcal{C}_{\text{PA}}$ .  $\mathcal{C}_{\text{IA}}$  contains 83 relations. In [6], Ligozat introduces a lattice (the interval lattice) arranging the base relations of  $B_{\text{IA}}$ , see Figure 1(b). The relations of  $\mathcal{C}_{\text{IA}}$  correspond to the intervals of this lattice. The ORD-Horn relations of IA (or preconvex relations) form a subclass denoted by  $\mathcal{H}$  in the sequel.  $\mathcal{H}$  is the maximal (for  $\subseteq$ ) subclass of IA containing singleton relations for which the consistency problem is polynomial. Ligozat attributes a dimension (an integer between 0 and 2) to each base relation of IA. The dimension of a base relation  $a \in B_{\text{IA}}$ , denoted by  $\dim(a)$ , corresponds to 2 minus the number of equalities of bounds of two intervals satisfying the base relation in consideration. Hence, the dimension of the base relations  $p, pi, o, oi, d, di$  is 2, the one of  $m, mi, s, si, f, fi$  is 1, and this of  $eq$  is 0. The dimension of a relation  $r \in 2^{B_{\text{IA}}}$  is the maximal dimension of its base relations:  $\dim(r) = \text{Max}\{\dim(a) : a \in r\}$ . A preconvex relation is a relation of IA such that its closure w.r.t. the class of the convex relations does not contain a new base relation with a dimension equals or greater than its dimension. To finalize these preliminaries, note that we have  $B_{\text{IA}} \subset \mathcal{C}_{\text{IA}} \subset \mathcal{P} \subset \mathcal{H}$  with  $B_{\text{IA}}$  the set of singleton relations of IA, *i.e.* the set  $\{\{b\} : b \in B_{\text{IA}}\}$ .

### 3 Global Consistency of the $\diamond$ -Consistent QCNs on $\mathcal{S}_{\text{PA}}$ and $\mathcal{S}_{\text{IA}}$

In this section, we will study two subclasses denoted by  $\mathcal{S}_{\text{PA}}$  and  $\mathcal{S}_{\text{IA}}$ .  $\mathcal{S}_{\text{PA}}$  corresponds to the relations of PA that do not allow us to express inequations. Formally,  $\mathcal{S}_{\text{PA}}$  is defined by  $\mathcal{S}_{\text{PA}} = \{\{\}, \{=\}, \{<\}, \{>\}, \{<, >\}, \{<, =, >\}\}$ , i.e. the relations of PA without the relations  $\{<, =\}$  and  $\{>, =\}$ .  $\mathcal{S}_{\text{IA}}$  is composed of those pointizable relations of IA one can translate into PA using relations of  $\mathcal{S}_{\text{PA}}$ . Formally, we can define the subclass  $\mathcal{S}_{\text{IA}}$  as follows:

**Definition 2.** A relation  $r \in 2^{\mathcal{B}_{\text{IA}}}$  belongs to  $\mathcal{S}_{\text{IA}}$  if and only if, we have :

- (a)  $\mathcal{C}_{\text{IA}}(r)$  is an interval  $[a, b]$  of the interval lattice with  $\dim(a) = \dim(b)$  ;
- (b)  $\mathcal{C}_{\text{IA}}(r) \setminus r = \bigcup E$ , with  $E$  a subset of  $\{\{m\}, \{mi\}, \{s, eq, si\}, \{f, eq, fi\}\}$ .

From this definition, we can easily translate a constraint  $x \, r \, y$ , with  $x, y$  are two interval variables and  $r \in \mathcal{S}_{\text{IA}}$ , into equivalent constraints between endpoints of  $x$  and  $y$  by using relations of  $\mathcal{S}_{\text{PA}}$ . First, we translate the constraint  $x \, \mathcal{C}_{\text{IA}}(r) \, y$  into PA by using strict inequalities or equalities. Then, we add inequations to remove base relations from relations of  $E$  belonging to  $\mathcal{C}_{\text{IA}}(r) \setminus r$ .  $\mathcal{S}_{\text{IA}}$  contains 82 relations. The relations belonging to  $\mathcal{S}_{\text{IA}}$  and  $\mathcal{C}_{\text{IA}}$  correspond exactly to the set  $\overline{\mathcal{B}_{\text{IA}}}$ . Hence, we have  $\overline{\mathcal{B}_{\text{IA}}} \subset \mathcal{S}_{\text{IA}} \subset \mathcal{P}$ ,  $\mathcal{B}_{\text{IA}} \subset \mathcal{C}_{\text{IA}} \subset \mathcal{P}$  and  $\mathcal{C}_{\text{IA}} \cap \mathcal{S}_{\text{IA}} = \overline{\mathcal{B}_{\text{IA}}}$ . For instance, the relation  $\{p, m, o\}$  belongs to the intersection of  $\mathcal{C}_{\text{IA}} \cap \mathcal{S}_{\text{IA}}$ , the relation  $\{p, o\}$  belongs to  $\mathcal{S}_{\text{IA}} \setminus \mathcal{C}_{\text{IA}}$  and  $\{p, m\}$  belongs to  $\mathcal{C}_{\text{IA}} \setminus \mathcal{S}_{\text{IA}}$ .

We are going to prove that the path-consistency method allows to obtain globally consistent QCNs for  $\mathcal{S}_{\text{PA}}$  and  $\mathcal{S}_{\text{IA}}$ .

**Proposition 1.** Let  $\mathcal{N} = (V, C)$  be a QCN of PA (resp. IA) and a subclass  $\mathcal{A}$  of PA (resp. IA) containing the total relation. If  $\mathcal{N}$  is  $\diamond$ -consistent then  $\mathcal{A}(\mathcal{N})$  is  $\diamond$ -consistent.

**Proof.** We have for all  $v, v', v'' \in V$ ,  $C(v, v') \subseteq C(v, v'') \diamond C(v'', v')$ . By monotonicity of  $\diamond$ ,  $C(v, v'') \diamond C(v'', v') \subseteq \mathcal{A}(C(v, v'')) \diamond \mathcal{A}(C(v'', v'))$ . Hence,  $C(v, v') \subseteq \mathcal{A}(C(v, v'')) \diamond \mathcal{A}(C(v'', v'))$ . As,  $\mathcal{A}(C(v, v'))$  is the smallest relation of  $\mathcal{A}$  including  $C(v, v')$  and  $(\mathcal{A}(C(v, v'')) \diamond \mathcal{A}(C(v'', v')) \in \mathcal{A}$ , we can conclude that  $\mathcal{A}(C(v, v')) \subseteq \mathcal{A}(C(v, v'')) \diamond \mathcal{A}(C(v'', v'))$ .  $\dashv$

**Theorem 1.** Let  $\mathcal{N}$  be a QCN of PA on  $\mathcal{S}_{\text{PA}}$ . We have  $\mathcal{N}$  is  $\diamond$ -consistent if and only if  $\mathcal{N}$  is globally consistent.

**Proof.** Let  $\mathcal{N}$  be a QCN defined on the subclass  $\mathcal{S}_{\text{PA}}$ . Obviously, if  $\mathcal{N}$  is globally consistent then  $\mathcal{N}$  is  $\diamond$ -consistent. Now, suppose that  $\mathcal{N}$  is  $\diamond$ -consistent, we will show that  $\mathcal{N}$  is globally consistent. Let  $\sigma$  be a partial solution of  $\mathcal{N}$  on  $V' \subseteq V$  with  $V' = \{v_0, \dots, v_i\}$ ,  $0 \leq i \leq |V| - 1$ . We are going to show that  $\sigma$  can be consistently extended to a variable  $v_{i+1} \in V \setminus V'$ .  $\mathcal{N}$  is  $\diamond$ -consistent, consequently  $\mathcal{N}$  is strongly 3-consistent. Hence, for the case  $i < 2$ , we can assert that  $\sigma$  can be extended to  $v_{i+1}$ . Now, consider the case  $i \geq 2$ . Examine the two following subcases :

- There exists  $v_j \in V'$  such that  $C(v_{i+1}, v_j) = \{=\}$ . Let  $\sigma(v_{i+1}) = \sigma(v_j)$ . As  $\mathcal{N}$  is  $\diamond$ -consistent, for any  $v_k \in V'$  we have  $C(v_j, v_k) \subseteq \{=\} \diamond C(v_{i+1}, v_k)$ . Hence, as  $\{=\} \diamond C(v_{i+1}, v_k) = C(v_{i+1}, v_k)$  we have  $C(v_j, v_k) \subseteq C(v_{i+1}, v_k)$ . On the other hand,  $(\sigma_j, \sigma_k)$  satisfies a base relation belonging to  $C(v_j, v_k)$ . Since  $\sigma(v_j) = \sigma(v_{i+1})$ ,

from all this we can assert that  $(\sigma(v_{i+1}), \sigma(v_k))$  satisfies a base relation belonging to  $C(v_{i+1}, v_k)$ . Hence,  $\sigma$  is extended to a partial solution on  $V' \cup \{v_{i+1}\}$ .

- It does not exist  $v_j \in V'$  such that  $C(v_{i+1}, v_j) = \{=\}$ . Let  $\mathcal{N}'$  be the QCN  $\mathcal{C}_{\text{IA}}(\mathcal{N})$ . From the previous proposition, we know that  $\mathcal{N}'$  is  $\diamond$ -consistent, moreover  $\mathcal{N}'$  is a convex QCN, hence  $\mathcal{N}$  is globally consistent [10]. For each  $v_k \in V'$  we define  $I_{v_k}$  by the interval  $\{u \in \mathbb{Q} : (u, \sigma(v_k)) \text{ satisfies } \mathcal{C}_{\text{IA}}(C(v_{i+1}, v_k))\}$ . Note that  $I_{v_k}$  is necessarily an open interval since  $\mathcal{C}_{\text{IA}}(C(v_{i+1}, v_k))$  is defined by one of the following relations:  $\{<, =, >\}, \{<\}, \{>\}$ . Let  $\mathcal{I}$  be the set  $\{I_{v_k} : v_k \in V'\}$ . We can assert that  $\sigma$  is a partial solution of  $\mathcal{N}'$  since  $C(v, v') \subseteq \mathcal{C}_{\text{IA}}(C(v, v'))$  for all  $v, v' \in V$ . Moreover, since  $\mathcal{N}'$  is globally consistent we can consistently extend  $\sigma$  to  $v_{i+1}$ . Consequently,  $\bigcap \mathcal{I} \neq \emptyset$ . On the other hand, as  $\mathcal{I}$  contains uniquely open intervals, we can assert that  $\bigcap \mathcal{I}$  is an open interval. Now consider the set of points  $\mathcal{E} = \{\sigma(v_k) : C(v_{i+1}, v_k) = \{<, >\}\}$ . By defining  $\sigma(v_{i+1})$  by a point belonging to  $(\bigcap \mathcal{I}) \setminus \mathcal{E}$  we can assert that  $(\sigma(v_{i+1}), \sigma(v_k))$  satisfies a base relation belonging to  $C(v_{i+1}, v_k)$  for all  $k \in V'$ . Hence, we can extend  $\sigma$  to a partial solution of  $\mathcal{N}$  on  $V' \cup \{v_{i+1}\}$ .  $\dashv$

**Proposition 2.** *Let  $\mathcal{N}$  be a QCN of IA on  $\mathcal{P}$ . If  $\mathcal{N}$  is  $\diamond$ -consistent then  $\text{PA}(\mathcal{N})$  is  $\diamond$ -consistent.*

**Proof.** By a program we have checked the property for all pointizable QCN defined on three variables. As three point variables are provided by at most three interval variables then the property holds for the general case.  $\dashv$

**Theorem 2.** *Let  $\mathcal{N}$  be a QCN of IA defined on  $\mathcal{S}_{\text{IA}}$ . If  $\mathcal{N}$  is  $\diamond$ -consistent then  $\mathcal{N}$  is globally consistent.*

**Proof.** Let  $\mathcal{N} = (V, C)$  be a  $\diamond$ -consistent QCN of IA on  $\mathcal{S}_{\text{IA}}$ . Let  $\mathcal{N}'$  be the QCN of PA defined by  $\mathcal{N}' = \text{PA}(\mathcal{N})$ .  $\mathcal{N}'$  is defined by relations of  $\mathcal{S}_{\text{PA}}$ . Moreover, from Proposition 2 we can assert that  $\mathcal{N}'$  is  $\diamond$ -consistent. Hence, from Theorem 1 we know that  $\mathcal{N}'$  is globally consistent. Since  $\mathcal{N}$  and  $\mathcal{N}'$  represent equivalent constraints, they have same solutions (through the link between points and intervals). From global consistency of  $\mathcal{N}'$  we can assert that each partial solution of  $\mathcal{N}$  on  $V' \subset V$  can be extended to  $v \in V \setminus V'$ .  $\dashv$

## 4 Maximality of $\mathcal{C}_{\text{IA}}$ and $\mathcal{S}_{\text{IA}}$ for $\diamond$ -Consistency and Global Consistency

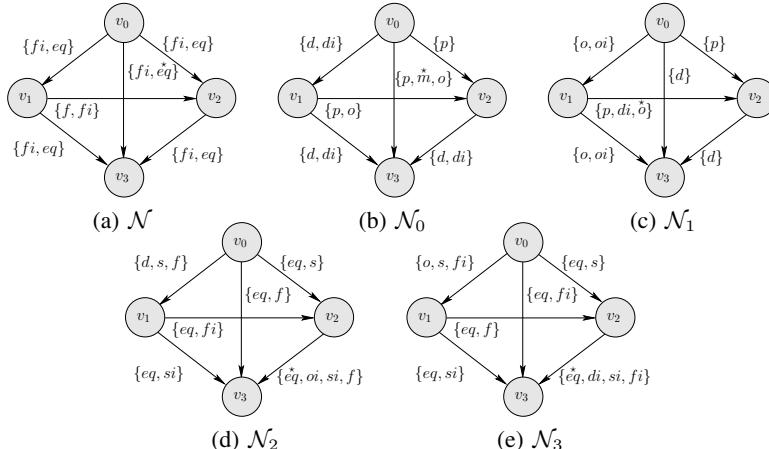
In this section, we will show that  $\mathcal{C}_{\text{IA}}$  and  $\mathcal{S}_{\text{IA}}$  are the only greatest subclasses of IA containing the singleton relations for which global consistency can be solved by the path-consistency method. For this, we will show that the strict extensions of  $\mathcal{C}_{\text{IA}}$  and  $\mathcal{S}_{\text{IA}}$  include subclasses for which  $\diamond$ -consistent QCN are not necessarily globally consistent. A meticulous analysis lead us to consider the seven following relations of IA :  $\rho_0 = \{d, di\}$ ,  $\rho_1 = \{o, oi\}$ ,  $\rho_2 = \{d, s, f\}$ ,  $\rho_3 = \{o, s, fi\}$ ,  $\rho_4 = \{o, oi, d, di, s, si, f, fi\}$ ,  $\rho_5 = \{o, oi, d, di, s, f, fi\}$  and  $\rho_6 = \{o, oi, d, di, s, si, fi\}$ . Note that  $\rho_0$  and  $\rho_1$  belong to  $2^{\mathcal{B}_{\text{IA}}} \setminus \mathcal{H}$  and  $\rho_2, \rho_3, \rho_4, \rho_5, \rho_6$  are relations of the set  $\mathcal{H} \setminus \mathcal{P}$ . For  $i \in \{0, \dots, 6\}$ , we define the subclass  $\mathcal{A}_i$  by  $\mathcal{A}_i = \overline{\mathcal{B}_{\text{IA}} \cup \{\rho_i\}}$ .

For each one of these subclasses, the minimal labeling problem cannot be solved or the property of global consistency cannot be reached by means of the path-consistency method. Indeed we have the following result:

**Proposition 3.** *Let  $\mathcal{N}$  be a  $\diamond$ -consistent QCN of IA. We have :*

- (a) *If  $\mathcal{N}$  is defined by relations belonging to a subclass belonging to the set  $\{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$  then  $\mathcal{N}$  is not necessarily a minimal QCN.*
- (b) *If  $\mathcal{N}$  is defined by relations belonging to the subclass  $\mathcal{A}_4, \mathcal{A}_5$  or  $\mathcal{A}_6$  then  $\mathcal{N}$  is not necessarily a globally consistent QCN.*

**Proof.** For each subclass  $\mathcal{A}$  of  $\{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ , Figure 2 shows a  $\diamond$ -consistent QCN whose constraints are defined by relations of  $\mathcal{A}$  and which is not minimal. For each subclass  $\mathcal{A}$  of  $\{\mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_6\}$ , Figure 3 depicts a  $\diamond$ -consistent QCN whose constraints are defined by relations of  $\mathcal{A}$  and which is minimal but not globally consistent. Let  $\sigma, \sigma'$  be the partial solutions:  $\sigma(v_0) = [1, 2], \sigma(v_2) = [2, 4], \sigma(v_3) = [1, 3], \sigma'(v_0) = [3, 4], \sigma'(v_2) = [1, 3], \sigma'(v_3) = [2, 4]$ .  $\sigma$  is a partial solution of  $\mathcal{N}_4$  and  $\mathcal{N}_5$  but cannot be extended to obtain a solution of  $\mathcal{N}_4$  and  $\mathcal{N}_5$ .  $\sigma'$  is a partial solution of  $\mathcal{N}_6$  which cannot be extended.  $\dashv$

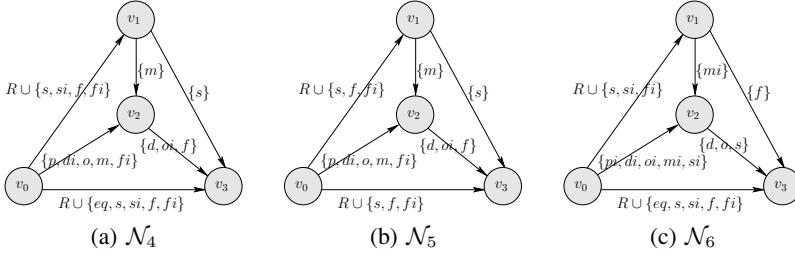


**Fig. 2.** Five  $\diamond$ -consistent and not minimal QCNs of IA.  $\mathcal{N}, \mathcal{N}_0, \mathcal{N}_1, \mathcal{N}_2$  and  $\mathcal{N}_3$  are respectively defined on  $\mathcal{P}, \mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$  and  $\mathcal{A}_3$ . Non feasible base relations are marked with a star.

By a program we can prove the following properties:

**Proposition 4.** *Let  $r, r_1, r_2, r_3, r_4, r_5 \in 2^{\mathcal{B}_{\text{IA}}}$  such that  $r \notin \mathcal{H}$ ,  $r_1 \in \mathcal{H} \setminus \mathcal{C}_{\text{IA}}$ ,  $r_2 \in \mathcal{H} \setminus \mathcal{S}_{\text{IA}}$ ,  $r_3 \in \mathcal{H} \setminus (\mathcal{C}_{\text{IA}} \cup \mathcal{S}_{\text{IA}})$ ,  $r_4 \in \mathcal{C}_{\text{IA}} \setminus \overline{\mathcal{B}_{\text{IA}}}$  and  $r_5 \in \mathcal{S}_{\text{IA}} \setminus \overline{\mathcal{B}_{\text{IA}}}$ . We have:*

- (a)  $\mathcal{A}_0 \subseteq \overline{\mathcal{B}_{\text{IA}} \cup \{r\}}$  or  $\mathcal{A}_1 \subseteq \overline{\mathcal{B}_{\text{IA}} \cup \{r\}}$ ;
- (b)  $\mathcal{A}_2 \subseteq \mathcal{C}_{\text{IA}} \cup \{r_1\}$  or  $\mathcal{A}_3 \subseteq \mathcal{C}_{\text{IA}} \cup \{r_1\}$  or  $\mathcal{P} \subseteq \overline{\mathcal{C}_{\text{IA}} \cup \{r_1\}}$ ;



**Fig. 3.** Three minimal  $\diamond$ -consistent and not globally consistent QCNs of IA.  $\mathcal{N}_4$ ,  $\mathcal{N}_5$  and  $\mathcal{N}_6$  are respectively defined on  $\mathcal{A}_4$ ,  $\mathcal{A}_5$  and  $\mathcal{A}_6$ . The relation  $R$  corresponds to  $\{o, oi, d, di\}$ .

- (c) at least one of the subclasses of  $\{\mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_6, \mathcal{C}_{IA}\}$  is included in  $\overline{\mathcal{B}_{IA} \cup \{r_2\}}$ ;
- (d)  $\mathcal{C}_{IA} \subseteq \overline{\mathcal{B}_{IA} \cup \{r_3\}}$  or  $\mathcal{S}_{IA} \subseteq \overline{\mathcal{B}_{IA} \cup \{r_3\}}$ ;
- (e)  $\overline{\mathcal{B}_{IA} \cup \{r_4, r_5\}} = \mathcal{P}$ .

**Theorem 3.** The subclass  $\mathcal{C}_{IA}$  is a maximal subclass among the subclasses for which  $\diamond$ -consistency implies the minimality property and is a maximal subclass among the subclasses for which  $\diamond$ -consistency implies global consistency.

**Proof.** We know that each  $\diamond$ -consistent QCN on  $\mathcal{C}_{IA}$  is globally consistent (and consequently minimal). We are going to show that for each strict extension of  $\mathcal{C}_{IA}$ , a  $\diamond$ -consistent QCN is not necessarily minimal (and consequently not necessarily globally consistent). Let  $r \in 2^{\mathcal{B}_{IA}} \setminus \mathcal{C}_{IA}$  and  $\mathcal{A}$  the subclass  $\mathcal{C}_{IA} \cup \{r\}$ . From Proposition 4 (properties (a) and (b)),  $\mathcal{A}$  includes at least one of the subclasses of the set  $\{\mathcal{P}, \mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ . From this and Proposition 3 (a), it follows that a  $\diamond$ -consistent QCN defined on  $\mathcal{A}$  is not necessarily minimal.  $\dashv$

**Theorem 4.** The subclass  $\mathcal{S}_{IA}$  is a maximal subclass among the subclasses for which  $\diamond$ -consistency implies global consistency.

**Proof.** We know that each  $\diamond$ -consistent QCN on  $\mathcal{S}_{IA}$  is globally consistent. We are going to show that for each strict extension of  $\mathcal{S}_{IA}$ , a  $\diamond$ -consistent QCN is not necessarily globally consistent. Let  $r \in 2^{\mathcal{B}_{IA}} \setminus \mathcal{S}_{IA}$  and  $\mathcal{A}$  the subclass  $\mathcal{S}_{IA} \cup \{r\}$ . From Proposition 4 (properties (a) and (c)),  $\mathcal{A}$  includes at least one of the subclasses of the set  $\{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_6\}$  (case 1) or the class  $\mathcal{C}_{IA}$  (case 2). If the case 1 holds, from Proposition 3, it follows that a  $\diamond$ -consistent QCN defined on  $\mathcal{A}$  is not necessarily globally consistent. Now, suppose that  $\mathcal{C}_{IA} \subseteq \mathcal{A}$ . Since  $\mathcal{S}_{IA} \subseteq \mathcal{A}$  and  $(\mathcal{S}_{IA} \setminus \mathcal{C}_{IA}) \neq \emptyset$  then  $\mathcal{C}_{IA} \subset \mathcal{A}$ . From Theorem 3, we can assert that a  $\diamond$ -consistent QCN on  $\mathcal{A}$  is not necessarily globally consistent.  $\dashv$

**Theorem 5.** The subclasses  $\mathcal{C}_{IA}$  and  $\mathcal{S}_{IA}$  are the unique greatest subclasses among the subclasses containing singleton relations of IA and for which  $\diamond$ -consistency implies global consistency.

**Proof.** Let  $\mathcal{A}$  be a subclass of IA such that  $\mathcal{B}_{IA} \subseteq \mathcal{A}$ ,  $\mathcal{C}_{IA} \not\subseteq \mathcal{A}$ ,  $\mathcal{S}_{IA} \not\subseteq \mathcal{A}$ ,  $\mathcal{A} \not\subseteq \mathcal{C}_{IA}$  and  $\mathcal{A} \not\subseteq \mathcal{S}_{IA}$  and  $\diamond$ -consistency implies global consistency. First, suppose that  $\mathcal{A} \subseteq (\mathcal{S}_{IA} \cup \mathcal{C}_{IA})$ . Hence, there exist  $r, r' \in \mathcal{A}$  such that  $r \in \mathcal{S}_{IA} \setminus \overline{\mathcal{B}_{IA}}$  and  $r' \in \mathcal{C}_{IA} \setminus \overline{\mathcal{B}_{IA}}$ . From Proposition 4(e) we have  $\mathcal{A} = \mathcal{P}$ . There is a contradiction since for  $\mathcal{P}$ ,  $\diamond$ -consistency does not imply global consistency. Hence, there exists a relation  $r \in \mathcal{A}$  such that  $r \notin (\mathcal{S}_{IA} \cup \mathcal{C}_{IA})$ . Suppose that  $r \notin \mathcal{H}$ , from Proposition 4(a) we can assert that  $\mathcal{A}$  includes  $\mathcal{A}_0$  or  $\mathcal{A}_1$ . From Proposition 3 it follows that a  $\diamond$ -consistent QCN defined on  $\mathcal{A}$  is not necessarily globally consistent. Now, suppose that  $r \in \mathcal{H}$ , from Proposition 4(d),  $\mathcal{C}_{IA} \subseteq \overline{\mathcal{B}_{IA} \cup \{r\}}$  or  $\mathcal{S}_{IA} \subseteq \overline{\mathcal{B}_{IA} \cup \{r\}}$ . Hence,  $\mathcal{C}_{IA} \subseteq \mathcal{A}$  or  $\mathcal{S}_{IA} \subseteq \mathcal{A}$ . There is a contradiction.  $\dashv$

Note that for PA, the maximality of  $\mathcal{C}_{PA}$  and  $\mathcal{S}_{PA}$  for the global consistency w.r.t the  $\diamond$ -consistency can be established in a direct way since the unique subclass of PA including (strictly)  $\mathcal{C}_{PA}$  or  $\mathcal{S}_{PA}$  is  $2^{\mathcal{B}_{PA}}$ .

## 5 Conclusion

We have studied the subclass  $\mathcal{S}_{PA}$  corresponding to the set of relations of the Point Algebra except the relations  $\{<, =\}$  and  $\{>, =\}$ , and the subclass  $\mathcal{S}_{IA}$  corresponding to the relations of the Interval Algebra one can express by mean of relations of  $\mathcal{S}_{PA}$ . We prove that for QCNs defined on  $\mathcal{S}_{PA}$  and  $\mathcal{S}_{IA}$  the path-consistency method allows us to obtain QCNs satisfying global consistency. Moreover, we show that  $\mathcal{S}_{IA}$  and  $\mathcal{C}_{IA}$  are the unique greatest subclasses of IA containing singleton relations for which path-consistency implies global consistency. The objective of this work is to identify all greatest subclasses of IA for which path-consistency implies global consistency or minimality. We will then consider the subclasses of IA that do not contain all singleton relations as in [2].

## References

- Allen, J.F.: An interval-based representation of temporal knowledge. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI 1981), pp. 221–226 (1981)
- Bessière, C., Isli, A., Ligozat, G.: Global consistency in interval algebra networks: Tractable subclasses. In: Proc. of the 12th European Conf. on Artificial Intelli. (ECAI 1996), pp. 3–7 (1996)
- Freuder, E.C.: Synthesizing constraint expressions. Commun. ACM 21(11), 958–966 (1978)
- Koubarakis, M.: From local to global consistency in temporal constraint networks. Theoretical Computer Science 173(1), 89–112 (1997)
- Krokhin, A.A., Jeavons, P., Jonsson, P.: Reasoning about temporal relations: The tractable subalgebras of allen’s interval algebra. Journal of ACM 50(5), 591–640 (2003)
- Ligozat, G.: A New Proof of Tractability for ORD-Horn Relations. In: Proc. of the Thirteenth National Conference on Artificial Intelligence (AAAI 1996), vol. 1, pp. 395–401 (1996)
- Ligozat, G., Renz, J.: What Is a Qualitative Calculus? A General Framework. In: Zhang, C., Guesgen, H.W., Yeap, W.-K. (eds.) PRICAI 2004. LNCS (LNAI), vol. 3157, pp. 53–64. Springer, Heidelberg (2004)
- Nebel, B., Bürkert, H.J.: Reasoning About Temporal Relations: A Maximal Tractable Subclass of Allen’s Interval Algebra. Journal of the ACM 42(1), 43–66 (1995)

9. Nökel, K. (ed.): Temporally Distributed Symptoms in Technical Diagnosis. LNCS, vol. 517, pp. 1–184. Springer, Heidelberg (1991)
10. van Beek, P.: Reasoning About Qualitative Temporal Information. Artificial Intelligence 58(1-3), 297–326 (1992)
11. Van Beek, P., Cohen, R.: Exact and approximate reasoning about temporal relations. Computational Intelligence 6, 133–144 (1990)
12. Vilain, M., Kautz, H.: Constraint Propagation Algorithms for Temporal Reasoning. In: Proceedings of the Fifth National Conf. on Artificial Intelligence (AAAI 1986), pp. 377–382 (1986)
13. Vilain, M., Kautz, H., van Beek, P.: Constraint propagation algorithms for temporal reasoning: a revised report. In: Qualitative Reasoning about Physical Systems, pp. 372–381 (1990)

# Rule Quality Measure-Based Induction of Unordered Sets of Regression Rules

Marek Sikora<sup>1,2</sup>, Adam Skowron<sup>1</sup>, and Łukasz Wróbel<sup>1</sup>

<sup>1</sup> Institute of Computer Science, Silesian University of Technology,  
ul. Akademicka 16, 44-100 Gliwice, Poland

{Marek.Sikora,Adam.Skowron,Lukasz.Wrobel}@polsl.pl

<sup>2</sup> Institute of Innovative Technologies EMAG,  
ul. Leopolda 31, 40-189 Katowice, Poland

**Abstract.** This paper presents the algorithm for induction of unordered sets of regression rules. It uses sequential covering strategy and dynamic reduction to classification approach. The main focus is put on quality measures which control the process of rule induction. We examined the effectiveness of nine quality measures. Moreover, we propose and compare three schemes of the prediction of target attribute value of examples covered by more than one rule. We also show rule filtration algorithm for the reduction of the number of generated rules. All experiments were carried out on 35 benchmark datasets.

**Keywords:** rule-based regression, rule quality measures, rule induction, rule filtration, prediction conflicts resolving.

## 1 Introduction

One of the most popular knowledge discovery methods applied for pattern recognition in data is rule induction [1–6]. So-called decision rules are the special kind of rules. Rule sets built by induction algorithms are usually designed for two basic aims. One aim is the developing a classification system [2, 4, 6]. The other aim is extraction of regularities hidden in the analyzed dataset [1, 3, 5]. Rules are employed for descriptive purposes (description based rule induction, subgroup discovery) because of their clearness.

Analyzing results obtained by various rule induction algorithms one can safely state that description and prediction abilities of determined rules depend not only on the algorithm of searching for rules but also on a measure evaluating quality of the induced rules. Quality measures are the most often applied in the rule growing and pruning phases. The quality measure applied in a rule induction algorithm is important for quality of an output rule set. This is confirmed by numerous empirical researches [7–14]. Another implication of these studies is that, depending on characteristics of the analyzed dataset, various measures may lead to various (statistically significant) results. In the paper [14] Sikora and Wróbel evaluated the performance of 30 quality measures on 34 datasets

and identified a set of nine measures the most effective according to classification and descriptive abilities of the induced rules.

Good descriptive and classification performance given by sequential covering rule induction algorithms has led a few authors to apply these algorithms to solve regression problems [15–22]. Application of simple regression rules which conclusions indicate a specific value of the decision attribute is particularly useful in situations where beside the small error of the prediction a description of the data is also important.

The purpose of the paper is an adaptation of the sequential covering rule induction algorithm and the verification of the effectiveness of the quality measures used for decision rules evaluation to control the process of the regression rules induction. The next aim is to verify the effectiveness of the three schemes for predicting target attribute values of examples covered by more than one rule. Analysis of the measure efficiency will be carried out in view of: the accuracy of prediction, the number of induced rules and their average coverage. Finally, we present a rule filtration algorithm which reduces the number of rules in order to improve readability of rule-based data model. In the analysis nine quality measures from the paper [14] and 35 benchmark datasets were used.

## 2 Related Work

The simplest approach to induction of regression rules is discretization of a continuous decision attribute and use standard decision rule induction algorithms. Such an approach is presented by Torgo and Gama [15], they transform the continuous decision attribute into a set of intervals using three methods: equal-frequency, equal-width and k-means clustering.

Regression rules can be also easily generated from regression trees such as CART [18] and M5 [16]. The straightforward approach is to generate one rule for each path from the root to the leaf of the tree. The other is to use divide-and-conquer strategy as in M5Rules algorithm [17].

The most computationally advanced methods of regression rule induction are based on ensemble techniques. Algorithms such as RuleFit [19] or RegENDER [20] lead the rule induction towards minimization of the loss function calculated on the training or validation set of examples. To supervise the induction of subsequent rules these algorithms apply various methods of optimization (e.g. gradient methods). The effects of their application are usually numerous sets of rules characterized by good quality of prediction.

Within our interest are methods of regression rule induction which use sequential covering (known also as separate-and-conquer) strategy. The rule induction is carried out to cover all training examples by rules with high quality. Algorithms implementing such a strategy are, among others, FRS [21] and PCR [23]. For the results presented in this paper particularly important is the Janssen and Fürnkranz work [22] in which they described the dynamic method of identification of positive and negative examples covered by induced regression rule. This method is described in the next section.

### 3 Induction of Regression Rules

The regression rule with constant conclusion is a logical expression in the following form (II):

$$\text{if } w_1 \text{ and } w_2 \text{ and } \dots \text{ and } w_n \text{ then } v \quad (1)$$

An object covered by the rule (i.e. satisfying all its elementary conditions  $w_i$ ) is assigned a real value  $v$  from conclusion. Each of the elementary conditions can be denoted as  $a_i \text{ op } Z_i$ , where  $a_i$  is the conditional attribute,  $\text{op}$  is one of the relation symbols from the set  $\{>, \geq, <, \leq, =\}$  and  $Z_i$  is a numerical or nominal value from the range of the attribute  $a_i$ .

Table II presents nine selected rule quality measures. They are defined on the basis of the values  $p$ ,  $n$ ,  $P$  and  $N$ , where  $p$  denotes the number of positive examples covered by the rule,  $P$  is number of all positive examples,  $n$  stands for the number of negative examples covered by the rule and  $N$  is the number of all negative examples. These measures are designed for evaluation of decision rules, however dynamic reduction to classification [22] approach, which we use in our implementation of the rule induction algorithm, allows to apply them in induction of regression rules. In accordance with this approach, the conclusion  $v$  is dynamically calculated on the basis of all examples covered by the rule as the median of target value of these examples. The example is labeled as positive if its target value is in the range  $v \pm sd$ , where  $sd$  denotes the standard deviation of target attribute of examples covered by the rule. The example is negative if its target value is outside of this range. It should be noted that the values  $p$ ,  $n$ ,  $P$  and  $N$  are calculated for each rule separately and differ between rules.

**Table 1.** Selected rule quality measures

Name and formula	
$C1 = Coleman \cdot \left( \frac{2+Cohen}{3} \right)$	$C2 = Coleman \cdot \left( \frac{P+p}{2P} \right)$
$\text{Correlation (Corr)} = \frac{pN-Pn}{\sqrt{(PN(p+n)(P-p+N-n))}}$	$g\text{-measure (}g, g=2\text{)} = \frac{p}{p+n+2}$
$s\text{-Bayesian confirmation (}s\text{)} = \frac{p}{p+n} - \frac{P-p}{P-p+N-n}$	$\text{Logical sufficiency (}LS\text{)} = \frac{pN}{nP}$
$\text{Rule specificity and sensitivity (RSS)} = \frac{p}{P} - \frac{n}{N}$	$\text{Mutual support (}MS\text{)} = \frac{p}{n+P}$
$\text{Weighted Laplace (}wLap\text{)} = \frac{(p+1)(P+N)}{(p+n+2)P}$	
where $Cohen = \frac{(P+N)\left(\frac{p}{p+n}\right)-P}{\left(\frac{P+N}{2}\right)\left(\frac{p+n+P}{p+n}\right)-P}$ and $Coleman = \frac{Np-Pn}{N(p+n)}$	

As it was shown in the papers [13, 14], the application of the measures from Table II to control the process of decision rule induction allows to obtain classifiers with high quality. Good results obtained by these measures in induction of classification rules, motivated us, therefore, to the verification of these measures in the problem of regression rule induction.

Our implementation of regression rule induction algorithm is based on a sequential covering procedure and uses dynamic reduction to classification approach [22]. The inputs of the algorithm are: training set of examples and the

rule quality measure. The outcome is a set of regression rules covering the input set of examples.

The process of a single rule induction is divided into two phases: growing and pruning. In the growing phase, the rule is built by the successive addition of elementary conditions. First elementary condition is added using entire set of training examples. Further conditions are added to the rule based on a set of examples covered by the conjunction of conditions which has been already added to the rule. In the case of nominal attributes the elementary condition can take the form of  $(a = q)$ , and for the numerical attributes it can take one of two forms:  $(a < q)$  or  $(a \geq q)$ . For the numerical attributes the value  $q$  is the arithmetic mean between two successive values from the range of attribute  $a$ . Each of the possible elementary condition is temporary added to the rule, and such a rule is evaluated using the selected quality measure. The condition for which rule obtains the highest value of the quality measure is selected as the final one.

In order to prevent induction of too specific rules, the candidate condition is only considered if rule with such a condition covers at least a fixed number of examples which are not covered by rules generated so far. A similar overfitting-avoidance mechanism is often incorporated in various implementations of tree and rule learners in the form of the parameter specifying the minimum number of examples per leaf/rule [24]. Naturally, the value of this parameter often requires an adjustment to a particular dataset, however in our experimental study, in which we compare algorithms on multiple datasets, we set this number to 3 what seems to be reasonable minimum for the calculation of median and standard deviation of the rule conclusion. Moreover, we assumed that the missing attribute values can not be covered by any elementary condition, therefore, if the example has unknown values then it can be covered only by the rule which does not contain attributes which values are unknown for this example.

The growth stage ends when the rule does not cover any negative examples or when the addition of the next elementary condition does not change the set of examples covered by the rule. Next, the rule is pruned. The aim of the pruning phase is further improvement of the rule according to the value of specified quality measure. The rule pruning uses a hill climbing strategy. At each iteration, it deletes the elementary condition without which the rule has the highest improvement in the value of quality measure. After each removal the value of rule quality measure is recalculated (on the entire set of training examples) and the deletion of conditions is repeated until it does not cause decrease in the current value of rule quality measure. After the pruning phase, the examples covered by created rule are removed from the set of uncovered examples, the rule is added to the final set of rules and the process of rule creating is repeated until the set of uncovered examples is not empty. The general outline of the algorithm can be represented by Algorithm 1.

In our implementation of rule induction algorithm, rules are returned in the form of unordered set. In contrast to systems with ordered rules, where the expected value is predicted based on the first matching rule, in our application

**Algorithm 1.** Pseudocode of the regression rule induction algorithm

---

```

function RULEINDUCTION(examples, ruleQualityMeasure)
    ruleSet  $\leftarrow \emptyset$ 
    uncoveredExamples  $\leftarrow \text{examples}$ 
    while uncoveredExamples  $\neq \emptyset$  do
        rule  $\leftarrow \text{Grow}(\text{examples}, \text{uncoveredExamples}, \text{ruleQualityMeasure})$ 
        rule  $\leftarrow \text{Prune}(\text{rule}, \text{examples}, \text{ruleQualityMeasure})$ 
        covered  $\leftarrow \text{Covered}(\text{rule}, \text{examples})$ 
        uncoveredExamples  $\leftarrow \text{uncoveredExamples} \setminus \text{covered}$ 
        ruleSet  $\leftarrow \text{ruleSet} \cup \{\text{rule}\}$ 
    end while
    return ruleSet
end function

```

---

we used an approach in which an example can be covered by several rules at the same time. Therefore, a method for resolving conflicts of prediction between rules is needed. In this paper we propose three complementary such methods referred to as: *mean of conclusions*, *median of covered* and *max rule quality*.

In the *mean of conclusions* method the expected value is predicted as the average value of all conclusions of conflicting rules. This method, however, treats all rules equally what is not appropriate when there is a large disproportion in the number of examples covered by conflicting rules. For the sake of argument, suppose that we have 2 conflicting rules – one which covers 1000 training examples with target value equal  $x$  and second which covers only 10 training examples with target value equal  $y$ . The greater distance between  $x$  and  $y$  the greater might be the prediction error. In order to overcame this problem the *median of covered* method was proposed. It predicts the median of target attribute of the union of examples covered by conflicting rules. The third, *max rule quality* method is strongly associated with the selected quality measure. It selects the conclusion of rule which has the highest value of quality measure used in induction.

In order to improve the clarity of the rule-based model and to speeds up the prediction we postprocess obtained rules with the use of rule filtration algorithm. It removes rules covering similar sets of training examples and works in nearly the same way as *Coverage* algorithm [13, 14] defined for filtration of decision rules. To put it briefly, it builds the coverage of a training set of examples using the ranking of the rules established by the selected quality measure. The rule is removed if a set of positive examples covered by this rule is a subset of the union of positive examples covered by the rules with a higher quality.

## 4 Experiments

The efficiency of nine selected rule quality measures from Table I and proposed methods of prediction conflict resolving were verified on 35 publicly available benchmark datasets known as: auto93, auto-mpg, auto-price, baseball, bodyfat, breasttumor, cholesterol, cloud, compressive, concrete, cpu, dee, diabetes,

echomonths, ele-1, ele-2, elusage, fishcatch, friedman, fruitfly, housing, kidney, laser, lowbwt, machine, mbagrade, meta, pbc, pharynx, pollution, pyrim, sensory, strike, triazines, veteran. The datasets differ in the number of examples and attributes as well as in the number of distinct values of the target attribute.

Each algorithm was tested with the use of single 10-fold cross validation. All presented results (Table 2, Table 3 and Figure 1) are an average over 35 analyzed datasets. The main evaluation criterion was root relative-squared error (2):

$$RRSE = 100\% \cdot \sqrt{\frac{(t_1 - p_1)^2 + \dots + (t_n - p_n)^2}{(t_1 - \bar{t})^2 + \dots + (t_n - \bar{t})^2}} \quad (2)$$

where  $n$  is the number of examples in the test set,  $t_1, \dots, t_n$  are target values of test examples,  $p_1, \dots, p_n$  are predicted values and  $\bar{t}$  is the mean value of target attribute over training dataset. The  $RRSE$  calculates total squared error relative to the error which is made by simple prediction using the mean value of target attribute of training examples. The lower value of  $RRSE$  the better is the algorithm. The  $RRSE$  is expressed as percentage what makes it suitable for evaluation of algorithms over multiple datasets.

Table 2 presents an average values of  $RRSE$  with corresponding ranks (lower is better) of three proposed methods of conflict resolving: *mean of conclusions*, *median of covered* and *max rule quality*. The comparison was made separately for each measure. Additionally, the average number of conflicting rules per test example is given. Depending on the rule quality measure applied in rule induction, there are on average 2-3 rules involved in the prediction of target value of a single test example. A Friedman test with Nemenyi post-hoc test (as proposed in [25]) were used in order to compare the methods of conflict resolving within each rule quality measure. The Friedman test indicates significant differences ( $p$ -value  $< 0.01$  for all considered measures) between compared techniques and shows that the method of conflict resolving is of great importance for the accuracy of predictions. According to Nemenyi test at 0.05 significance level, the method which predicts mean value of conclusions clearly outperforms the other ones. The winning method has the lowest average ranks for each tested quality measure, and its average ranks differ more than critical distance  $CD = 0.56$  from the ranks of other two methods for each measure except Mutual Support. The method which predicts the mean value of conclusions of conflicting rules is, therefore, recommended as default one and it is used in further part of this experimental study.

Table 3 presents more detailed results for selected rule quality measures and the *mean of conclusions* method. The table shows the average  $RRSE$ , number and coverage of rules as well as the influence of proposed filtration algorithm on these criteria. The coverage of the rule is calculated as a ratio of the number of training examples covered by the rule to the number of all training examples.

For comparison purposes, the results obtained by M5Rules from Weka package [24] and RegENDER [26] algorithms are also given in Table 3. The M5Rules algorithm was run with -R parameter which causes that generated rules have constant, instead of linear model, in conclusion (as in our implementation). For

**Table 2.** Average *RRSE* with corresponding ranks of three proposed methods of conflicts resolving for each selected rule quality measure on 35 benchmark datasets. The average number of conflicting rules per test example is also given (*conflicts* column).

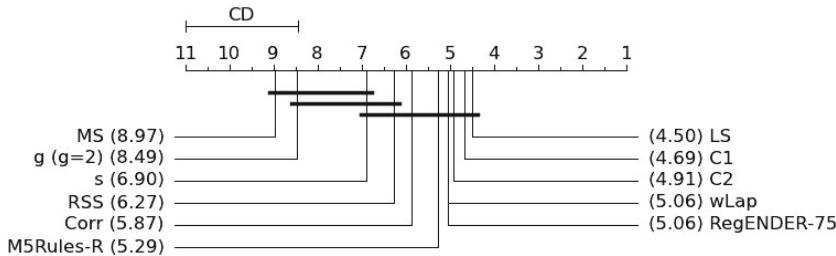
measure	conflicts	mean of concl		median of cov		max rule qual	
		RRSE	rank	RRSE	rank	RRSE	rank
C1	3.08	73.31	1.31	77.34	2.2	77.68	2.49
C2	3.26	74.77	1.17	78.54	2.26	79.64	2.57
Corr	3.55	77.65	1.29	82.05	2.29	81.82	2.43
g (g=2)	2.85	88.61	1.26	92.29	2.47	91.61	2.27
LS	2.54	72.53	1.31	74.86	2.14	76.95	2.54
MS	1.94	90.78	1.6	94.43	2.37	92.4	2.03
RSS	3.36	78.5	1.31	82.67	2.23	82.02	2.46
s	2.55	82.15	1.29	88.34	2.59	85.17	2.13
wLap	2.6	73.77	1.46	76.69	2.14	79.03	2.4

**Table 3.** Average *RRSE*, number of rules and coverage of rules for selected quality measures (without and with filtration), M5Rules and RegENDER on 35 benchmark datasets.

measure	without filtration			with filtration		
	RRSE	rules	cov	RRSE	rules	cov
C1	73.31	49.74	0.11	73.46	40.35	0.12
C2	74.77	37.34	0.15	74.88	28.72	0.17
Corr	77.65	20.01	0.24	77.93(-)	13.86	0.27
g (g=2)	88.61	14.76	0.29	88.63	13.17	0.30
LS	72.53	72.66	0.07	72.93	64.46	0.07
MS	90.78	5.03	0.56	90.98(-)	3.64	0.65
RSS	78.5	14.48	0.29	78.48	11.31	0.31
s	82.15	28.99	0.27	82.06	23.16	0.28
wLap	73.77	75.03	0.06	73.25	70.11	0.06
M5Rules-R	77.3	8.92				
RegENDER-75	76.85	75				

RegENDER, the parameter of the number of output rules was set to 75 what corresponds to the average number of rules induced by Weighted Laplace measure which, on average, gives the largest set of rules. The rest of the parameters of these algorithms were set to their default values.

The results presented in Table 3 show that rule-based models composed of various number of rules and with different prediction abilities can be obtained depending on the applied rule quality measure. The Friedman test shows statistical significant difference ( $p\text{-value} < 10^{-6}$ ) in the values of *RRSE* for the group of all eleven algorithms from Table 3. The differences are visualized in Figure 11 with the use of Nemenyi post-hoc test at 0.05 significance level. The figure presents CD diagram of nine rule quality measures without filtration together with M5Rules and RegENDER algorithms. The number in parenthesis, next to the name of the algorithm, shows its average rank according to *RRSE*.



**Fig. 1.** Comparison of all quality measures without filtration, M5Rules and RegENDER against each other with the Nemenyi test according to *RRSE* criterion. Groups of algorithms that are not significantly different at 0.05 level are connected.

The lowest rank was obtained by Logical Sufficiency, however there is no statistical difference between Logical Sufficiency and other algorithms except Mutual Support and  $g$ -measure ( $g=2$ ). It can be observed that the lowest ranks were obtained by the measures which induce greater number of rules at the cost of lower average coverage (C1, C2, Logical Sufficiency, Weighted Laplace). However, taking the number of rules and prediction accuracy at the same time into account, the Correlation and RSS measures seem to be the best choice.

An application of filtration algorithm to the rules obtained by each of the selected quality measures leads to smaller sets of rules and slightly increases their average coverage. According to the Wilcoxon signed-ranks test the decrease in the number of rules is always statistically significant ( $p$ -value  $< 10^{-5}$  for each examined rule quality measure). The sign (-) next to *RRSE* with filtration means that *RRSE* of the filtered set of rules is on average worse (the Wilcoxon test at 0.05 significance level) than for the set not subject to filtration. Statistically significant degradation in *RRSE* can be observed only for two measures: Correlation ( $p$ -value = 0.0326) and Mutual Support ( $p$ -value = 0.0053). For other measures there is no statistical difference in *RRSE* after filtration. The proposed rule filtration algorithm is, therefore, good choice for reduction of the number of rules.

## 5 Conclusions and Future Work

We presented the algorithm for induction of unordered sets of regression rules. The algorithm is based on the dynamic reduction to classification approach and it uses sequential covering strategy. The process of rule induction is controlled by rule quality measure which is used in the growing and pruning phases of the algorithm. We examined nine rule quality measures which were identified in the paper [14] as the measures the most effective according to classification and descriptive abilities of the induced rules.

As it was shown in the experimental part o this study, the choice of quality measure affects the size of sets of rules as well as the accuracy of predictions. It

can be observed that the lowest values of  $RRSE$  was obtained by C1, C2, Logical Sufficiency and Weighted Laplace measures. These measures have tendency to generate larger sets of rules at the cost of decreased average coverage. Interestingly enough, the Nemenyi test does not prove significant differences between the best seven (according to the value of  $RRSE$ ) measures. However, it should be noted that the Nemenyi test is very restrictive and it rarely indicates that one algorithm wins with all the others [26]. Nevertheless, taking the number of rules and the results of the Nemenyi test into account, the Correlation and RSS measures seem to be a good compromise between the value of  $RRSE$  and the number of rules.

We also presented and tested three methods to resolve conflicts between rules during the prediction of target value of test example and one method of rule filtration. The best method to resolve conflicts proved to be the simple averaging of conclusions of rules matching the example. The proposed filtration algorithm enabled us to reduce the size of the rule sets and keep prediction abilities at the same time.

Our future efforts will focus on several directions. A promising path of development should be research on filtration algorithms that lead to smaller size of sets and does not affect much on  $RRSE$ . Other way should be the verification of the use of the approach of data-driven adaptive selection of rule quality measure as was presented for the problem of classification [13, 14]. We also noticed that most of used quality measures generally behave similarly as in classification task [13, 14], but to prove this trend further researches on more quality measures are needed.

**Acknowledgements.** This work was supported by the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09).

## References

1. Kryszkiewicz, M.: Fast Discovery of Representative Association Rules. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 214–221. Springer, Heidelberg (1998)
2. Fürnkranz, J.: Separate-and-conquer rule learning. Artificial Intelligence Review 13, 3–54 (1999)
3. Stefanowski, J., Vanderpoorten, D.: Induction of decision rules in classification and discovery-oriented perspectives. Int. J. Intell. Syst. 16(1), 13–27 (2001)
4. Grzymala-Busse, J.W., Ziarko, W.: Data mining based on rough sets. In: Data Mining: Opportunities and Challenges, pp. 142–173. IGI Global (2003)
5. Wojtusiak, J., Michalski, R.S., Kaufman, K.A., Pietrzykowski, J.: The AQ21 natural induction program for pattern discovery: Initial version and its novel features. In: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, pp. 13–15 (2006)
6. Fawcett, T.: PRIE: a system for generating rulelists to maximize ROC performance. Data Mining and Knowledge Discovery 17(2), 207–224 (2008)
7. An, A., Cercone, N.: Rule quality measures for rule induction systems: Description and evaluation. Computational Intelligence 17(3), 409–424 (2001)

8. Bruha, I., Tkadlec, J.: Rule quality for multiple-rule classifier: Empirical expertise and theoretical methodology. *Intell. Data Anal.* 7(2), 99–124 (2003)
9. Brzezinska, I., Greco, S., Slowinski, R.: Mining pareto-optimal rules with respect to support and confirmation or support and anti-support. *Eng. Appl. Artif. Intell.* 20(5), 587–600 (2007)
10. Janssen, F., Fürnkranz, J.: On the quest for optimal rule learning heuristics. *Machine Learning* 78(3), 343–379 (2010)
11. Sikora, M.: Rule Quality Measures in Creation and Reduction of Data Rule Models. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Slowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 716–725. Springer, Heidelberg (2006)
12. Sikora, M.: Decision Rule-Based Data Models Using TRS and NetTRS – Methods and Algorithms. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets XI. LNCS, vol. 5946, pp. 130–160. Springer, Heidelberg (2010)
13. Sikora, M., Wróbel, L.: Data-Driven Adaptive Selection of Rules Quality Measures for Improving the Rules Induction Algorithm. In: Kuznetsov, S.O., Ślezak, D., Hepting, D.H., Mirkin, B.G. (eds.) RSFDGrC 2011. LNCS, vol. 6743, pp. 278–285. Springer, Heidelberg (2011)
14. Sikora, M., Wróbel, L.: Data-driven adaptive selection of rule quality measures for improving rule induction and filtration algorithms. *International Journal of General Systems* (to appear, 2012)
15. Torgo, L., Gama, J.: Regression by Classification. In: Borges, D.L., Kaestner, C.A.A. (eds.) SBIA 1996. LNCS, vol. 1159, pp. 51–60. Springer, Heidelberg (1996)
16. Quinlan, R.J.: Learning with continuous classes. In: 5th Australian Joint Conference on Artificial Intelligence, pp. 343–348. World Scientific, Singapore (1992)
17. Holmes, G., Hall, M., Frank, E.: Generating Rule Sets from Model Trees. In: Foo, N.Y. (ed.) AI 1999. LNCS, vol. 1747, pp. 1–12. Springer, Heidelberg (1999)
18. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth (1984)
19. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 916–954 (2008)
20. Dembczyński, K., Kotłowski, W., Slowiński, R.: Solving Regression by Learning an Ensemble of Decision Rules. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 533–544. Springer, Heidelberg (2008)
21. Demšar, D.: Obravnavanje numeričnih problemov z induktivnim logičnim programiranjem. Master's thesis, Faculty of Computer and Information Science, University of Ljubljana, Slovenia, 59, 64 (1999) (in Slovene)
22. Janssen, F., Fürnkranz, J.: Heuristic rule-based regression via dynamic reduction to classification. In: Walsh, T. (ed.) Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), pp. 1330–1335 (2011)
23. Ženko, B., Džeroski, S., Struyf, J.: Learning Predictive Clustering Rules. In: Bonchi, F., Boulicaut, J.-F. (eds.) KDID 2005. LNCS, vol. 3933, pp. 234–250. Springer, Heidelberg (2006)
24. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann, Amsterdam (2011)
25. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7, 1–30 (2006)
26. Garcia, S., Herrera, F.: An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *Journal of Machine Learning Research* 9, 2677–2694 (2008)

# Swarm Capability of Finding Eigenvalues

Jacek Kabziński

Institute of Automatic Control, Technical University of Lodz,  
Stefanowskiego 18/22 90-924 Lodz, Poland  
[jacek.kabzinski@p.lodz.pl](mailto:jacek.kabzinski@p.lodz.pl)

**Abstract.** We consider particle swarm optimization algorithm with a constriction coefficient and investigate its competence in finding matrix eigenvalues. We propose a new objective function formulation and several slight algorithm modifications to compute eigenvalues effectively. We compare obtained results with standard numerical procedures for eigenvalues calculation and illustrate presented critical analysis with numerical examples. The problem generates interesting and challenging objective function landscapes, so it is a good benchmark to test abilities of PSO variants.

**Keywords:** global optimization, particle swarm optimization, eigenvalues.

## 1 Introduction

Since the pioneer work by Eberhart and Kennedy [1] introducing particle swarm optimization (PSO), hundreds of researchers have proved its high capability to cope with non-convex, non-smooth, multimodal optimization problems. The algorithm was investigated extensively and now we may conclude that a kind of contemporary standard PSO is established and widely accepted. The purpose of this contribution is to investigate standard PSO algorithm capability to solve numerical eigenvalue problem for square, real, nonsymmetrical matrices. The problem generates interesting and challenging objective function landscapes. Several adjectives may be used to describe the observed phenomena – multi-modal, nondifferentiable, multi-funnel, with misleading information, needle-in-a-haystack. That's why finding eigenvalues is a good benchmark to test abilities of PSO variants.

Recently several publications addressed numerical analysis applications of PSO, also for finding matrix eigenvalues [2,3,4]. We propose another approach to objective function selection and investigate some techniques to improve algorithm efficiency.

The eigenproblem has been very wildly investigated since early 19<sup>th</sup> century and we have to compare PSO performance with well established and professionally implemented numerical methods. For example in Matlab (the most popular software in many engineering and other applications) versions of QR method taken from LAPACK [5] are used. Precisely, for nonsymmetrical matrices it is routine SGEEV (with scaling factor SCLFAC = 2 instead of 8 which is default in LAPACK's SGEBAL), or if we disable scaling (command `eig(A, 'nobalance')`), routines SGEHRD and SHSEQR are used [5]. All these methods work satisfactory and are

commonly accepted as ‘state of art’ among professionals, but severe difficulties are reported in case of multiple eigenvalues, high-dimensional problems and badly conditioned matrices. Therefore the constant tendency to look for better solutions is justified.

## 2 Unified, Constricted PSO

We consider one the most popular versions of particle swarm optimization algorithm wildly known as UPSO – unified particle swarm optimization [6].

Potential solutions are represented by  $S$  numbered individuals or particles –  $M$ -dimensional vectors  $x_i \quad i = 1, 2, \dots, S$ , whose size is the number of degrees of freedom of the optimization problem. For each particle we define its neighborhood: for the  $i$ -th particle its neighborhood consists of the particles numbered  $i - r, i - r + 1, \dots, i, \dots, i + r$  and  $r$  is called neighborhood radius. A population of particles is initialized with random positions  $x_i(0)$  and velocities  $v_i(0)$  in the search space.

At the  $t$ -th iteration particles are evaluated by the objective function  $F(x)$  and the following values are memorized:  $p_i(t)$  - the best position of the  $i$ -th particle,  $i = 1, 2, \dots, S$  during iterations  $1, 2, \dots, t$  and  $p_G(t)$  – the global best position, found so far by the whole swarm,  $p_{Li}(t)$  – the local best position, found until now by members of the  $i$ -th particle neighborhood. Next, position update (‘velocity’) components are build for each particle  $i = 1, 2, \dots, S$ :

$$v_{Gi}(t+1) = \chi[v_i(t) + c_1 R_{Gi1} \circ (p_i(t) - x_i(t)) + c_2 R_{Gi2} \circ (p_G(t) - x_i(t))] \quad (1)$$

$$v_{Li}(t+1) = \chi[v_i(t) + c_1 R_{Li1} \circ (p_i(t) - x_i(t)) + c_2 R_{Li2} \circ (p_{Li}(t) - x_i(t))] \quad (2)$$

where:  $\chi$  - is the constriction coefficient,  $c_1, c_2$  - are cognitive and social (acceleration) parameters,  $v_i(t)$  - is the previous velocity of the  $i$ -th particle,  $R_{Gi1}, R_{Gi2}, R_{Li1}, R_{Li2}$  - are random  $M$ -dimensional vectors, whose components are uniformly distributed in  $[0, 1]$  and  $\circ$  denotes component-wise multiplication.

The new velocity and the position of the particle are calculated as

$$v_i(t+1) = uv_{Gi}(t+1) + (1-u)v_{Li}(t+1), \quad x_i(t+1) = x_i(t) + v_i(t+1) \quad (3)$$

where  $u \in [0, 1]$  is the unifying coefficient.

The iterations are repeated until the stopping criterion is fulfilled. The algorithm parameters are: number of particles  $S$ , radius of neighborhood  $r$ . In the standard PSO algorithm it is usually assumed that the search space of feasible solution is a  $M$ -dimensional hyperrectangle defined by the bounds  $x_{min}, x_{max}$  and the velocity components are also bounded by  $v_{min}, v_{max}$ . Previous investigations of particle dynamics [6] motivated the choice of particular parameters values:  $\chi = 0.729$ ,  $c_1 = c_2 = 2.05$  assuring a compromise between particle convergence and exploration abilities, and these values will be applied in this contribution.

### 3 Optimization Problem

Let us consider  $n$ -dimensional, square, nonsymmetrical real matrix  $A$  and the task of numerical approximation of its eigenvalue  $s$ . The problem will be formulated as minimization of a certain objective function and solved by PSO.

It is not difficult to distinguish the subset of the complex plane containing all eigenvalues of matrix  $A$ . Gershgorin circle theorem, Brauer's ovals theorem [9], or some fresh results [10] give us tools to find the area. The proposed approach may be also used for looking for eigenvalues of a priori defined set of the complex plane.

If we consider a rectangular set of the complex plane it is convenient to code the potential eigenvalue as:

$$s = x_1 + jx_2 \rightarrow x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad x_{imin} \leq x_i \leq x_{imax} \quad i = 1,2, \quad (4)$$

or if we prefer to search in circles, we will choose

$$s = Re^{j\varphi} \rightarrow x = \begin{bmatrix} R \\ \varphi \end{bmatrix}, \quad 0 \leq R \leq R_{max}, \quad -\pi \leq \varphi < \pi. \quad (5)$$

As the eigenvalue is the root of the characteristic equation it is tempting to minimize objective function

$$F(s) = |\det(sI - A)| \quad (6)$$

and this approach is chosen in [3,4]. Unfortunately calculation of determinant to test how far we are from the eigenvalue is not a reasonable idea. Determinant is computed as a product of diagonal entries of matrix  $U$  obtained from LU-factorization (triangular factorization) of  $sI - A$ . If  $s$  is an eigenvalue of  $A$ , at least one of diagonal elements of  $U$  should equal zero. Because of round off errors it will be a small number, depending on used floating point arithmetic. For example in Matlab it may be  $10^{-13}$ . The other diagonal entries may be as big as  $s$ . Even though we apply scaling, such that all entries of  $A$  are smaller than 1 it is not difficult to have, say,  $s \approx 10$  (see section 4). In this situation we will observe  $|\det(sI - A)| \approx 10^{n-1}10^{-13}$ , so far from zero even for moderate  $n$ . Therefore we recommend to apply objective function calculated as:

$$F(s) = \min\{|u_i|: u_i \in \{\text{diag}(U)\}\} \quad (7)$$

where  $U$  is obtained from LU-factorization of  $sI - A$  and  $\{\text{diag}(U)\}$  is a set of diagonal entries of  $U$ .

Several characteristic features of the formulated optimization problems are important for the solution strategy. The search space is two-dimensional. The objective function possesses the same number of minima as number of distinct eigenvalues, and all minimal values equal zero. Therefore it is easy to decide that the minimum is found.

Particle swarm optimization algorithm will find the first eigenvalue on random in the searched area. The aim of optimization is to find all eigenvalues, so we have to elaborate procedures to find the subsequent eigenvalues. There exist reduction techniques allowing to replace the found eigenvalue by zero, but this operation is a

source of additional errors. Therefore we decided to adopt another approach. PSO will be run several times and repulsion technique, similar to proposed in [6], will be introduced to avoid finding the known minima.

Let  $x^*$  be the already found minimizer of the objective function. If the particle position  $x_i$ , updated by (3), is placed closer than  $\rho_{max}$  from  $x^*$  it will be moved away from  $x^*$  by a displacement vector, which length equals  $\delta$ :

$$\rho_i = \|x_i - x^*\| < \rho_{max} \Rightarrow x_i := x_i + \frac{\delta}{\rho_i} (x_i - x^*). \quad (8)$$

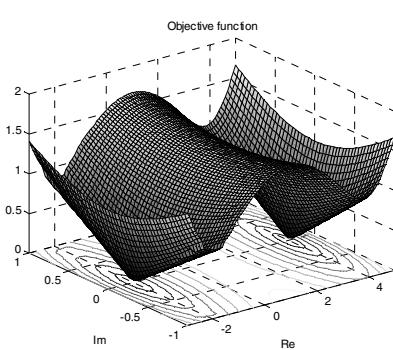
The method introduces two extra parameters  $\rho_{max}$  and  $\delta$ . Of course  $\rho_{max}$  should be smaller than half of the distance between two nearest eigenvalues, the quantity which is sometimes difficult to judge. It is quite reasonable to connect  $\rho_{max}$  with the smallest radius of Gershgorin discs. The set of eigenvalues is self-conjugated, and this will be taken into account during optimization. The well-thought-out politics of gradual enlargement of the searched area, or searching sub-areas one by one will provide noticeable benefits to algorithm reliability.

## 4 Numerical Examples

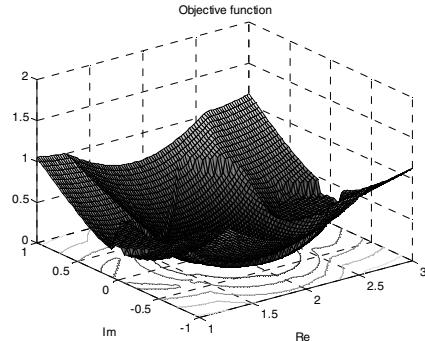
Example 1. (multiple eigenvalues) Let us consider an example taken from [3]:

$$A=[-3 \ 1 \ -1; -7 \ 5 \ -1; -6 \ 6 \ -2] \quad (9)$$

The exact eigenvalues are 4, -2, -2. Matlab calculates approximations 4.000000000000002, -2.000000029129411, -1.999999970870585, so the error is about  $10^{-8}$ .



**Fig. 1.** The objective function landscape for the matrix with double eigenvalue



**Fig. 2.** The objective function landscape for the matrix with quadruple eigenvalue

The objective function landscape is presented in figure 1. Particle swarm optimization algorithm was implemented in Matlab and run ten times with parameters  $S=30$ ,  $u=0.5$ ,  $\rho_{max} = 0.2$ ,  $\delta = 0.5$ . After each run the recently found eigenvalue constitutes the set of known minima. As a success we recognize a run with minimal

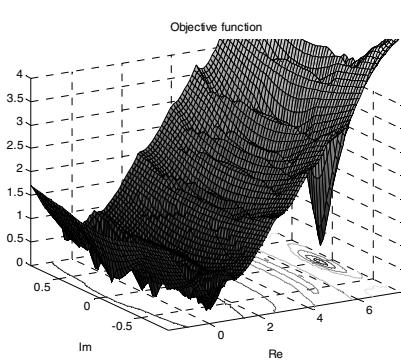
value smaller than  $10^{-17}$ . The success rate was 100%. The biggest error was smaller than  $10^{-16}$ . Hence in this case PSO was better than standard Matlab procedures.

Next we have constructed a matrix with a quadruple eigenvalue at point 2 by similarity orthogonal transformation of a Jordan block. Orthogonal transformation matrix was obtained executing command

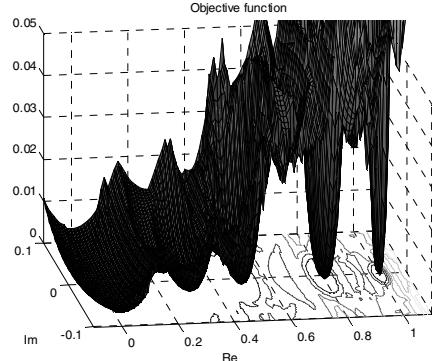
```
orth([2 1 4 2; 1 1 1 1; 1 2 4 0; 0 1 4 2]).
```

Matlab calculates its eigenvalues as: 1.999925819590832, 2.000074112606478,  $2.000000033901340 \pm 0.000074146523240i$ , so the error is about  $10^{-5}$ . The objective function landscape is presented in figure 2. The performance of PSO optimizer was similar to the previous case.

Example 2. (15-dimensional matrix) We have constructed 15-dimensional matrix choosing its entries on random, uniformly in  $[0, 1]$ . Particle swarm optimizer with repulsion was run with parameters  $S=30$ ,  $u=0.5$ ,  $\rho_{max} = 0.05$ ,  $\delta = 0.05$  and each time the recently found eigenvalue was added to the set of known minima. The success was reported if the objective function value was smaller than  $10^{-13}$ . A typical objective function landscape was plotted in figure 3. As we see the last eigenvalue is a bottom of a very deep and narrow funnel. The algorithm had difficulties with finding this minimum, when we were searching the area containing all eigenvalues. If we split the searched region into two separate parts, we will find all eigenvalues easily. Approximate values of eigenvalues computed by Matlab and by the particle swarm optimizer with repulsion differs by components smaller than  $10^{-14}$ .



**Fig. 3.** The objective function landscape for 15-dimensional matrix



**Fig. 4.** The objective function landscape for matrix with five triple eigenvalues

Experiments were repeated with an artificial matrix with five triple eigenvalues 0, 0.3, 0.5, 0.8, 1. The objective function landscape was plotted in figure 4. The swarm with repulsion parameters  $\rho_{max} = 0.1$ ,  $\delta = 0.1$  was able to find all eigenvalues with 100% success rate and with the accuracy better than  $10^{-6}$ .

## 5 Conclusions

We have investigated and positively verified swarm optimizer capability to compute matrix eigenvalues. We have proposed a new objective function to transform the problem of eigenvalue calculation into minimization task. Although the problem is only two-dimensional one, the objective function landscape is rather unfriendly – non-smooth, with as many global minima as the number of distinct eigenvalues. The landscape may be described as a multi-funnel one. Such type of optimization problems were recently studied in [9]. Conclusions presented there were rather pessimistic. It was claimed that usually a small number of population members in a ‘good funnel’ is virtually ignored by a swarm whose majority is operating in another, suboptimal funnel. This observation was only partially confirmed in the last example. Generally speaking the applied techniques – repulsion from the known minima and sequential enlarging the searched area - allow the swarm to behave quite satisfactory.

PSO is not a serious challenger in finding all eigenvalues of high dimensional matrices, but we have demonstrated that it gives better results than standard procedures in case of multiple eigenvalues. It is also a good tool to find an eigenvalue in a pre-defined region of the complex plane. This problem appears in many engineering applications, for example to test if a construction will generate vibrations in a certain frequency range.

The proposed approach may be widespread to a problem of eigenvector calculation and to generalized eigenproblem.

Finally, finding eigenvalues by optimization constitutes interesting benchmarks to test effectiveness of different PSO variants.

## References

- [1] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. 1995 IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
- [2] Huang, J., Cao, D.: Particle Swarm Optimization Method for Solving Quadratic Eigenvalues. In: 3rd International Conference on Biomedical Engineering and Informatics (BMEI 2010), pp. 2932–2935 (2010)
- [3] Zhou, Y., Wei, X.: A Novel Numerical Computation Method Based on Particle Swarm Optimization Algorithm. *Journal of Computers* 5, 226–233 (2010)
- [4] Wei, X., Zhou, Y.: PSO Algorithm for Solving Matrix Eigenvalues and Eigenvectors. *Computer Engineering* 36, 189–191 (2010)
- [5] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK User’s Guide, 3rd edn. SIAM, Philadelphia (1999), [http://www.netlib.org/lapack/lug/lapack\\_lug.html](http://www.netlib.org/lapack/lug/lapack_lug.html)
- [6] Parsopoulos, K., Vrachatis, M.N.: Particle swarm optimization and intelligence: advances and applications, Hershey, New York (2009)
- [7] Varga, R.S.: Gershgorin-type eigenvalue inclusion theorems and their sharpness. *Electronic Transactions on Numerical Analysis* 12, 113–133 (2001)
- [8] Cveticic, L., Kostic, V., Varga, R.S.: A New Gershgorin-type Eigenvalue Inclusion Set. *Electronic Transactions on Numerical Analysis* 18, 73–80 (2004)
- [9] Sutton, A.M., Whitley, D., Lunacek, M., Howe, A.: PSO and multi-funnel landscapes: how cooperation might limit exploration. In: Genetic and Evolutionary Computation Conference, GECCO (2006)

# A Study on the Utility of Parametric Uniform Crossover for Adaptation of Crossover Operator

Farhad Nadi and Ahamad Tajudin Khader

School of Computer Sciences, Universiti Sains Malaysia, Penang Malaysia  
`fnm.cod09@student.usm.my, tajudin@cs.usm.my`

**Abstract.** The parametric uniform crossover is a general form of the uniform crossover operator by which the swapping probability of each locus could be controlled. The swap probability could control the amount of disruption of high order hyper planes. Several variations of selecto-recombinative genetic algorithms are proposed for controlling the swap probability in the parametric uniform crossover operator. The suitability of the operator for diversifying the population while reducing disruption of the good partial solutions is studied. The experiments showed significant improvement in the performance of the algorithms when the parametric uniform crossover were used in comparison to algorithms that uniform crossover have been used in them.

**Keywords:** Genetic Algorithms, Diversify, Parameter Control, Parametric Uniform Crossover.

## 1 Introduction

Recombination operator, plays an important role in Genetic Algorithms (GAs) [1]. It has been shown that different crossover operators have different disruptive effects [1][2][3][4]. The disruption of the sampling distribution could be translated to exploration of the search space. The level of disruption varies in different crossover operators. However, uniform crossover has the most disruptive effect regardless of the order of the hyper-planes [1]. Mutation operator has some disruptive effects as well; however its effect is negligible due to its low rate, most of the time. Parametric Uniform Crossover (PUC) is a general form of uniform crossover operator on which the swap rate of the alleles ( $p_0$ ) could be controlled. It has been shown that the value of  $P_0$  is effective in increasing the survival probability of high order hyper-planes [1]. Premature convergence of the would cause the loss of the diversity in the population and would stop the search from further progresses. This problem could be rectified by diversifying the population. However, diversification would cause disruption of the existing good solutions. This research aims in diversify the population while reducing the amount of disruption of good solutions. The virtues of PUC suggests that this aim could be achieved by adaptation of the swap rate of the PUC.

## 2 Deterministic Parametric Uniform Crossover

Four variations of algorithms are proposed that adapt the swap rate of the PUC. The general theme in all of them is to diversify the population if the population's diversity goes below a certain threshold. Diversification of the population will be start then, while the value  $p_0$  will be set using different strategies. Algorithm I shows the general pseudo code for the four variations of the algorithms. The Genotypic Diversity Measure (GDM) of the population calculated using,

$$GDM = 1 - \left( \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{h(I_i, I_j)}{l}}{\binom{N}{2}} \cdot 1 - \left( \frac{X}{N} \right) \right).$$

Where  $N$  is the population size,  $X$  is the number of similar individuals within the population, and  $h(.,.)$  returns the hamming distance between two given individuals.

---

**Algorithm 1.** Pseudo code of the utilized selecto-recombinative GA in one iteration of run.

---

```

Data:  $I_1, I_2, s_p, p_0 // I_1, I_2$  represent parent solutions
Result:  $I'_1, I'_2 // I'_1, I'_2$  represent off-spring solutions
1  $I_1 = tournament(\rho, s_p)$  //tournament(.,.) would select an individual from population ( $\rho$ ) with
   selection pressure of  $s_p$ 
2 if  $GDM() < 0.2 // GDM()$  returns the population's diversity.
3 then
4    $| p_0 = \alpha$ 
5    $| I_2 = tournament(2)$ 
6 else
7    $| I_2 = Random\_solution()$ 
8    $| p_0 = \beta$ 
9 end
10  $I'_1, I'_2 = PUC(I_1, I_2, p_0);$ 
11 if  $f(I'_1) > f(I'_2)$  then
12    $| Replace(\rho, I'_1)$  //Replace(.,.) would find the worst individual within the population ( $\rho$ ) and will
      replace it with the provided individual in the second parameter.
13 else
14    $| Replace(\rho, I'_2);$ 
15 end
```

---

Table I lists the proposed algorithms as well as other algorithms that have been used for comparisons are listed as well.

**Table 1.** Different variations of deterministic methods and other types of GA with their assigned labels

label	type of algorithm
$D_v$	$\begin{cases} \alpha = 3/l & GDM() > 0.8 \\ \beta = 0.1 & \text{otherwise} \end{cases}$
$D_vE$	like $D_v$ , and $I_1$ is the best individual
$D_F$	$\beta = \alpha = 0.1$
$D_F E$	$\beta = \alpha = 0.1$ , and $I_1$ is the best individual
$S_r$	selecto-recombinative GA
$S_r E$	like $s_r$ , and $I_2$ is the best
$GA_{vpm}$	SGA where $\begin{cases} p_m = 3/l & GDM() > 0.8 \\ p_m = 0.1 & \text{otherwise} \end{cases}$
$GA_{01}$	SGA with $p_m = 0.1$
$GA_{3l}$	SGA with $p_m = 3/l$

### 3 Experimental Setup

All of the experiments are ran 30 times and the Mean of the Best Fitness (MBF) [5] of each experiment is reported in the results. A steady state GA with tournament selection with selection pressure of 2, population size of 20, and maximum number of iteration is set to 500 has been used in the experiments. The performance of the proposed method is assessed over Onemax and 4-bits Trap functions. Six instances of benchmarks have been used for each of the test problems including , (50 ,100 ,200 ,400 ,800 ,1600),and (5 ,10 ,15 ,20, 25, 30)respectively for Onemax, and 4-bits Trap functions. Four variations of the proposed deterministic methods are compared with two selecto-recombinative algorithms, and three variation of Simple Genetic Algorithm(SGA).

### 4 Results and Discussion

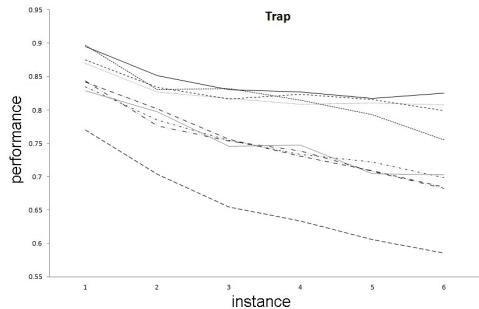
Figure 1(a) depicts the comparison of the different methods' performance over instances of 4-bits Trap function which could be categorized into three main clusters. One cluster includes the  $S_rE$ , which got the worst performance among the compared methods. The other cluster contains four algorithms with almost the same performances. These algorithms are,  $S_r$ ,  $GA_{vpm}$ ,  $GA_{01}$ , and  $GA_{3l}$ . The third cluster, with relatively better performances, includes all of the deterministic algorithms.

Figure 1(b) shows the comparison of the methods over the instances of One-max function. The compared algorithms could be categorised as in Trap function. However, for the deterministic methods, it seems that the obtained results could not be grouped together as in the two previous benchmarks.

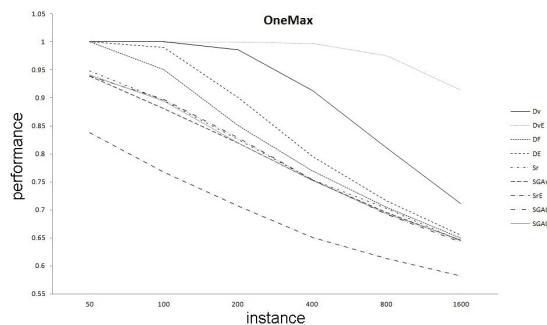
The  $S_rE$  algorithm obtains the worst performance among all of the benchmark algorithms. This algorithm is a selecto-recombinative GA (SrGA), where one of the parents is always the best existing solution. The only difference of this algorithm from the  $S_r$  is the selection of the best existing solution as one of the parents. As the  $S_r$  has shown a comparatively better performance, it could be concluded that selection of best individual, as one parent, would be the reason of such lower performance of the  $S_rE$ . In this algorithm ( $S_rE$ ), the rate of homogeneity of the population will increase faster, and thus the diversity of the population will reduce faster than the normal case. This would prevent the search from exploring the search space adequately.

The second cluster includes,  $SrGA$ ,  $S_r$ , and three variations of (SGA),  $GA_{vpm}$ ,  $GA_{01}$ , and  $GA_{3l}$ . The differences between the performances of the algorithms in this cluster are not significant and are almost the same. Basically the methods in this cluster are differing in the mutation rate. Such results are not very surprising in view of the fact that mutation would cause small changes in the off-spring.

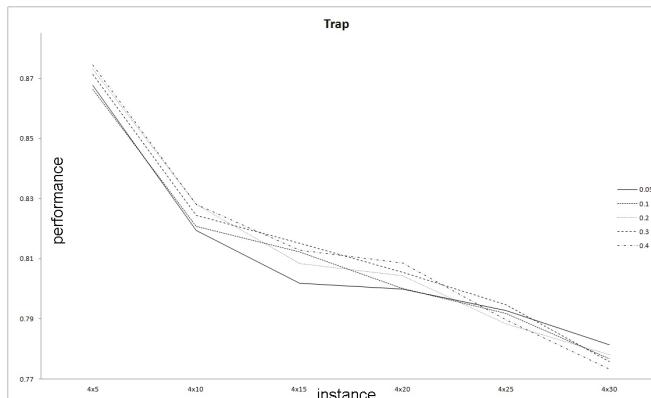
The third cluster contains all of the deterministic methods. These methods have shown a relatively better performance over the other methods including variations of SGA and SrGA. A very intuitive inference would be that the utilization of the PUC has shown to be promising.



(a) Comparison of the compared methods over Trap function.

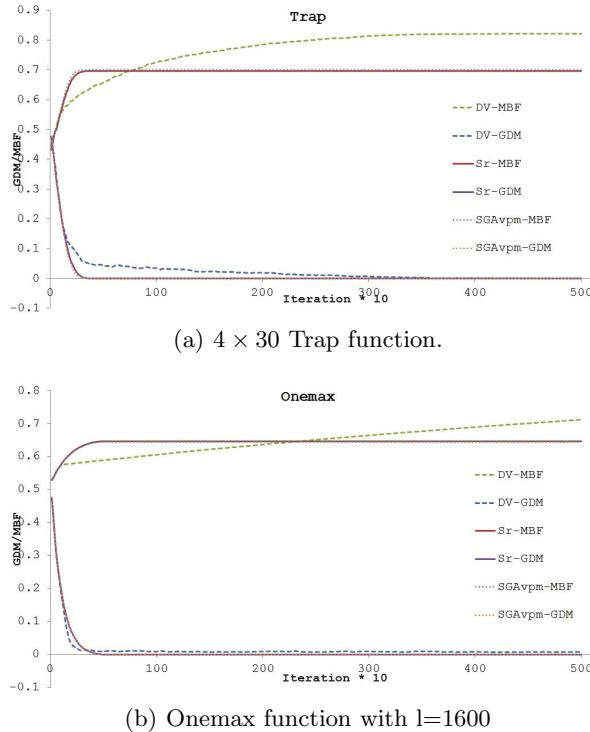


(b) Comparison of the compared methods over One-max problem.



(c) Effect of diversity threshold on the performance of the algorithms

**Fig. 1.** Comparison of the performance of different methods over different benchmark problems



**Fig. 2.** Change in GDM and MBF of different methods on every 100 generations. GDM and MBF values are averaged over 30 runs.

The agenda in the deterministic methods was to perform mate mate the existing solutions with a new random solution using a low rate of  $p_0$ . Although this seems very similar to mutating of existing solutions, the experiments have revealed contradictory results.

As mentioned earlier, if the diversity of the population decreases below a certain percentage, 20 percent in this case, the PUC with a new rate for  $p_0$  will be applied. The diversity threshold indeed has effect on the performance of the algorithms. Figure 1(c) depicts average performance of the deterministic methods running with different diversity threshold over Trap function. It seems that low diversity thresholds, 0.05 and 0.1, are more suitable for harder problems, while higher thresholds are good for the easy instances. Apart from this, it could be said that the higher diversity thresholds resulted in better performances. However, a mid-value diversity threshold, 0.2, has shown the most stable performance in comparison to the others.

Figure 2 compares the changes in the *MBF* value with regards to GDM for the three methods. The MBF and GDM values of each method have been averaged over 30 runs, every 100 generations. In all the benchmark problems,

the performance of the  $D_V$  method has not been comparable with the two other methods in the early stages.

However, the Performance of the  $D_V$  method gets better after some generations. The algorithms have performed almost the same in the early stages. Right after the GDM decreases below the threshold, the rate of MBF's improvements in  $D_V$  method has decreased. After this time, the rate of GDM has been decreased as well.

In summary it could be said that diversifying the population in a disruptive algorithm would result in increase of the GDM. However, as it could be seen in Figure 2, The population is not actually diversified, by mean of disruption, but the steepness of the slope of the diversity is diminished. In all of the benchmarks it took some generations for the  $D_V$  method to defeat the other two methods. The other two algorithms have been performing almost the same in all of the benchmarks either in MDF or GDM.

## 5 Conclusion

Parametric Uniform Crossover (PUC) is a general form of uniform crossover by which it would be possible to control the mixing amount of genetic materials. Four variations of the algorithms that adapted the swapping rate of the PUC has been compare with some other variations of the GA and selecto-recombinative GA to study the effectiveness of PUC for diversification of the converged population. The experiments have shown that utilization of the deterministic PUC was beneficial and the improvement is significant.

**Acknowledgement.** This research was partially supported by FRGS Grant 1001/PKOMP/844037 from the Universiti Sains Malaysia.

## References

1. Spears, W., de Jong, K.: On the virtues of parameterized uniform crossover. In: Belew, R., Booker, L. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 230–236. Morgan Kaufman, San Mateo (1991)
2. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975)
3. Syswerda, G.: Uniform crossover in genetic algorithms. In: Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 2–9. Morgan Kaufmann Publishers Inc., San Francisco (1989)
4. Eshelman, L., Caruana, R., Schaffer, J.: Biases in the crossover landscape. In: Proceedings of the Third International Conference on Genetic Algorithms, pp. 10–19. Morgan Kaufmann Publishers Inc. (1989)
5. Eiben, A., Smith, J.: Introduction to Evolutionary Computing, 2nd edn. Natural Computing Series. Springer (2007)

# Decomposition, Merging, and Refinement Approach to Boost Inductive Logic Programming Algorithms

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 (ILP) deals with the problem of finding a hypothesis covering positive examples and excluding negative examples. It uses first-order logic as a uniform representation for examples and hypothesis. In this paper we propose a method to boost any ILP learning algorithm by first decomposing the set of examples to subsets and applying the learning algorithm to each subset separately, second, merging the hypotheses obtained for the subsets to get a single hypothesis for the complete set of examples, and finally refining this single hypothesis to make it shorter. The proposed technique significantly outperforms existing approaches.

**Keywords:** inductive logic programming, boosting, problem decomposition.

## 1 Introduction

Assume the problem of identifying a common structure in the set of proteins such that this structure does not appear in another set of proteins. This can be done by describing the graphical structure of the protein as a set of grounded atoms such as  $\{\text{arc}(a,b), \text{arc}(b,c), \text{arc}(c,a)\}$ . We will call this set a *positive example*, if it represents the protein where the structure appears, and a *negative example* otherwise. The common sub-structure can also be described as a set of atoms, now with the variables, such as  $\{\text{arc}(X,Y), \text{arc}(Y,Z)\}$ . This set is called a *hypothesis*. *Inductive logic programming* [7] deals with the problem of finding a consistent hypothesis that entails all positive examples and does not entail any negative example. In such a setting, the process of inventing the consistent hypothesis (ILP consistency algorithm) may consist of determining its structure, that is, which atoms and how many of them are in the hypothesis, and subsequent finding of the unifications between the variables in the atoms in such a way that we obtain a consistent hypothesis [5].

In this paper we propose a boosting technique that significantly speeds up the core ILP consistency algorithm from [2]. This technique is based on the idea of divide-and-conquer. We first decompose the set of examples into several subsets. Then we find a consistent hypothesis for each subset of examples separately. After that we merge the obtained hypotheses to get a single hypothesis consistent with the original set of examples. As a shorter hypothesis is usually preferred (Ockham's razor) we suggest a final refinement stage that shortens the hypothesis. The resulting algorithm is called DeMeR (Decomposition-Merging-Refinement).

In the paper we will first give a formal background of ILP and describe the techniques behind the core ILP consistency algorithm from [2]. Then we will give the technical details of the DeMeR algorithm. We will conclude the paper with the experimental study showing efficiency of DeMeR and its stages.

## 2 Background

As usually done in ILP, we will assume 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) are 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.

As usual in ILP,  $\theta$ -*subsumption* [8] is used to approximate the entailment relation. Hypothesis  $H$  *subsumes* example  $E$ , if there exists a substitution  $\theta$  of variables such that  $H\theta \subseteq E$ . In the above example, substitution  $\theta = \{X/a, Y/b, Z/c\}$  implies that  $H$  subsumes  $E$  (there are more such substitutions, for example  $\{X/b, Y/c, Z/a\}$ ). The requirement that a negative example  $E^-$  is not subsumed by hypothesis  $H$  means that there does not exist any substitution  $\theta$  such that  $H\theta \subseteq E^-$ . For example, hypothesis  $G = \{\text{arc}(X,Y), \text{arc}(Y,X)\}$  does not subsume the above example  $E$ . Maloberti and Sebag [6] proposed a fast  $\theta$ -subsumption algorithm called *Django* that is based on reformulation of  $\theta$ -subsumption as a binary constraint satisfaction problem.

ILP *template consistency problem* [5] is formulated as follows: given a clause  $T$  with unique appearance of variables (each variable appears exactly once in  $T$ ) called a *template*, find a substitution  $\sigma$  making hypothesis  $H = T\sigma$  consistent with the learning examples. Since all terms in  $H$  are supposed to be variables, the task lies in determining which subsets of variables in  $T$  should be unified. For instance an exemplary hypothesis  $H$  may be obtained from template  $T = \{\text{arc}(X_1, X_2), \text{arc}(X_3, X_4)\}$  by applying unification  $X_2 = X_3$  (and then suitably renaming the variables). Note that the unifications are introduced primarily to break subsumption of negative examples in the evidence (if  $H$  subsumes some example  $E$ , i.e.,  $\exists \theta H\theta \subseteq E$ , then  $T$  subsumes that example too because  $H = T\sigma$  and hence  $T\sigma\theta \subseteq E$ ). Barták et al. [2] showed how to effectively formulate the template consistency problem as a constraint satisfaction problem. This algorithm uses an iterative generate-and-test method, where all possible templates are systematically explored with the increasing length of the template and for each template the procedure to solve the template consistency problem is invoked to find a consistent hypothesis. If no consistent hypothesis exists then another template is tried as shown in Algorithm 1.

**Algorithm 1.** Core ILP Consistency Algorithm

---

```

1:    $k \leftarrow$  the number of all different predicates in examples
2:   repeat
3:      $S_T \leftarrow$  all templates of length  $k$  (each predicate appears in the template)
4:     for each  $T \in S_T$  do
5:        $H \leftarrow \text{Decide-Consistency}(T)$ 
6:       if  $H$  is consistent then return  $H$ 
7:     end for
8:      $k \leftarrow k + 1$ 
9:   until stop criterion satisfied

```

---

### 3 Decompose-Merge-Refine Approach

Our experimental evaluation of Algorithm 1 showed that the iterative algorithm has very poor runtime properties. In this section we suggest another algorithm for finding a consistent hypothesis based on problem decomposition. When investigating the behavior of Algorithm 1, we can observe that its overall performance decreases dramatically with the length of the template it processes. Furthermore, the bigger evidence (the larger number of examples) is used in the setting, the more complicated final template is required. On the other hand, it is worth noticing that we are not very concerned about the complexity of the positive evidence because unifications of variables in the template consistency algorithm are meant to break only the negative evidence [2]. Following these observations we propose a novel algorithm DeMeR, which abbreviates *decompose – merge – refine*. DeMeR is a method taking advantage of the well-known divide-and-conquer technique. In particular, the algorithm consists of three subsequent stages:

- 1) the evidence (examples) is *decomposed* into several subsets and Algorithm 1 is applied to each subset separately,
- 2) all the obtained partial hypotheses are then *merged* into a single hypothesis,
- 3) the possibly long hypothesis is post-processed to reduce its length (*refinement*).

#### 3.1 Decomposition Stage

The main idea behind the decomposition stage is to divide the evidence into  $k$  groups in order to decrease its complexity and consequently to decrease overall runtime of  $k$  instances of Algorithm 1. First we must settle a method how the evidence is divided into subgroups. We went for an approach when we grouped certain negative examples with all positive examples in each group. Each partial hypothesis generated for the corresponding group subsumes all positive examples and hence the union of the partial hypothesis also subsumes all positive examples. Moreover, each partial hypothesis breaks some negative examples and hence the union of the hypothesis breaks all negative examples from the groups. So after the union of all partial hypotheses, we easily obtain a consistent hypothesis with respect to the original evidence as Proposition 1 says (all formal proofs are in [3]).

**Proposition 1.** *Let  $E_1 = E^+ \cup E_1^-$ , ...,  $E_k = E^+ \cup E_k^-$  be a decomposition of the evidence  $E = E^+ \cup E^-$  into  $k$  groups, such that  $E^- = \bigcup_{i=1}^k E_i^-$ . Further let  $H_i$  be a consistent hypothesis with respect to  $E_i$ . Then  $H = \bigcup_{i=1}^k H_i$  is a consistent hypothesis (after standardizing variables apart) with respect to the original evidence  $E$ .*

A non-trivial question is about the rule determining which negative examples should be put into the same group. Intuitively, we would like to put those negative examples that have a similar structure into the same group. In that case some set of unifications breaking one negative example in the group will likely break other negative examples in the group as well. Unfortunately, it is not clear how to define the similarity of the negative examples. Actually it is the original task of ILP to discover such similarities. That is why we chose the simplest approach in our algorithms and grouped the negative examples as they appeared in the input data.

Another important issue about the decomposition is deciding the number  $k$  of individual groups created by the division of the negative evidence. If we choose  $k = |E|$ , then there are all positive examples and just one negative example in each group giving  $|E|$  separate subtasks. Conversely, choosing  $k = 1$  gives the original problem. A particular value of  $k$  defines the trade-off between the runtime of the algorithm and the length of the joint hypothesis. The bigger  $k$  decreases the runtime but lengthens the final hypothesis. The smaller  $k$  produces shorter hypotheses at the cost of longer runtime (see the experimental results later in the text).

In Algorithm 2 we show the pseudo code of the decomposition stage.

### Algorithm 2. Decomposition stage of DeMeR

---

```

1: initialize  $k$ - the number of subtasks, evidence  $E$ 
2: function DECOMPOSE-AND-SOLVE( $k, E$ )
3:    $S_E \leftarrow$  decomposition of the evidence into  $k$  groups
4:    $S_H \leftarrow \emptyset$  {a set of partial hypotheses}
5:   for each  $E \in S_E$  do
6:      $H_{\text{part}} \leftarrow$  hypothesis consistent with respect to  $E$ 
7:      $S_H \leftarrow S_H \cup \{H_{\text{part}}\}$ 
8:   end for
9:   return  $S_H$ 
10:  end function

```

---

## 3.2 Merge Stage

The joint consistent hypothesis can be produced by concatenation (union) of all partial hypotheses (including standardizing variables apart) obtained from the decomposition stage (set  $S_H$  in Algorithm 2). According to Proposition 1 this method is sound but the resulting hypothesis is frequently very long. Now we will show a technique how to create a shorter final hypothesis. In particular, we will examine the impact of *contracting* some predicates from the hypotheses while merging them.

Assume without loss of generality that we are concatenating two partial hypotheses. We pick two predicates with the same predicate symbol, each predicate from one hypothesis. What does happen if we *contract* the two predicates into one and then concatenate the rest of the hypotheses? If we pick the predicates right and do some little extra work, the joint hypothesis will be shorter by one predicate and still consistent. Definition 1 describes this process formally.

**Definition 1.** Let  $H_1 = \{p_1, \dots, p_n\}$ ,  $H_2 = \{q_1, \dots, q_m\}$  be two hypotheses with disjoint sets of variables,  $p_i = p(X_1, \dots, X_k)$ ,  $q_j = p(Y_1, \dots, Y_k)$  be two predicates with the same predicate symbol. A (*single*) *contraction* of the hypotheses  $H_1, H_2$  with respect to predicates  $p_i$  and  $q_j$  is a hypothesis  $H_3$ , such that  $H_3 = \{p_1, \dots, p_n, q'_1, \dots, q'_{j-1}, q'_{j+1}, \dots, q'_m\}$  where  $q'_i = q_i\theta$  with  $\theta = \{Y_1/X_1, \dots, Y_k/X_k\}$ . The substitution  $\theta$  is called the *contracting substitution* and the predicate  $p_i = q_j\theta$  is called the *contracted predicate*. We denote  $H_3 = (H_1 + H_2)_{[p_i, q_j]}$ .

**Example 1.** Suppose we are given two hypotheses  $H_1 = \{\text{arc}_1(X_1, X_2), \text{arc}_1(X_3, X_1), \text{arc}_1(X_2, X_4), b_1(X_4)\}$  and  $H_2 = \{\text{arc}_2(Y_1, Y_3), \text{arc}_2(Y_1, Y_2), b_2(Y_2)\}$  (predicate symbols  $\text{arc}_1$  and  $\text{arc}_2$  are identical; they are distinguished to show their membership to the hypotheses  $H_1$ ,  $H_2$ ). A contraction of predicates  $\text{arc}_1(X_2, X_4)$  and  $\text{arc}_2(Y_1, Y_2)$  is a hypothesis  $H_3 = \{\text{arc}_1(X_1, X_2), \text{arc}_1(X_3, X_1), \text{arc}_1(X_2, X_4), b_1(X_4), \text{arc}_2(X_2, Y_3)\}$ .

Notice that if the resulting hypothesis  $H_3$  contains predicates  $p_s$  and  $q_t$ , such that  $p_s = q_t$  for some  $s$  and  $t$ , then we can remove one of them, say  $q_t$  (as  $b_2(X_4)$  in Example 1).

The contraction imposes two features:

- 1) the length of the contracted hypothesis is now shorter by (at least) one predicate;
- 2) the contraction introduces an additional *binding* of predicates in the final hypothesis (predicates  $\text{arc}_1(X_1, X_2)$  and  $\text{arc}_2(X_2, Y_3)$  now share variable  $X_2$ ).

If we look at the contractions from the previous example, we can see that we additionally bound all predicates containing variables  $X_2$ ,  $X_4$  in the hypothesis  $H_1$  with all predicates containing variables  $Y_1$ ,  $Y_2$  from the hypothesis  $H_2$ . In particular, the newly bound predicates are  $\text{arc}_1(X_1, X_2)$ ,  $\text{arc}_1(X_2, X_4)$ ,  $\text{arc}_2(X_2, Y_3)$  and  $\text{arc}_1(X_2, X_4)$ ,  $b_2(X_4)$  (which is removed). Such binding is dangerous because merging two hypotheses consistent with respect to the positive evidence can create a hypothesis that is no more consistent with respect to the positive evidence. On the other hand, as Proposition 2 says, the contraction of two hypotheses consistent with respect to negative evidence gives a hypothesis that is still consistent with negative evidence.

**Proposition 2.** Let  $H_1$  be a hypothesis consistent with respect to negative evidence  $E_1^-$ ,  $H_2$  be a hypothesis consistent with respect to negative evidence  $E_2^-$ , and  $H_3 = H_1 + H_2$  be the contraction of  $H_1$  and  $H_2$ . Then  $H_3$  is consistent with the negative evidence  $E^- = E_1^- \cup E_2^-$ .

We observed that inconsistency of the contracted hypothesis could only be due to the positive evidence. It would be possible to check certain structural properties of the contracted predicates in advance, for example using techniques such as vertex invariants under isomorphism between hypotheses [11], but this is too time consuming. We rather decided to simply contract the hypotheses and then to run the  $\theta$ -subsumption test (Django algorithm) on the joint hypothesis to check that it still subsumes the positive evidence. If the test succeeds then the contraction is accepted. Otherwise we pick two other predicates for contraction and repeat the process.

By now we were contracting just one pair of predicates. However, it is often possible to do more contractions within two hypotheses. We extend the definition of the single contraction to *multi-contraction* in order to be able to reduce more predicates at once. A basic idea is that after each contraction has been done, we apply the corresponding substitutions, remove the contracted predicate from one of the hypotheses but we do not concatenate the hypotheses immediately.

**Definition 2.** Let  $H_1 = \{p_1, \dots, p_n\}$ ,  $H_2 = \{q_1, \dots, q_m\}$  be two hypotheses with different sets of variables. Further, let  $P_1 = \{p'_1, \dots, p'_k\}$ ,  $P_2 = \{q'_1, \dots, q'_k\}$  be the sequences of predicates chosen from  $H_1$ , resp.  $H_2$ ,  $p'_i = p(X_1, \dots, X_l)$ ,  $q'_i = p(Y_1, \dots, Y_l)$ . For each pair  $p'_i, q'_i$  let us denote  $\theta_i = \{Y_1/X_1, \dots, Y_l/X_l\}$  a corresponding unification of variables. The  $k$ -multi-contraction of the hypothesis  $H_1$  and  $H_2$  with respect to the sequences of predicates  $P_1$  and  $P_2$  is a hypothesis  $H_3$ , such that  $H_3 = H_1\theta \cup (H_2 \setminus P_2)\theta$ , where  $\theta = \theta_k \dots \theta_1$  is a *contracting unification*. We denote  $H_3 = (H_1 + H_2)_{[P_1, P_2]}^k$ .

Let us explain the notation. In Definition 2,  $P_1$  and  $P_2$  are two sequences of predicates that will be pair-wise contracted. In order the contractions to make sense, the predicates on the  $i$ -th position in the sequences  $P_1, P_2$  must have an identical predicate symbol. Notice that hypothesis contains only variables (no constants or terms) so the substitutions in Definitions 1 and 2 are in fact unifications of variables. The combination of unifications is always well defined which is a necessary condition to apply multi-contraction as described in Definition 2. The contracted hypothesis  $H_3$  is then built from all predicates in  $H_1$  and all un-contracted predicates in  $H_2$  such that the combined unifications  $\theta_i$  are applied to the predicates. See Example 2 how Definition 2 is applied in practice.

Algorithm 3 is a branch-and-bound algorithm looking for the largest subset of contracted predicates by exploring all possible pairs of unifiable predicates – one predicate is from hypothesis  $H_1$  and the other one from  $H_2$ .

### Algorithm 3. Multi-contractions

---

```

1: function MULTI-CONTRACTION( $H_1, H_2$ )
2:    $H_{best} = H_1 \cup H_2$  {a global variable}
3:   CONTRACT( $H_1, H_2, \{\}, \{\}$ )
4:   return  $H_{best}$ 
5: end function

6: procedure CONTRACT( $H_1, H_2, S_1, S_2$ )
7:   for each  $p \in H_1 \& p \notin S_1$  do
8:     for each  $q \in H_2 \& q \notin S_2 \& p = q$  do //  $p$  and  $q$  are unifiable predicates
9:        $H_3 \leftarrow (H_1 + H_2)_{[p,q]}$ 
10:      if  $H_3$  subsumes positive evidence then
11:        if  $|H_3| < |H_{best}|$  then  $H_{best} \leftarrow H_3$  end if
12:         $\theta \leftarrow$  contracting unification between  $p, q$ 
13:        CONTRACT( $H_1\theta, (H_2 \setminus \{q\})\theta, S_1 \cup \{p\}, S_2 \cup \{q\}$ )
14:      end if
15:    end for
16:  end for
17: end procedure

```

---

**Example 2.** Suppose we are given two hypotheses  $H_1 = \{\text{arc}_1(X_1, X_2), \text{arc}_1(X_3, X_1), b_1(X_2)\}$ ,  $G_1 = \{\text{arc}_2(Y_1, Y_2), \text{arc}_2(Y_2, Y_3), b_2(Y_2)\}$  and we are going to perform 2-multi-contraction with  $P_1 = \{\text{arc}_1(X_1, X_2), b_1(X_2)\}$ ,  $P_2 = \{\text{arc}_2(Y_2, Y_3), b_2(Y_2)\}$  (again, the predicate symbols  $\text{arc}_1$ ,  $\text{arc}_2$  and  $b_1$ ,  $b_2$  are meant to be identical, the subscript just distinguishes the parent hypothesis of the predicates). We do the following steps:

- 1) We find a unification  $\theta_1$  of predicates  $p_1 = \text{arc}_1(X_1, X_2)$ ,  $q_1 = \text{arc}_2(Y_2, Y_3)$ :  $\theta_1 = \{Y_2 / X_1, Y_3 / X_2\}$ .
- 2) We find a unification  $\theta_2$  of predicates  $p_2 = b_1(X_2)$ ,  $q_2 = b_2(Y_2)$ :  $\theta_2 = \{Y_2 / X_2\}$ .
- 3) Now we put  $\theta = \theta_1\theta_2$  a combination of the two unifications and we apply  $\theta$  on the hypotheses  $H_1$ ,  $G_1$  yielding a new pair of the hypotheses  $H_2 = \{\text{arc}_1(X_1, X_1), \text{arc}_1(X_3, X_1), b_1(X_1)\}$ ,  $G_2 = \{\text{arc}_2(Y_1, X_1), \text{arc}_2(X_1, X_1), b_2(X_1)\}$
- 4) Finally, we remove the contracted predicates from the hypothesis  $G_2$  and get  $(H_1 + G_1)^2_{[P_1, P_2]} = \{\text{arc}_1(X_1, X_1), \text{arc}_1(X_3, X_1), b_1(X_1), \text{arc}_2(Y_1, X_1)\}$

If we know how to contract two hypotheses, then it is very easy to outline the algorithm representing the merge stage of DeMeR algorithm (Algorithm 4). The algorithm takes a set of partial hypotheses generated in the decomposition stage as its input and using Multi-Contraction procedure (Algorithm 3) it generates a single hypothesis as its output.

#### Algorithm 4. Advanced merge stage

---

```

1:  $S_H$  is a set of partial hypotheses from the decomposition stage
2: function ADVANCED-MERGE( $S_H$ )
3:    $H_{\text{result}} \leftarrow \{\}$ 
4:   for each  $H \in S_H$  do
5:      $H_{\text{result}} \leftarrow \text{MULTI-CONTRACTION}(H_{\text{result}}, H)$ 
6:   end for
7:   return  $H_{\text{result}}$ 
8: end function

```

---

### 3.3 Refinement Stage

The goal of refinement is to further decrease the size of the hypothesis obtained from the merge stage. This is done by picking an arbitrary predicate and removing it from the hypothesis. This is called the *refinement step*. The refinement step never violates consistency of hypothesis with respect to the positive examples as Proposition 3 says.

**Proposition 3.** Let  $H$  be a hypothesis consistent with the positive evidence  $E^+$ . Then removal of any predicate from  $H$  does not violate the consistency with respect to  $E^+$ .

Unfortunately, it may happen that the reduced hypothesis subsumes some negative example so the hypothesis is no more consistent with the negative evidence. We solve this problem by treating the hypothesis as a template where some variables are already unified and trying to unify additional variables to obtain a consistent hypothesis again. Algorithm from [2] can be used to find such unifications. If it is not possible to construct a consistent hypothesis then the removed predicate is returned back and another predicate is tried. Algorithm 5 describes this process.

**Algorithm 5.** Basic refinement algorithm

---

```

1: function START-REFINEMENT(H)
2:   Hbest = H {a global variable}
3:   REFINE(H, {})
4:   return Hbest
5: end function

6: procedure REFINE(H, S)
7:   for each p ∈ H & p ∉ S do
8:     H2 ← H \ {p}
9:     Hnew ← Decide-Consistency(H2)
10:    if Hnew is consistent then
11:      if |Hnew| < |Hbest| then Hbest ← Hnew end if
12:      REFINE(Hnew, S)
13:    end if
14:    S ← S ∪ {p}
15:  end for
16: end procedure

```

---

Algorithm 5 is a complete depth-first search algorithm that explores the space of all possible refinements. To make it practically applicable we use its incomplete version using the ideas of iterative broadening [4] and using a global cutoff time limit and a local cutoff time limit for the procedure of restoring consistency (line 9). With these restrictions the order of predicates to be tried for removal becomes important. Naturally, we prefer to remove first the predicates that will cause the least number of negative examples to be subsumed. The intuition is that the smaller number of unifications is broken by the deletion of the predicate, the smaller number of negative examples is likely to become subsumed by the hypothesis. We will use an *occurrence graph* introduced in [9] that describes how the variables are shared between the predicates in the hypothesis.

**Definition 3.** Let H be a hypothesis. G = (H, E) is the *occurrence graph* of the hypothesis H where  $(l_i, l_j, \pi_i \rightarrow \pi_j) \in E$  iff there is a variable x that occurs in predicate  $l_i \in H$  at argument position  $\pi_i$  and in predicate  $l_j \in H$  at argument position  $\pi_j$ .

Several heuristics for selection of predicates can be defined based on the occurrence graph. After preliminary experiments we decided to use the heuristic that combines global and local information from the occurrence graph. It is called the *weighted minimal degree heuristic* where the weight is the size of the connected component that contains the given predicate. The degree of node corresponds to the number of predicates that share some variables with the given predicate while the size of the connected component shows how many other predicates are further influenced. Naturally, the predicates appearing in smaller components and with smaller degree are tried first during the refinement.

## 4 Experimental Results

We evaluated all methods in SICStus Prolog 4.1.2 (2.0 GHz Intel Xeon processor, 12 GB RAM under Gentoo Linux). All experiments were run on instances of identifying common structures in random graphs generated according to Barabási-Réka model [1] that allowed us to generate a larger number of problem instances of different hardness. We used random graphs consisting of 20 nodes that were constructed by incrementally adding new nodes by connecting them with three arcs to existing nodes in the graph. The hidden structure that we were looking for consisted of five nodes. Both positive and negative evidence contained ten instances of the graphs.

In Table 1 we present thorough comparison between DeMeR (decomposition into 10 subtasks used) and two existing algorithms (original Algorithm 1 and well known ILP system Aleph [10]). For each algorithm we show its running time (in seconds) and length of the final hypothesis (measured by the number of atoms). In case of DeMeR we also provide information on how each stage itself performed and how it formed the final hypothesis. Finally we give the overall runtime of the algorithm. From the results it immediately follows that DeMeR outperformed the other two algorithms for almost every input instance by more than two orders of magnitude. At the same time we can see that the length of the final hypothesis is equal or only slightly higher than the length of the hypothesis in case of Algorithm 1 (which always finds the shortest hypothesis). Having look at individual stages of DeMeR we can notice that the first stage finds a solution very quickly but at cost of very long hypothesis. Other two stages subsequently modify the structure of the hypothesis up to the end of the refinement stage where the final hypothesis is of acceptable size.

**Table 1.** Comparison of DeMeR with the existing algorithms. For DeMeR we used decomposition into ten subtasks, refinement iterative broadening limit was 15, refinement cutoff timeout was 30 seconds. Maximal time limit for every algorithm was 1200 seconds.

Alg. 1		Aleph		DeMeR (10 subtasks)							
				Decomposition		Merge		Refinement			
t	len	t	len	t	len	t	len	t	len	t <sub>total</sub>	
17.4	<b>7</b>	>1200	-	1.7	43	2.0	11	1.1	<b>7</b>	<b>4.8</b>	
436	<b>8</b>	>1200	-	7.2	54	6.3	20	30.0	10	<b>43.5</b>	
460	<b>8</b>	>1200	-	54.2	60	39.3	29	30.0	10	<b>123.5</b>	
508	<b>8</b>	>1200	-	23.9	53	10.9	43	30.0	<b>8</b>	<b>72.9</b>	
>1200	-	>1200	-	10.2	51	6.8	15	3.5	<b>12</b>	<b>20.5</b>	
>1200	-	>1200	-	23.4	55	15.4	33	30.0	<b>14</b>	<b>68.8</b>	
>1200	-	>1200	-	90.9	60	31.3	31	30.0	<b>9</b>	<b>152.2</b>	
>1200	-	>1200	-	88.8	61	41.8	36	30.0	<b>10</b>	<b>160.6</b>	
>1200	-	>1200	-	103.1	53	41.3	28	30.0	<b>9</b>	<b>174.4</b>	
>1200	-	>1200	-	99.9	57	48.3	34	30.0	<b>11</b>	<b>178.2</b>	

To see how the number of subtasks in the decomposition stage influences efficiency we did the experiment with different numbers of subtasks. Table 2 shows the results for the same datasets as in Table 1. Decomposition into 5 subtasks still

performs better comparing to both Algorithm 1 and Aleph whereas decomposition into 3 subtasks lost noticeably for some input instances. The results confirmed that the larger number of subtasks decreases runtime with the tradeoff of increasing the size of the concatenated hypothesis. Nevertheless, thanks to smart merging and refinement, we can still obtain the final hypothesis of good quality (size).

**Table 2.** Results for two different decomposition parameters of DeMeR. The runtime limit was 1200 seconds.

DeMeR (5 subtasks)							DeMeR (3 subtasks)						
Decomp.		Merge		Refin.		t <sub>total</sub>	Decomp.		Merge		Refin.		t <sub>total</sub>
t	len	t	len	t	len		t	len	t	len	t	len	
21.4	28	2.1	7	0.1	7	23.6	21.5	19	1.1	7	0.1	7	22.7
396.8	31	2.6	8	0.2	8	399.6	394.0	21	1.7	8	0.2	8	395.9
170.9	33	7.4	19	30.0	9	208.3	519.8	22	2.1	14	6.8	8	528.7
623.7	33	3.8	27	30.0	8	657.5	892.5	23	1.0	15	25.8	8	919.3
41.2	30	1.7	12	0.3	12	43.2	29.7	18	0.8	12	0.3	12	30.8
124.5	34	5.2	21	30.0	14	159.7	624.7	21	0.7	21	30.0	15	655.4
539.1	34	11.5	22	30.0	9	580.6	>1200	-	-	-	-	-	-
>1200	-	-	-	-	-	-	>1200	-	-	-	-	-	-
786.6	32	5.4	28	30.0	9	822.0	>1200	-	-	-	-	-	-
179.4	32	4.7	31	30.0	11	214.1	147.0	20	2.4	15	30.0	8	179.4

## 5 Conclusions

The paper proposed a novel solution method to general ILP consistency problem by boosting an existing algorithm. The new algorithm DeMeR significantly reduces runtime while keeping the size of the resulting hypothesis close to the optimum. The advantage of the proposed method is that it can be applied to any ILP consistency algorithm that may even produce non-optimal solutions in terms of their size. We focused on the crisp problem where the hypothesis strictly separates positive and negative examples. An open question for future research is whether and how this technique can be applied to problems with noisy data where other objectives such as the number of covered and excluded examples are necessary.

**Acknowledgements.** Roman Barták is supported by the Czech Science Foundation under the contract P202/12/G061. The authors thank Filip Železný and Ondřej Kuželka 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., Kuželka, O., Železný, F.: Formulating Template Consistency in Inductive Logic Programming as a Constraint Satisfaction Problem. Technical Report WS-10-08 (WARA), pp. 2–7. AAAI Press (2010)

3. Chovanec, A.: Constraint satisfaction for inductive logic programming. Master Thesis, Charles University in Prague (2011)
4. Ginsberg, M.L., Harvey, W.D.: Iterative broadening. *Artificial Intelligence* 55(2), 367–383 (1992)
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. Muggleton, S., De Raedt, L.: Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19, 629–679 (1994)
8. Plotkin, G.: A note on inductive generalization. *Machine Intelligence* 5, 153–163 (1970)
9. Scheffer, T., Herbrich, R., Wysotski, F.: Efficient Q-Subsumption Based on Graph Algorithms. In: Muggleton, S. (ed.) ILP 1996. LNCS (LNAI), vol. 1314, pp. 212–228. Springer, Heidelberg (1997)
10. Srinivasan, A.: The Aleph Manual (2012), <http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/aleph.html> (accessed January 23, 2012)
11. Wysotski, F., Kolbe, W., Selbig, J.: Concept Learning by Structured Examples - An Algebraic Approach. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI 1981), pp. 153–158. Morgan Kaufmann (1981)

# Learning Teamwork Behaviors Approach: Learning by Observation Meets Case-Based Planning

Banafsheh Rekabdar, Bita Shadgar, and Alireza Osareh

Department of Computer Engineering, Faculty of Engineering,  
Shahid Chamran University, Ahvaz, Iran  
`b-rekabdar@mscstu.scu.ac.ir,`  
`{bita.shadgar,alireza.osareh}@scu.ac.ir`

**Abstract.** Learning collaborative behaviors is an essential part of multi agent systems. One of the suitable techniques for learning collaborative behaviors is observational learning. This paper describes a hybrid method for learning teamwork behaviors from an expert team by observation. More specifically, this paper describes a technique based on implicit knowledge acquired from observational learning and some domain expertise knowledge. In our method an expert team is observed by a team of learners to learn plans and save them in a plan base. Learners then use case-based planning to effectively act based on learned plans in order to imitate the expert team. To evaluate our method a simulated soccer team is developed. We argue that this approach provides a powerful complement to existing teamwork learning methods, specifically in learning complex goal oriented behaviors.

**Keywords:** multi agent learning, case-based planning, observational learning, soccer, teamwork, collaborative behavior, learning collaborative behaviors.

## 1 Introduction

Learning by observation is an effective approach to learn behaviors, without programming them. In this approach a behavior is automatically learned by observing experts in action. Current observational learning methods have some limitations in complex domains. A basic limitation in all observational learning methods is that learner cannot observe any internal state of experts (e.g. beliefs, desires and intentions) [1]. This, results in associating each action of the expert only to the state of the environment, but in complex domains, the experts' actions heavily depend on their private internal states rather than the state of the environment only. Another limitation is the lack of generalization ability to handle radically different unseen situations.

In this paper, a new hybrid learning method is employed in order to learn teamwork behaviors efficiently in an effort to address the mentioned limitations. The proposed method uses, a combination of implicit knowledge acquired by observational learning techniques and some explicit domain knowledge to learn collaborative as well as individual behaviors. The general idea is to learn plans for

tasks from observation and then use the learned plans in a case-based planning method. The explicit domain knowledge helps us identify basic properties of the domain. To show the effectiveness of our method, an implemented prototype for a soccer simulation environment is also introduced and evaluated. Soccer simulation is a suitable environment to put our method to test because the experts exhibit goal-oriented behaviors in it.

The rest of this paper is organized as follows: section 2 provides a brief explanation of previous works. Section 3 explains the proposed method. Experiments and results are presented in section 4 followed by acknowledgment and concluding remarks in Section 5 and 6.

## 2 Background

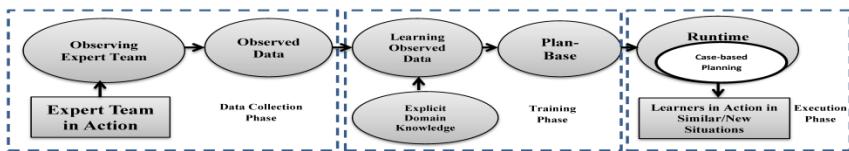
The learning process in humans usually consists of observing other humans and then trying to imitate their behaviors in new situations. The efficiency of observational learning in humans is studied in [2]. Observational learning is an efficient kind of learning in different domains such as training assistant robots [3]. Observational learning requires observational data, which usually contain the state of the environment and the actions of experts. Therefore data-based algorithmic methods are suitable in this context, such as Decision tree (DT) [4], [5], and Artificial Neural Networks (ANNs) [6], [7], etc. All above methods have been used by one learner to learn from one or multiple experts. In [8] a multi-agent framework capable of learning teamwork by observation was presented in domain of Bucket Bridge.

Case-based planning is the application of the Case-Based Reasoning (CBR) methodology [9]. Learning by observation was incorporated with case-based planning [10] in the domain of Real Time Strategy (RTS) games. Authors in [11] incorporate case-based planning with learning by demonstration in RTS games. Floyd *et.al* [12] presents a CBR Framework for developing agents using learning by observation.

## 3 The Proposed Method

This paper plans a hybrid collaborative case-based observational learning method which can learn teamwork behaviors efficiently by combining implicit knowledge acquired by observation and explicit domain knowledge. To present our method several concepts need to be defined. Task is a behavior that agents do intentionally to achieve a goal. It is the base concept in this method which is obtained from explicit domain knowledge. In our work a task is defined with four properties, including: pre and post-conditions, propriety relation, and maintenance property. Pre and post-condition of a task represent a condition where an agent can start a task and where an agent has reached the goal of that task, respectively. The maintenance property represents a condition that if not satisfied in a situation, an agent should stop doing that task. Finally the agent uses priority relation to choose the most important task to start; from the set of active tasks (i.e. tasks with satisfied pre-conditions). The proposed method has three phases; data collection, training and execution. Figure 1 represents the overall procedure of the proposed method. In the data collection phase,

expert team is being observed; the output of this phase is observed data which is stored in some log files. For every expert in the team there is a log file that contains the state of the environment as well as the state of the team followed by the actions of the player in that state. In the training phase, each successful completion of a task is considered as a plan. The agent determines the start and end of plans with pre and post-conditions and it uses the maintenance property to prevent addition of unsuccessful plans to the plan-base. The result of this phase is a plan-base. In the execution phase the agent chooses the current task with respect to pre and post-conditions and the priority relation between tasks. Each task has a corresponding case-based planning system which has a plan retrieval procedure to choose the best plan based on a plan similarity criterion, and an adaptation procedure to reuse plans in a new situation. To solve the current case, case-based planning module searches its plan-base to find the suitable plan according to the plan similarity criterion. It then tailors the plan for the current situation and the agent executes this tailored plan in an effort to accomplish the active task.



**Fig. 1.** Process of the proposed method

## 4 Experiments

To evaluate the performance of the proposed method the domain of soccer game is chosen. This environment is continuous, as both state space and action space are very large. The proposed method is implemented in TeamBots, which is a development and simulation framework for control systems on mobile robots [13]. As the expert team, AIKHomo was chosen who is awarded as the most successful team in a tournament in Johnson's work [14]. However in this work an attempt to imitate AIKHomo team has failed Since AIKHomo is rather complex soccer team and the author switched to a simpler team. AIKHomo team players exhibit complex goal-oriented behaviors. Therefore it is very suitable for our method, because effectively imitating its goal-oriented behaviors, empirically shows the effectiveness of our method. Our method is named LCBO (Learning Collaborative Behavior by Observation). Nine teams are selected as opponents for the tournaments. The expert team and LCBO team compete with these opponents, in 10 different games. In Data collection phase an observation module was added to the original source code for AIKHomo to log states and actions for every player in the team. For every player in the team there is a log file that contains the state of the game and the actions of the player during a single game. Table 1 shows the elements of observed data and their data types in the data set collected by observation. Some of these elements like player heading, number and can kick are private to players and some other elements like ball

**Table 1.** Elements and their data types in the observed data set (State vector)

Data Element	Data type	Data Element	Data type
Time	Integer-Millisecond	Opponent vector[5]	Array of 5 double
Ball vector	Vector	Player Position	Double, Double
Team Goal Vector	Vector	Player Heading	Double
Opponent Goal Vector	Vector	Player Number	Integer-(0-4)
Teammate vector [4]	Array of 4 double	Can Kick	Boolean

position, time and teammates or opponent positions are publicly available. Public elements can be considered as the team state, because they are shared between team members and they can use these elements to imitate the collaborative behaviors in the expert team effectively.

In the training phase the log files from the data collection phase were used to learn plans for tasks. In order to learn from expert team a list of tasks with their properties was extracted from the expert's team behaviors. They are introduced in table 2. The agent starts scanning the corresponding log file that contains observational data sequentially and evaluates the precondition of all defined tasks, until at least a task has satisfied precondition. It then chooses the task with highest priority in case more than one task is active and then adds all the following steps to a plan until the post-condition of the active task is satisfied. This learned plan is added to the plan-base. This work is done by all of the agents in the team, until the end of their log files.

**Table 2.** Tasks name and their corresponding properites

Order	Task name	Pre-condition	Post-condition	Maintenance
1	Move to ball	Be closest to ball among team members	Can kick the ball	Be closest to ball among team members
2	Dribble	Can kick the ball	Kick the ball toward opponent goal	Cannot kick the ball

In the execution phase our soccer team is implemented, so we need to make use of the learned plans, in the context of case-based planning that are available for each task. A case-based planning system needs a similarity criterion for measuring how much a plan is suitable for a situation. The similarity criterion is based on the Euclidean distance between the state vector of the first plan-step and the state vector of current situation. In addition to similarity criterions, a case-based planning system needs an adaptation procedure. The adaptation procedures for each case-based planning system in each on the tasks are shown in table 3.

#### 4.1 Experimental Results

In order to evaluate the performance of our proposed method, we compared the method with alternative teamwork learning method introduced in [14] named COLTS.

In a soccer game the performance of a team is usually evaluated based on the number of goals a team can score, or the number of winning games in a tournament between several teams. However in observational learning methods the aim is imitating the expert team as closely as possible, so the raw performance of a good observational learning method is bounded to the performance of its expert.

**Table 3.** Plan adaption for each task

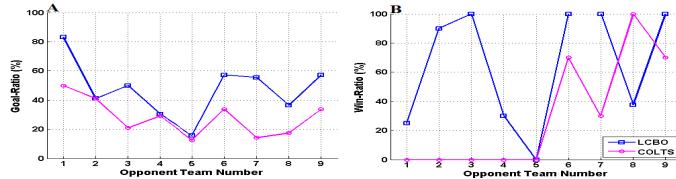
Task name	Plan adaptation
Move to ball	1. Steer heading = Rotate ball vector in current situation with the difference in player heading and ball vector angle in the plan-step. 2. Speed = speed in plan-step. 3. Get next plan-step
Dribble	1. Steer heading = steer heading in plan-step. 2. Speed = speed in plan-step. 3. Get next plan step

Two different success ratios as evaluation criterion named *Goal\_Ratio* and *Win\_Ratio* were defined by equation 1 and 2 to evaluate the performance of the method.

$$Goal\_Ratio(i) = \frac{\sum_{j=1}^n \text{number of goals of team } ij}{\sum_{j=1}^n \text{number of goals of expert team } j} \times 100 \quad (1)$$

$$Win\_Ratio(i) = \frac{\sum_{j=1}^n \text{number of win records of team } ij}{\sum_{j=1}^n \text{number of win records of expert team } j} \times 100 \quad (2)$$

In equation 1 and 2,  $i$  is the opponent number that the proposed method and expert team competed with.  $n$  is the number of games executed with different random seeds. In order to compare our method with COLTS, a competition between LCBO and each of the nine opponent teams were held, in which LCBO played against a team for 10 times (seeds 4 to 13). The same thing was done for COLTS and for LCBO experts who are AIKHomo and SchemaDemo respectively. The *Goal\_Ratio* and *Win\_Ratio* were computed for LCBO and COLTS against all the opponents. As shown in Figure 2A, the *Goal\_Ratio* in LCBO is higher than COTLS for all of the nine teams. This shows that LCBO is more successful in learning from its expert than COLTS. Figure 2B shows the comparison of *Win\_Ratio* for LCBO and COLTS. As shown in Figure 2B the *Win\_Ratios* are also higher than COLTS, except for opponent 8 which is BriSpec. SchemaDemo performs very poorly against BriSpec (1 wins out of 10 games), but AIKHomo is performing much better (8 wins out of 10 games). COLTS have a greater *Win\_Ratio* than our method with this particular team, not because it imitates its expert better, but because its expert performs very poorly.



**Fig. 2.** Results of comparison of Win\_Ratio and Goal\_Ratio

## 5 Conclusion Remarks

In this paper we described a hybrid approach for learning teamwork behaviors by observation. Since learning teamwork behaviors is known as a complex problem, the method is aimed to combine implicit knowledge learned by observation of expert team with domain knowledge to create plan-bases. Plans from the plan-base are then used by a case-based planning method to enable learners to imitate expert team efficiently. To evaluate the performance of the proposed method, TeamBots soccer domain was used. As the result of the experiments shows, it is clear that the proposed method to learn collaborative behaviors by observation gives higher performance than other alternative methods.

**Acknowledgements.** The authors thank the Iran Telecommunication Research Center for their support.

## References

- Allen, J., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., Taysom, W.: Plow: A collaborative task learning agent. In: Proceedings of AAAI 2007, p. 1514. AAAI Press, MIT Press, Menlo Park, Cambridge (2007)
- Bandura, A.: Social-learning theory of identificatory processes. In: Handbook of Socialization Theory and Research, pp. 213–262 (1969)
- Rybski, P.E., Stolarz, J., Yoon, K., Veloso, M.: Using dialog and human observations to dictate tasks to a learning robot assistant. Intelligent Service Robotics 1(2), 159–167 (2008)
- Sammut, C., Hurst, S., Kedzier, D., Michie, D.: 7 Learning to Fly. In: Imitation in Animals and Artifacts, p.171 (2002)
- Isaac, A., Sammut, C.: Goal-directed learning to fly. In: ICMI, p. 258 (2003)
- Rekabdar, B., Joorabian, M., Shadgar, B.: Artificial Neural Network Ensemble Approach for Creating a Negotiation Model with Ethical Artificial Agents. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 493–501. Springer, Heidelberg (2012)
- Sidani, T.A., Gonzalez, A.J.: Learning situational knowledge through observation of expert performance in a simulation-based environment. In: Southcon 1994 Conference Record, pp. 65–70. IEEE (1994)
- Johnson, C.L., Gonzalez, A.J.: Learning Collaborative Behavior by Observation. In: 2010 Ninth International Conference on Machine Learning and Applications (ICMLA), pp. 99–104. IEEE (2010)

9. Hammond, K.J.: Case-based planning: A framework for planning from experience. *Cognitive Science* 14(3), 385–443 (1990)
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. Ontañón, S., Mishra, K., Sugandh, N., Ram, A.: Learning from Demonstration and Case-based Planning for Real-time Strategy Games. In: Prasad, B. (ed.) Soft Computing Applications in Industry. STUDFUZZ, vol. 226, pp. 293–310. Springer, Heidelberg (2008)
12. Floyd, M.W., Esfandiari, B.: A Case-Based Reasoning Framework for Developing Agents Using Learning by Observation. In: 2011 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 531–538. IEEE (2011)
13. Balch, T.: The TeamBots Environment for Multi-Robot Systems Development. Working notes of Tutorial on Mobile Robot Programming Paradigms, ICRA (2002)
14. Johnson, C.L.: A contextual approach to learning collaborative behavior via observation [electronic resource]. University of Central Florida, Orlando (2011)

# Evolutionary Approach to Multiobjective Optimization of Portfolios That Reflect the Behaviour of Investment Funds

Krzysztof Michalak<sup>1</sup>, Patryk Filipiak<sup>2</sup>, and Piotr Lipiński<sup>2</sup>

<sup>1</sup> Institute of Business Informatics,  
Wroclaw University of Economics, Wroclaw, Poland

[krzysztof.michalak@ue.wroc.pl](mailto:krzysztof.michalak@ue.wroc.pl)

<sup>2</sup> Institute of Computer Science,  
University of Wroclaw, Wroclaw, Poland  
[{patryk.filipiak,lipinski}@ii.uni.wroc.pl](mailto:{patryk.filipiak,lipinski}@ii.uni.wroc.pl)

**Abstract.** This paper addresses a problem of finding portfolios that perform better than investment funds while showing similar behaviour. The quality of investment portfolio can be measured using various criteria such as the return and some kind of risk measurement. Investors seek to maximize return while minimizing risk. In order to achieve this goal various instruments are considered. One of the possibilities is to entrust the assets to an investment fund. Investment funds build their own portfolios of stocks, bonds, commodities, currencies, etc.

In this paper we consider the problem of finding a portfolio which outperforms a given investment fund with respect to both the return and the risk and which also behaves in a similar way to the given fund. The rationale behind such an approach is that investment strategies of mutual funds are prepared by experts and are therefore expected to be reasonably good in terms of both the return and the risk. To achieve the presented goal we use a multiobjective evolutionary algorithm with a dedicated "division mutation" operator and a local search procedure. Presented method seems capable of building portfolios with desired qualities.

**Keywords:** investment funds, portfolio optimization, multiobjective evolutionary optimization.

## 1 Introduction

Bio-inspired methods are often applied to various economic and financial problems [15][15] such as supporting financial decision making [10][14][17] and discovering trading rules for stock market speculations [4][11][12]. Optimization of investment portfolios [2] is an important task with practical implications to which evolutionary methods are often employed [13][16]. Usually, there is more than one criterion to optimize. Typically the return and at least one measure of investment risk are considered. In this paper we focused on finding portfolios of

investment instruments that not only maximize the return and minimize the risk but also behave similarly to a given investment fund. Performing such search is motivated by the fact, that we would like to know if in the "neighbourhood" of an investment fund strategy there exist other strategies that can get better results.

In this paper we assume that we have a number of categories  $C_j, j = 1, \dots, N^{(c)}$  of investment instruments, such as stocks, bonds, currencies, etc. Each of the considered instruments  $I_i, i = 1, \dots, N^{(i)}$  belongs to exactly one instrument category  $C_j$ . Therefore, obviously,  $N^{(i)} = \sum_{j=1}^{N^{(c)}} |C_j|$ . We denote the quotations of instrument  $I_i$  by  $q_i(t)$ .

The portfolio is represented as a vector  $w \in R^{N^{(i)}}$  in which each coordinate  $w_i$  is a "weight" of instrument  $I_i$  (the number of units of this instrument bought when the investment is made). We calculate the quotations  $p(t)$  of portfolio  $w$  as:

$$p(t) = \sum_{i=1}^{N^{(i)}} w_i q_i(t) . \quad (1)$$

The quotations for the instruments (and therefore for the entire portfolio as well) are assumed to be available for a time period  $[t_{min}, t_{max}]$ .

There are three criteria which are optimized in this paper: the return measure, the risk measure and the dissimilarity to a given investment fund. The return  $R$  is assumed to be the ratio between the last and the first quotation, so  $R_f = f(t_{max})/f(t_{min})$  for the investment fund,  $R_{I_i} = q_i(t_{max})/q_i(t_{min})$  for instruments and  $R_w = p(t_{max})/p(t_{min})$  for portfolios. As this is essentially a return calculation for a buy-and-hold strategy, trading costs have similar impact on the return of all portfolios and are therefore not taken into consideration in this study. The risk for the investment fund ( $V_f$ ), the instruments ( $V_{I_i}$ ) and portfolios ( $V_w$ ) is measured using the classical measure of Value at Risk [9] calculated for the entire period  $[t_{min}, t_{max}]$ . The dissimilarity  $D_{w,f}$  between portfolio  $w$  quotations  $p(t)$  and investment fund quotations  $f(t)$  is measured using Mean Squared Error (MSE) measure calculated for the entire period  $[t_{min}, t_{max}]$ . We intend to find portfolios which maximize the return  $R_w$ , and minimize both the risk  $V_w$  and dissimilarity  $D_{w,f}$  for a given investment fund  $f$ .

## 2 Proposed Method

Finding portfolios which maximize the return  $R_w$ , and minimize both the risk  $V_w$  and dissimilarity  $D_{w,f}$  for a given investment fund  $f$  requires three-objective optimization. It is well-known that multiobjective optimization can be performed effectively using evolutionary methods [6][18]. One of the reasons for which such methods are particularly useful in this context is that population of solutions can readily be used to represent a Pareto set and a Pareto front of a solved problem. In this paper we propose an evolutionary algorithm described below. The algorithm keeps a subpopulation  $P_{C_j}$  of specimens for each category  $C_j$ ,

$j = 1, \dots, N^{(c)}$  of investment instruments and one global population  $P_{all}$  which, apart from evolving on its own, is also updated by importing specimens from other subpopulations. For a given  $j$ , specimens in subpopulation  $P_{C_j}$  have non-zero weights assigned only to instruments that belong to category  $C_j$  (eg. stocks). All genetic operations and other procedures are implemented to ensure that this condition is met during entire program execution. Algorithm 1 presents details of the evolutionary algorithm used for performing portfolio optimization. In the algorithm  $P_{all}(g)$  and  $P_{C_j}(g)$  denote global and category-specific subpopulations at the generation  $g$ .

The following quantities parameterize the algorithm:

- $N^{(gen)}$  - the number of generations,
- $N^{(pop)}$  - initial size of each subpopulation (global and category-specific),
- $N^{(cross)}$  - the number of crossover operations per generation,
- $P^{(mut)}$  - the percentage of individuals that produce offspring using mutation,
- $P^{(weight\_mut)}$  - the percentage of weights mutated in one specimen.
- $P^{(div\_mut)}$  - the percentage of individuals that produce offspring using the dedicated "division mutation" operator,

---

**Algorithm 1.** Evolutionary algorithm with subpopulations for optimization of portfolios.

---

```

 $P_{all}(1) = InitPopulation(N^{(pop)})$ 
for  $j = 1 \rightarrow N^{(c)}$  do
     $P_{C_j}(1) = InitPopulation(N^{(pop)})$ 
end for
for  $g = 1 \rightarrow N^{(gen)}$  do
    for  $j = 1 \rightarrow N^{(c)}$  do
         $P_{C_j}(g) = P_{C_j}(g) \cup Crossover(P_{C_j}(g))$ 
         $P_{C_j}(g) = P_{C_j}(g) \cup Mutate(P_{C_j}(g))$ 
         $P_{C_j}(g) = P_{C_j}(g) \cup DivMutate(P_{C_j}(g))$ 
         $P_{C_j}(g) = P_{C_j}(g) \cup LocalSearch(P_{C_j}(g))$ 
    end for
    for  $j = 1 \rightarrow N^{(c)}$  do
         $P_{all}(g) = P_{all}(g) \cup P_{C_j}(g)$ 
    end for
     $P_{all}(g) = P_{all}(g) \cup Crossover(P_{C_j}(g))$ 
     $P_{all}(g) = P_{all}(g) \cup Mutate(P_{C_j}(g))$ 
     $P_{all}(g) = P_{all}(g) \cup DivMutate(P_{C_j}(g))$ 
     $P_{all}(g) = P_{all}(g) \cup LocalSearch(P_{C_j}(g))$ 
     $P_{all}(g + 1) = Select(P_{all}(g))$ 
    for  $j = 1 \rightarrow N^{(c)}$  do
         $P_{C_j}(g + 1) = Select(P_{C_j}(g))$ 
    end for
end for

```

---

The algorithm uses the following procedures and operators:

*InitPopulation* - creates a new population consisting of a given number of specimens. Each specimen is a vector  $w \in R^{N^{(i)}}$  in which each coordinate  $w_i$  is drawn with uniform probability from the range  $[0, w_i^{(max)}]$ , where:

$$w_i^{(max)} = \frac{\max_{t \in [t_{min}, t_{max}]} f(t)}{\min_{t \in [t_{min}, t_{max}]} I_i(t)}. \quad (2)$$

*Crossover* - a standard single-point crossover operator .  $2N^{(cross)}$  new specimens are generated by selecting two parents at random from existing population and then splicing the chromosomes at a randomly selected point. These new specimens are then added to existing population.

*Mutate* - creates  $[P^{(mut)} \cdot \text{population\_size}/100]$  new specimens. Each new specimen is generated by selecting at random from existing population one specimen in which  $P^{(weight\_mut)}$  percent of weights are mutated. A single weight is mutated by selecting an integer exponent  $\eta$  with uniform probability from range  $[-6, 6]$ , a real number increment  $\delta$  with uniform probability from range  $[-5, 5]$  and then adding  $\eta \cdot \delta$  to the weight. The new value is then clipped to the range  $[0, w_i^{(max)}]$ , where  $w_i^{(max)}$  is given by the Equation (2).

*DivMutate* - a dedicated mutation operator described below.

*LocalSearch* - a local search procedure described below.

*Select* - selection procedure based on non-domination sorting and crowding distance known from the NSGA-II algorithm . This selection scheme involves performing a series of binary tournaments in which the fitter of two individuals is selected as a winner. Specimen fitness is not directly calculated, but rather the selection is based on the ranks of Pareto fronts to which the specimens belong and the density of population near each of the specimens. In addition to specimens selected in binary tournaments, a set of specimens that outperform the given investment fund with respect to both the return and the risk is copied from one generation to the next. This set may contain at most  $N^{(elite)}$  specimens with the largest crowding distance that are non-dominated by each other (dominated ones are removed).

## 2.1 Dedicated "Division Mutation" Operator

During preliminary experiments we noticed, that if the initial portfolio time series differs significantly from the series of quotations of the investment fund the convergence is very slow. We hypothesized, that this is caused by the fact that specimens are evaluated using three criteria, so even specimens relatively distant from the target time series may survive. We also suspected, that the simple mutation operator described above may not be very effective in generating new specimens that lie far from the original ones. This could be remedied by increasing the probability of mutations or the range of possible weight change. Unfortunately, such approach would cause abrupt changes in the structure of

portfolios, which we would like to avoid. Therefore, we proposed a new genetic operator which acts as mutation and is specific to the problem we tackle.

”Division mutation” operator creates  $[P^{(div\_mut)} \cdot \text{population\_size}/100]$  new specimens. Each new specimen is generated by selecting at random from existing population one specimen and modifying it according to the Algorithm 2.

---

**Algorithm 2.** Division mutation for a single specimen  $w \in R^{N^{(i)}}$ .

---

```

Select at random  $t_0 \in [t_{min}, t_{max}]$ 
 $\delta = p(t_0)/f(t_0)$ 
for  $j = 1 \rightarrow N^{(i)}$  do
     $w_i = w_i/\delta$ 
end for
```

---

From the Equation (1) it is obvious that after the application of this operator the new quotations  $p'(t)$  of the portfolio satisfy a condition:  $\forall_t \cdot p'(t) = p(t)/\delta$ , where  $p(t)$  are the quotations before the application of division mutation. Therefore  $p'(t_0) = p(t_0)/\delta = p(t_0)/(p(t_0)/f(t_0)) = f(t_0)$ , which means, that the ”division mutation” operator ensures that the time series of portfolio quotations intersects with the time series of fund quotations at a randomly selected point. This operator however, scales all the weights by the same factor and thus it leaves the structure of the portfolio intact as opposed to the regular mutation which works on individual weights.

## 2.2 Local Search

Local search is a well-known approach to improving the quality of solutions generated by an evolutionary algorithm [8]. In this paper we introduce a local search procedure which tries to find a solution that improves two objectives by combining solutions that are good with respect to one of the objectives.

In the case of the particular problem of finding portfolios of investment instruments we try to find portfolios that improve both the return and the risk by combining portfolios that give higher returns than a given investment fund or have a lower risk. The local search procedure operates as shown in Algorithm 3. It selects two random samples each containing  $N^{(ls)}$  specimens which are better than the given investment fund with respect to one of the objectives. Then, linear combinations of all pairs of specimens from the two samples are generated. New specimens for which both objectives are better than threshold values  $\theta_1$  and  $\theta_2$  are added to the result set. In this paper we set  $\theta_1 = R_f$  and  $\theta_2 = V_f$ , i.e., specimens that have both higher return and lower risk than the given fund are added to the result set.

If any specimens that improve both the return and the risk are found during local search they are given a chance to produce offspring using division mutation operator. Finally, from the result set  $A$  only these specimens are selected that are not dominated by any specimens in  $A$ .

**Algorithm 3.** Local search procedure.

---

```

 $A = \emptyset$  - a set of specimens found by the local search
 $S_1$  = a random sample of  $N^{(ls)}$  specimens with the objective  $O_1$  better than  $\theta_1$ 
 $S_2$  = a random sample of  $N^{(ls)}$  specimens with the objective  $O_2$  better than  $\theta_2$ 
for  $i = 1 \rightarrow N^{(ls)}$  do
    for  $j = 1 \rightarrow N^{(ls)}$  do
         $w_i$  = weights of specimen  $S_1[i]$ 
         $w_j$  = weights of specimen  $S_2[j]$ 
        for  $\alpha = 0.01 \rightarrow 1$  step 0.01 do
             $w' = \alpha w_i + (1 - \alpha)w_j$ 
             $O'_1$  = objective 1 obtained from  $w'$ 
             $O'_2$  = objective 2 obtained from  $w'$ 
            if  $O'_1$  is better than  $\theta_1$  and  $O'_2$  is better than  $\theta_2$  then
                 $A = A \cup \{ \text{a new specimen with weights } w' \}$ 
            end if
        end for
    end for
end for
 $A = A \cup \text{DivMutate}(A)$ 
 $A = \text{SelectNonDominated}(A)$ 
return  $A$ 

```

---

### 3 Experiments

In the experiments we tested the ability of the presented algorithm to construct portfolios of stocks and currencies that outperform a given investment fund with respect to both the return and the risk at the same time. We performed tests for 18 Polish investment funds from which some invest mainly in stocks and other invest mainly in instruments having a lower risk such as bonds. Details of the funds are summarized in Table 1. The available investment instruments were 317 stocks from Warsaw Stock Exchange and 6 currencies that have exchange rates to Polish currency noted on FOREX market: Swiss franc (CHF), Czech koruna (CZK), Euro (EUR), British pound (GBP), Hungarian forint (HUF) and US dollar (USD). We assumed that the investment is made for a period of half a year from  $t_{min} = 2010.06.30$  to  $t_{max} = 2010.12.31$ . This period equals the interval between semi-annual reports that investment funds are required to publish from which we could learn about the structure of portfolio of instruments used by each of the funds.

In the experiments we ran the algorithm for  $N^{(gen)} = 50$  generations. As there were two categories of investment instruments (stocks and currencies) the algorithm kept three subpopulations (one global and two category-specific) which initially consisted of  $N^{(pop)} = 200$  specimens. The other parameters of the evolutionary algorithm were set to  $N^{(cross)} = 50$ ,  $P^{(mut)} = 200$ ,  $P^{(weight\_mut)} = 50$ ,  $P^{(div\_mut)} = 10$ ,  $N^{(elite)} = 100$ . Local search sample size was set to  $N^{(ls)} = 20$ .

Table 2 summarizes the experiments. This table presents, for each investment fund, the number of generation at which the first portfolio with higher return and lower risk than the given fund was found.

**Table 1.** Details of investment funds considered in the paper

ID	Issuer	Main investment instrument (as reported on 2010.06.30)	Return on $[t_{min}, t_{max}]$	VaR on $[t_{min}, t_{max}]$
96	Millenium	stocks (92.7%)	1.193616	0.009697616
97	Millenium	stocks (59.6%)	1.126392	0.007921864
107	Commercial Union	debt securities (97%)	1.020221	0.0005879808
108	Commercial Union	debt securities (89%)	1.021743	0.00259334
109	Commercial Union	stocks (94%)	1.231499	0.01066058
121	Commercial Union	debt securities (62%)	1.098829	0.003664982
149	Millenium	debt securities (63,6%)	1.081201	0.004770102
159	Commercial Union	debt securities (63%)	1.075789	0.001771456
261	Commercial Union	stocks (51%)	1.13904	0.005648553
1619	Commercial Union	investment funds (50%)	1.056212	0.006571422
2378	Commercial Union	stocks (96%)	1.248263	0.01144357
2379	Commercial Union	stocks (90%)	1.288662	0.006936816
2380	Commercial Union	stocks (80%)	1.250754	0.007484911
2381	Commercial Union	stocks (92%)	1.279447	0.007234238
3463	Millenium	investment funds (81.5%)	0.972389	0.009257582
3466	Millenium	investment funds (86.7%)	1.036602	0.01081419
3467	Millenium	investment funds (80.8%)	0.8973272	0.01727394
3470	Millenium	investment funds (88%)	1.016006	0.01577246

**Table 2.** Number of generation at which the first portfolio with higher return and lower risk than the given investment fund was found

ID	96	97	107*	108*	109	121*	149*	159*	261	1619	2378	2379	2380	2381	3463	3466	3467	3470
Generation $Pop_{all}$	1	1	-	-	1	-	-	-	39	2	1	16	5	3	1	1	1	1
Generation $Pop_{stocks}$	2	12	-	-	2	-	-	-	-	-	3	-	-	-	1	1	1	1
Generation $Pop_{currencies}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-

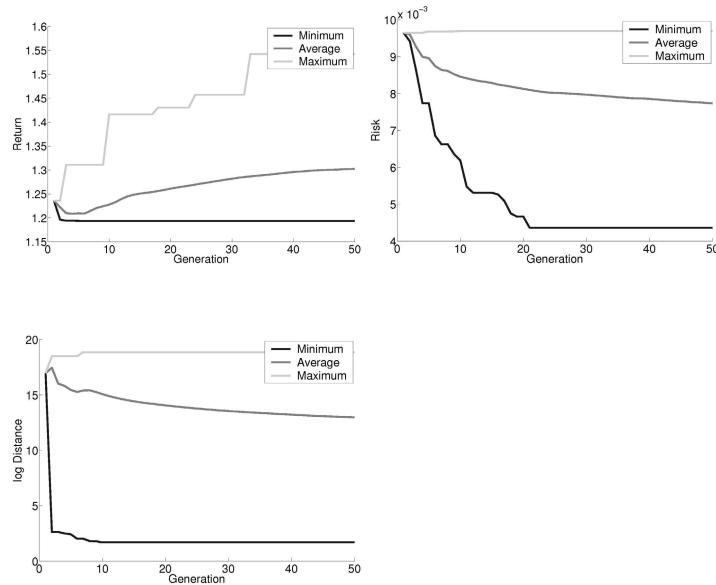
\*Higher return portfolios have been found, but not lower risk portfolios.

As can be seen from the results we were able to find portfolios of both stocks and currencies that outperform investment funds which invest mainly in stocks or other investment funds. It was not possible to outperform funds that invest in debt securities. A comments under Table 2 along with a glance at VaR values in Table 1 explain why. The funds that invest in debt securities have very low risk, which is not surprising as debt securities are known to possess exactly that feature. The algorithm was able to construct portfolios that give higher return, but at the expense of increasing the risk. It is also worth noticing, that the algorithm had not have any low-risk instruments at its disposal (only stocks and currencies were available).

It seems that in some cases the presence of currencies in the investment portfolio was required to outperform the given investment fund. Only in one case however a portfolio containing only currencies was enough to ensure enough return and sufficiently low risk.

In order to visualize the behaviour of the algorithm we include graphs that present minimum, average and maximum values of the three objective throughout all 50 generations of evolution for calculations performed for one of the funds ( $ID = 96$ ). Due to space limitations it is not possible to present similar graphs for all investment funds considered in the tests. Top-left graph in Figure 3 shows values of the return obtained from investment in a given portfolio. Maximum values in this graph reach higher than 1.5, while the return of the fund is less than 1.2, but this might have been achieved at the expense of increasing the risk (which is not shown in this particular graph). In top-right graph in Figure 3 values of Value-at-Risk measure are shown. Again, risk given by individual specimens is more than two times lower than that of the fund, but it might have been achieved at the expense of lower return. Bottom-left graph in Figure 3 presents values of Mean Squared Error calculated on the range  $[t_{min}, t_{max}]$  (2010.06.30-2010.12.31) between portfolio quotations  $p(t)$  and investment fund quotations  $f(t)$ . Due to the large changes in the value of the MSE especially at the beginning of the evolution this graph has a logarithmic scale.

As it can be seen from the graphs in the case of this particular fund the algorithm gradually improves the return for about 35 generations and the risk for about 25 generations. A significant pressure is visible with respect to similarity to the investment fund behaviour.



**Fig. 1.** Minimum, average and maximum values of return (top left), risk (top right) and the logarithm of MSE measuring dissimilarity between fund and portfolio quotations (bottom left) in the population in each of the generations

## 4 Conclusion

In this paper we address a problem of finding portfolios of investment instruments that give higher return and achieve lower risk than a given investment fund. An additional criterion that was expected to be optimized was that the constructed investment strategy should behave similarly to the investment fund. We proposed an multiobjective evolutionary algorithm that optimized the solutions taking into consideration three objectives: investment return, Value-at-Risk as a risk measure and Mean Squared Error as a measure of dissimilarity between the time series of portfolio quotations and the time series of quotations of the given investment fund. In the paper we introduce a new mutation operator aimed at increasing convergence to the baseline time series and a local search procedure which combines solutions which are superior with respect to at least one objective (risk or return) in order to find solutions superior with respect to two criteria. During the tests we found out that the new mutation operator speeds up convergence to the given investment fund time series by generating specimens that intersect with the fund quotations at at least one instant in time.

The experiments were performed using stocks and currencies as possible investment instruments. In the experiments portfolios that outperform investment funds were found in cases when the fund invests mainly in stocks or other investment funds. In the case when the fund invests mainly in debt securities finding portfolios outperforming the fund in minimizing the risk was not possible. This is most probably due to inherently higher risk of stock and currency investments compared to debt securities which are known to have lower investment risk.

In general, however, the proposed algorithm seems to be capable of finding portfolios with required properties, i.e. similar to a given investment fund but with higher return and lower risk, provided that proper investment instruments are available for portfolio construction.

## References

1. Bauer, R.: *Genetic Algorithms and Investment Strategies*. Wiley, Chichester (1994)
2. Best, M.J.: *Portfolio Optimization*. Chapman & Hall/CRC (2010)
3. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) *PPSN 2000*. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
4. Dempsey, I., O'Neill, M., Brabazon, A.: Adaptive Trading with Grammatical Evolution. In: *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, pp. 2587–2592. IEEE, Los Alamitos (2006)
5. Dempster, M., Jones, C.: A Real-Time Adaptive Trading System using Genetic Programming. *Quantitative Finance* 1, 397–413 (2001)
6. Filipiak, P., Michalak, K., Lipinski, P.: Infeasibility Driven Evolutionary Algorithm with ARIMA-Based Prediction Mechanism. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) *IDEAL 2011*. LNCS, vol. 6936, pp. 345–352. Springer, Heidelberg (2011)

7. Haupt, R.L., Werner, D.H.: Genetic Algorithms in Electromagnetics, pp. 37–38. John Wiley & Sons, Inc. (2007)
8. Jaszkiewicz, A.: Genetic local search for multiobjective combinatorial optimization. European J. of Operational Research 137, 50–71 (2002)
9. Jorion, P.: Value at Risk: The New Benchmark for Managing Financial Risk. McGraw-Hill (2006)
10. Li, J., Taiwo, S.: Enhancing Financial Decision Making Using Multi-Objective Financial Genetic Programming. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 2171–2178 (2006)
11. Lipinski, P.: Dependency Mining in Large Sets of Stock Market Trading Rules. In: Pejas, J., Pieglat, A. (eds.) Enhanced Methods in Computer Security, Biometric and Intelligent Systems, pp. 329–336. Kluwer Academic Publishers (2005)
12. Lipinski, P.: Discovering Stock Market Trading Rules Using Multi-layer Perceptrons. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 1114–1121. Springer, Heidelberg (2007)
13. Lipinski, P.: Evolutionary Strategies for Building Risk-Optimal Portfolios. In: Brabazon, A., O'Neill, M. (eds.) Natural Computing in Computational Finance. SCI, vol. 100, pp. 53–65. Springer, Heidelberg (2008)
14. Lipinski, P.: Frequent Knowledge Patterns in Evolutionary Decision Support Systems for Financial Time Series Analysis. In: Brabazon, A., O'Neill, M., Maringer, D.G. (eds.) Natural Computing in Computational Finance. SCI, vol. 293, pp. 131–145. Springer, Heidelberg (2010)
15. Michalak, K., Lipinski, P.: Prediction of high increases in stock prices using neural networks. Neural Network World 15(4), 359–366 (2005)
16. Radziukyniene, I., Zilinskas, A.: Evolutionary Methods for Multi-Objective Portfolio Optimization. In: Proceedings of the World Congress on Engineering 2008, pp. 1155–1159. Newswood Limited (2008)
17. Tsang, E., Li, J., Markose, S., Er, H., Salhi, A., Iori, G.: EDDIE in Financial Decision Making. Journal of Management and Economics 4(4) (2000)
18. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)

# Recurrent Neural Identification and an I-Term Direct Adaptive Control of Nonlinear Oscillatory Plant

Ieroham Baruch<sup>1</sup>, Sergio M. Hernandez<sup>1</sup>, Jacob Moreno-Cruz<sup>1</sup>,  
and Elena Gortcheva<sup>2</sup>,

<sup>1</sup> Department of Automatic Control, CINVESTAV-IPN, Ave. IPN No 2508,  
A.P. 14-740, 07360 Mexico City, Mexico  
[baruch@ctrl.cinvestav.mx](mailto:baruch@ctrl.cinvestav.mx)

<sup>2</sup> Information & Technology Systems Department, Graduate School, University of Maryland  
University College, 3501 University Blvd. East, Adelphi, MD 20783, USA  
[elena.gortcheva@umuc.edu](mailto:elena.gortcheva@umuc.edu)

**Abstract.** A new Modular Recurrent Trainable Neural Network (MRTNN) has been used for system identification of two-mass-resort-damper nonlinear oscillatory plant. The first MRTNN module identified the exponential part of the unknown plant and the second one - the oscillatory part of the plant. The plant has been controlled by a direct adaptive neural control system with integral term. The RTNN controller used the estimated parameters and states to suppress the plant oscillations and the static plant output control error is reduced by an I-term added to the control.

**Keywords:** Modular recurrent neural network; direct adaptive neural control with integral term; nonlinear oscillatory plant, two-mass-resort-damper system.

## 1 Introduction

In the last decade, the Computational Intelligence tools (CI), including Artificial Neural Networks (ANN) and Fuzzy Systems (FS), became universal means for many applications. Because of their approximation and learning capabilities, [1], the ANNs have been widely employed to dynamic process modeling, identification, prediction and control, [2]. Many applications have been done for identification and control of oscillatory mechanical plants too, [2]-[9]. Among several possible neural network architectures the ones most widely used are the Feedforward NN (FFNN) and the Recurrent NN (RNN), [1], [2]. The main NN property namely the ability to approximate complex non-linear relationships without prior knowledge of the model structure makes them a very attractive alternative to the classical modeling and control techniques. This property has been proved for both types of NNs by the universal approximation theorem [1]. The preference given to NN identification with respect to the classical methods of process identification is clearly demonstrated in the solution of the “bias-variance dilemma” [1].

In many cases when the dynamics of the system is nonlinear it is preferable to use RNN to identify the oscillatory mechanical plant, [2]. In [3] it is compared the use of

RNN with the use of an observer to find the torque applied to the plant. In [4] it is proposed to use a RNN and a PI controller so to suppress vibrations of mechanical system driven by linear AC motor. In [5], [6] it is proposed to use a FFNN identifier and a FFNN controller to suppress system sounds and vibrations. In [7] it is applied a RNN for inverse system modeling and feedforward control of mechanical plant. In [8], a second order recurrent neural network is proposed for oscillatory plant approximation. Here the stability of the learning is achieved adding a second order weight difference to the momentum term backpropagation law of learning which made it rather complicated. In [9] a block diagonal RNN is proposed to resolve the approximation of oscillatory plants but the learning of the two dimensional diagonal state blocks needs weight restriction in order to stabilize the learning process. In [10], [11] a new RNN (named Recurrent Trainable Neural Network-RTNN) has been defined, and the vector-matricial Backpropagation (BP), and Levenberg-Marquardt (L-M) algorithms of its learning have been derived using the diagrammatic method. Here the applied RTNN will be extended to Modular RTNN (MRTNN) so to identify better the plant oscillations. The identified by RTNN oscillations are suppressed by Direct Adaptive Neural Control (DANC) system containing I-term. The proposed adaptive control system will be applied for identification and DANC of a two-mass-resort-damper system model taken from [12]. This plant model for us will be an unknown nonlinear oscillatory plant data generator, subject to MRTNN identification. The new point here is the application of the second order optimal L-M learning algorithm, [11], which do not need to use a learning rate constant and do not need to restrict the weights of the feedback matrix "A" so to achieve the MRTNN stability during the learning.

## 2 Topology and BP Learning of the Modular RTNN

The modular RTNN topology has two modules. The first module represents the real (exponential) part of the MRTNN. The second module represents the complex (oscillatory) part of the MRTNN. Using the diagrammatic rules, cited in [11], we could derive the adjoint MRTNN used for learning. Then we could derive the dynamic BP algorithm of its learning based on the MRTNN topology, and its adjoint MRTNN. The MRTNN topology and its BP learning algorithm are described in vector-matrix form as:

$$X(k+1) = AX(k) + BU(k); Z(k) = G[X(k)]; \quad (1)$$

$$V(k) = CZ(k); Y(k) = F[V(k)]; \quad (2)$$

$$X = [X_1; X_2]^T; Z = [Z_1; Z_2]^T; V = [V_1; V_2]^T;$$

$$Y = [Y_1; Y_2]^T; B = [B_1; B_2]^T; C = [C_1; C_2];$$

$$A = \text{block-diag } (A_i), i=1,2; \quad (3)$$

$$\begin{aligned}
A &= \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}; A_1 = diag(\lambda_i); |\lambda_i| < 1; \\
A_2 &= diag(A_{2i}); A_{2i} = \begin{bmatrix} a_{1i} & a_{2i} \\ a_{3i} & a_{4i} \end{bmatrix}; \\
\alpha &= tr(A_{2i})/2 < 1; \beta = \det(A_{2i}) < 1; (\alpha - \beta) < 0
\end{aligned} \tag{4}$$

$$W(k+1) = W(k) + \eta \Delta W(k) + \alpha \Delta W_{ij}(k-1); \tag{5}$$

$$E(k) = T(k) \cdot Y(k); E_1(k) = F'[Y(k)] E(k); F'[Y(k)] = [1 - Y^2(k)]; \tag{6}$$

$$E_3(k) = G'[Z(k)] E_2(k); E_2(k) = C^T(k) E_1(k); G'[Z(k)] = [1 - Z^2(k)]; \tag{7}$$

$$\Delta C(k) = E_1(k) Z^T(k); \Delta B(k) = E_3(k) U^T(k); \tag{8}$$

$$\Delta A_1(k) = E_{31}(k) {}^o X_1(k); \Delta A_2(k) = E_{32}(k) X_2^T(k) \tag{9}$$

$$E = [E_1; E_2]^T; E_1 = [E_{11}; E_{12}]^T; E_2 = [E_{21}; E_{22}]^T; E_3 = [E_{31}; E_{32}]^T$$

where:  $X$ ,  $Y$ ,  $U$  are state, output, and input vectors with dimensions  $n$ ,  $l$ ,  $m$ , respectively;  $Z$  is  $n$ -dimensional output of the hidden layer;  $V$  is  $l$ -dimensional post-synaptic activity of the output layer;  $T$  is  $l$ -dimensional plant output vector, considered as a RTNN reference;  $A$  is  $(nxn)$  state weight matrix;  $B$  and  $C$  are  $(nxm)$  and  $(lxn)$  input and output weight matrices;  $F[.]$ ,  $G[.]$  are vector-valued  $\tanh(.)$ -activation functions;  $F'[.]$ ,  $G'[.]$  are the derivatives of these functions; the state dimensions of both exponential and oscillatory RTNN modules are  $n_1$  and  $n_2$  (this dimension is even), and  $n = n_1+n_2$ ,  $l_1$  and  $l_2$  ( $l = l_1+l_2$ ) are output dimensions of the exponential and oscillatory modules of the modular RTNN. The matrix  $W$  of the BP learning algorithm (5) is a general weight, denoting each weight matrix ( $C$ ,  $A$ ,  $B$ ) to be updated;  $\Delta W$  ( $\Delta C$ ,  $\Delta A$ ,  $\Delta B$ ), is the weight correction of  $W$ ;  $\eta$ ,  $\alpha$  are learning rate parameters (to choose). The diagonal blocks (3) of the state matrix  $A$  corresponds to the exponential and the oscillatory modules of the RTNN. The error vectors  $E$ ,  $E_1$ ,  $E_2$ ,  $E_3$ , predicted by the adjoint RTNN model, have appropriate dimensions. These vectors have also two sub-vectors with appropriate dimensions, corresponding to the output and state dimensions of the exponential and oscillatory RTNN modules. Note that the RTNN topology in fact represented a nonlinear Wiener-Hammerstein model where the states depend linearly on the affine inputs, and the outputs of both layers are nonlinear. This model could approximate various nonlinear plants with different structures, where important thing is that the output of the RTNN follows the output of the plant. The controllability and observability (the identifiability) of this RTNN model is discussed in [11]. The stability of the RTNN model is assured by the activation functions (-1, 1) bounds and by the local stability weight bound condition, given by (4). The first two conditions for the matrix  $A_2$  are stability conditions and the last condition for the matrix  $A_2$  is condition for oscillation existence. The learning of the exponential part of the feedback matrix is an element by element product of two vectors (see (9)).

### 3 Recursive Levenberg-Marquardt Learning of the MRTNN

The application of the recursive L-M algorithm for learning of MRTNN is the new point of this paper. The general recursive L-M algorithm of learning, [11], is given by the following equations:

$$W(k+1) = W(k) + P(k) \nabla Y[W(k)] E[W(k)] \quad (10)$$

$$Y[W(k)] = g[W(k), U(k)] \quad (11)$$

$$E^2[W(k)] = \{Y_p(k) - g[W(k), U(k)]\}^2 \quad (12)$$

$$DY[W(k)] = \frac{\partial}{\partial W} g[W, U(k)] \Big|_{W=W(k)} \quad (13)$$

Where: W is a general weight matrix (A, B, C) under modification; P is a symmetric matrix updated by (16); DY[.] is an nw-dimensional gradient vector; Y is the RTNN output vector which depends of the updated weights and the input; E is an error vector; Yp is the plant output vector, which is in fact the target vector. Using the same RTNN adjoint topology, it is possible to obtain the values of the gradients DY[.] for each updated weight, propagating the value D(k) = I through it. Here the matrices D are sectioned in parts corresponding to the division of the error vectors. Applying (13) the corresponding gradient components are obtained as it is:

$$DY[C_{ij}(k)] = D_{1,i}(k) Z_j(k); \quad DY[A_{ij}(k)] = D_{2,i}(k) X_j(k); \quad DY[B_{ij}(k)] = D_{2,i}(k) U_j(k) \quad (14)$$

$$D_{1,i}(k) = F'_i[Y_i(k)]; \quad D_{2,i}(k) = G'_i[Z_i(k)] C_i D_{1,i}(k) \quad (15)$$

The P(k) matrix was computed recursively by the equation:

$$P(k) = \alpha^{-1}(k) \{P(k-1) - P(k-1) \cdot \Omega[W(k)] S^{-1}[W(k)] \Omega^T[W(k)] P(k-1)\} \quad (16)$$

where the S(.), and Ω(.) matrices are given as follows:

$$S[W(k)] = \alpha(k) \Lambda(k) + \Omega^T[W(k)] P(k-1) \Omega[W(k)] \quad (17)$$

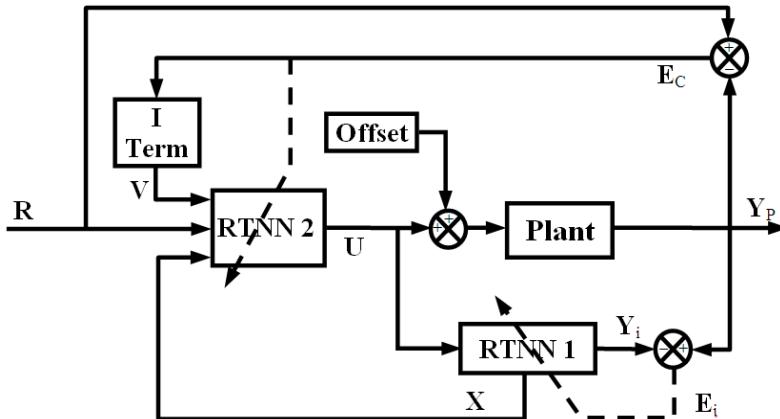
$$\Omega^T[W(k)] = \begin{bmatrix} \nabla Y^T[W(k)] & \\ 0 & \dots & 1 & \dots & 0 \end{bmatrix}; \quad \Lambda(k)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix}; \quad 10^{-4} \leq \rho \leq 10^{-6};$$

$$0.97 \leq \alpha(k) \leq 1; \quad 10^3 \leq P(0) \leq 10^6; \quad i = k \bmod(nw) + 1; \quad k > nw$$

The matrix Ω(.) has dimension (nw×2), whereas the second row has only one unity element (the others were zero). The position of that element is computed by the last statement of (18). After this, a short description of the direct adaptive neural control system with I-term will be given.

## 4 Direct Adaptive Neural Control System with I-Term Design

The block-diagram of the DANC system is given on Fig. 1. It contains a recurrent modular neural identifier RTNN-1, and a modular neural RTNN-2 controller with entries – the reference signal R, the I-term signal, and the state vector X estimated by the modular RTNN-1.



**Fig. 1.** Block diagram of the direct adaptive I-term control

The stable nonlinear oscillatory plant model is described by:

$$X_d(k+1) = f[ X_d(k), U(k) ]; \quad Y_d(k) = g[ X_d(k) ] \quad (19)$$

The input of the plant is perturbed by a constant load perturbation “Offset” which takes into account also the imperfect identification of the plant model. The RTNN-1 and RTNN-2 topologies are given by equations (1)-(4). Let us to linearize the nonlinear equations of the plant, and the modular RTNN 1, RTNN-2, and to introduce the equation of the I-term as:

$$V_I(k+1) = V_I(k) + k_I T_o E_C(k) \quad (20)$$

Where: the dimension of the I-term vector  $V_I(k)$  is equal of the dimension of the error vector, equal of the dimension of the plant output 1. Now we could obtain the following z-transfer functions with respect to  $V_I$ ,  $X$ ,  $R$ , (see Fig. 1):

$$W_p(z) = C_p (zI - A_p)^{-1} B_p \quad (21)$$

$$P_i(z) = (zI - A_i)^{-1} B_i \quad (22)$$

$$Q_1(z) = C_c (zI - A_c)^{-1} B_{cv} \quad (23)$$

$$Q_2(z) = C_c (zI - A_c)^{-1} B_{cx} \quad (24)$$

$$Q_3(z) = C_c (zI - A_c)^{-1} B_{cr} \quad (25)$$

$$I(z) = T_o (zI - I)^{-1} \quad (26)$$

The MRTNN topology is controllable and observable, and the BP/L-M algorithms of learning are convergent, [11]. Then the identification and control errors tend to zero ( $E_i(k) = Y_p(k) - Y(k) \rightarrow 0$  and  $E_c(k) = R(k) - Y_p(k) \rightarrow 0$ ;  $k \rightarrow \infty$ ). This means that each transfer function given by equations (21)-(25) is stable with minimum phase. The z-transfer functions (21)-(25) are connected by the next equation, derived following the block-diagram of the Fig. 1, and given in z-operational form:

$$\{I + W_p(z) [I - Q_2(z) P_i(z)]^{-1} Q_1(z) I(z)\} Y_p(z) = \\ = W_p(z) [I - Q_2(z) P_i(z)]^{-1} [Q_1(z) I(z) + Q_3(z)] R(z) + W_p(z) O_f(z) \quad (27)$$

Substituting (26) in (27), finally we obtained:

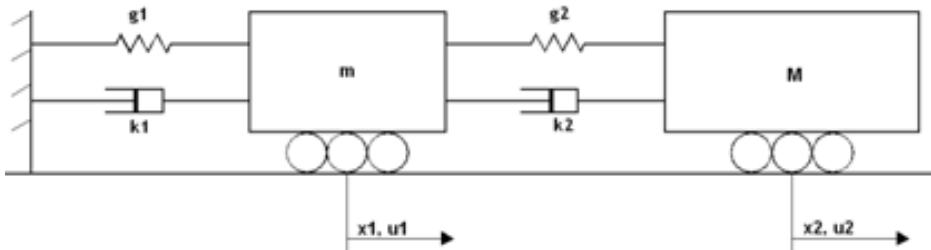
$$\{(z-1)I + W_p(z) [I - Q_2(z) P_i(z)]^{-1} Q_1(z) T_o\} Y_p(z) = \\ = W_p(z) [I - Q_2(z) P_i(z)]^{-1} [Q_1(z) T_o + (z-1) Q_3(z)] R(z) + W_p(z) (z-1) O_f(z) \quad (28)$$

The equation (28) show that the closed-loop system is stable ( $Y_p(k) \rightarrow R(k)$ ;  $k \rightarrow \infty$ ) and that the I-term eliminated the constant perturbation  $O_f$  because the last term tended to zero when  $z$  tended to 1. In the case when we deal with a system with excessive measurements ( $l > m$ ), the DANC performed a data fusion so to elaborate the control action. So we need the plant input error to train the RTNN-2 controller. An approximated way to obtain the input error from the output error is pre-multiplying it by the  $(CB)^+$  using the estimated by the RTNN-1, C,B-matrices, which gives:

$$E_u(k) = (CB)^+ E_c(k), (CB)^+ = [(CB)^T (CB)]^{-1} (CB)^T \quad (29)$$

## 5 Analytical Model of the Oscillatory Plant

In this work as a stable input-output data generator we used a two-mass-resort-damper nonlinear system model, [12], shown in Fig.2, with parameters given on Table 1.



**Fig. 2.** Sketch of the two-mass-resort oscillatory plant with damping

The analytical model of this oscillatory system is given by the following equations:

$$\dot{x}_1 = x_2 \quad (30)$$

$$\dot{x}_2 = -\frac{1}{m} [g_1(x_1) + k_1 x_2 + g_2(x_1 - x_3) + k_2(x_2 - x_4)] + \frac{u_1}{m} \quad (31)$$

$$\dot{x}_3 = x_4 \quad (32)$$

$$\dot{x}_4 = -\frac{1}{M} [g_2(x_3 - x_1) + k_2(x_4 - x_2)] + \frac{u_2}{M} \quad (33)$$

Where, following the Fig.2 -  $x_1$  and  $x_3$  are the two mass positions along the axe x;  $x_2$  and  $x_4$  are the two mass velocities; the two masses are denoted by m and M, respectively;  $k_1$  and  $k_2$  are damping constants, and the resort forces are given by:

$$\begin{aligned} g_1(\epsilon) &= a_1 + b_1 \epsilon^3 \\ g_2(\epsilon) &= a_2 - b_2 \epsilon^3 \end{aligned} \quad (34)$$

**Table 1.** Parameters of the Plant

Parameter	Value
$k_1$	5
$k_2$	3
M	10 kg
m	20 kg
$a_1$	3
$a_2$	2
$b_1$	1.5
$b_2$	-2.0

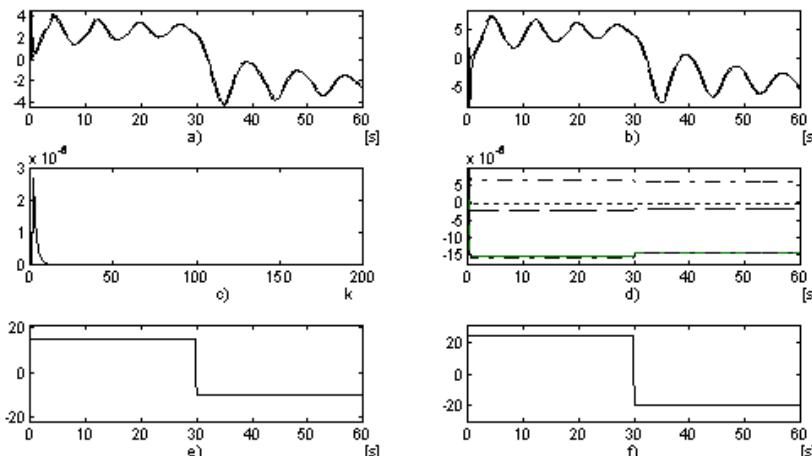
## 6 Simulation Results

The system Identification is performed by a modular RTNN. The topology of the modular RTNN is (2, 5, 2), where one block with state dimension 1 and two blocks with state dimension 2 are used. The activation functions are  $\tanh(\cdot)$  for both layers of the RTNN. The learning rate parameter of L-M is  $\rho = 4$ . The plant model, used as an input/output data generator is given by equations (30)-(34). The simulation results of

MRTNN system identification are obtained on-line during 60 seconds (with step To=0.3 s) using both BP and L-M learning. Both identification inputs are square waves. The graphical simulation results obtained by L-M learning are given on Fig. 3. The results show that the outputs of the MRTNN follow perfectly the outputs of the plant. The quality of learning has been studied using some statistics obtained repeating 20 times the learning experiment and computing the final value of the Means Squared Error (MSE) of learning. The statistical values are computed by the following formulas:

$$\bar{\epsilon} = \frac{1}{n} \sum_{k=1}^n \xi_{avk}; \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n \Delta_i^2}; \Delta = \xi_{av} - \bar{\epsilon} \quad (35)$$

Where:  $\bar{\epsilon}$  is the mean value of the MSE;  $\sigma$  is the standard deviation;  $\Delta$  is the deviation with respect to the mean. The final MSE for 20 runs of the L-M learning algorithm is given on Table 2. The final MSE for 20 runs of BP learning algorithms maintained values around 8.67E-06. The Table 3 gives the values of  $\bar{\epsilon}$  and  $\sigma$  obtained by BP and L-M learning algorithms. The comparative results showed inferior MSE,  $\epsilon$ , and  $\sigma$  for the L-M algorithm of learning with respect to the BP one.



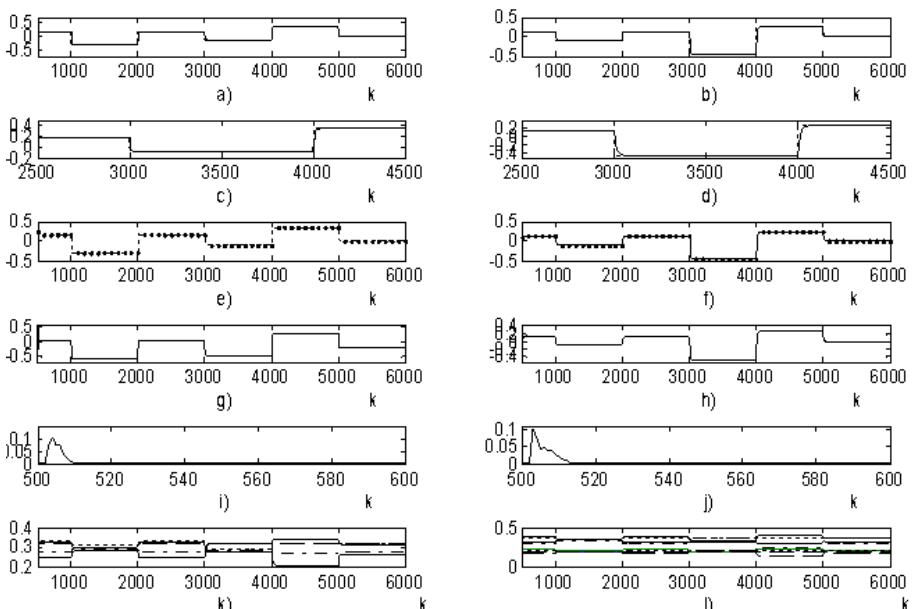
**Fig. 3.** Graphical simulation results of MRTNN system identification obtained by L-M learning algorithm; Continued line-plant output, Pointed line-MRTNN output; a) Position of the mass m; b) Position of the mass M; c) MSE; d) Estimated MRTNN states; e) First plant input; f) Second plant input

**Table 2.** Final MSE% of 20 runs of the identification program using the RTNN L-M algorithm

No	1	2	3	4	5
MSE%	3.10670E-06	3.10671E-06	3.10666E-06	3.10670E-06	3.10670E-06
No	6	7	8	9	10
MSE%	3.10670E-06	3.10671E-06	3.10671E-06	3.10674E-06	3.10670E-06
No	11	12	13	14	15
MSE%	3.10671E-06	3.10669E-06	3.10673E-06	3.10666E-06	3.10667E-06
No	16	17	18	19	20
MSE%	3.10668E-06	3.10678E-06	3.10671E-06	3.10670E-06	3.10668E-06

**Table 3.** Standard deviations and mean average values of identification validation using the BP and L-M algorithms of RTNN learning

BP algorithm	L-M algorithm
$\varepsilon = 8.6712E-06$	$\varepsilon = 3.1067E-06$
$\sigma = 3.0235E-08$	$\sigma = 2.6800E-08$

**Fig. 4.** Graphical simulation results of I-term DANC using L-M learning algorithm; Continued line-plant output, Pointed line-system reference; a) Position of the mass m (total time); b) Position of the mass M (total time); c) Position of the mass m (partial time); d) Position of the mass M (partial time); Closed loop system identification (Plant output vs. MRTNN-1 output); e) Position of the mass m (total time); f) Position of the mass M (total time); g) First control; h) Second control; i) MSE of identification; j) MSE of control; k) States of the identification MRTNN-1; l) States of the control RTNN-2.

The real-time I-term DANC (see Fig.1) contained MRTNN-1 identifier, as used before, RTNN-2 controller, and I-term. The MRTNN-1 and RTNN-2 are learned by the L-M algorithm with topology (2, 5, 2). The simulation results of DANC are obtained on-line during 10,000 iterations. The Fig. 4 showed the control results, for the total time and shorter time of learning, respectively, the MSE of the control system and the MSE of closed loop identification. The Fig. 4 showed the MRTNN-1 and RTNN-2 states as well. Here the final MSE is obtained also for 20 runs of the L-M learning algorithm and the whole control program. The obtained computed statistics are  $\epsilon = 1.3532E-05$ , and  $\sigma = 7.6576E-07$ . Some results are obtained controlling noisy plant using DANC without I-term. The results show that the control system could not eliminate the static error occurred by a constant input noise.

## 7 Conclusions

The paper proposed to use a new MRTNN for system identification and direct adaptive neural control of a two-mass-resort-damper oscillatory plant. The first MRTNN module identified the exponential part of the unknown plant and the second one- the oscillatory part of the plant. The RTNN controller used the estimated states to suppress the plant oscillations and to follow the system reference. The constant plant noise is eliminated by an I-term added to the system. The good quality of the proposed DANC system is confirmed by simulation results obtained with a two-mass-resort-damper oscillatory plant which mathematical model is taken from the literature. The graphical and numerical (MSE) results show a fast L-M convergence, and great DANC precision.

**Acknowledgments.** The MS student Jacob Moreno Cruz and the Ph.D. student Sergio-Miguel Hernandez Manzano are thankful to CONACYT, Mexico for the scholarships received during their studies in the Department of Automatic Control, CINVESTAV-IPN, Mexico City, MEXICO.

## References

1. Haykin, S.: Neural Networks, a Comprehensive Foundation, 2nd edn., sec. 2.13, 4.13, pp. 84–89, 208–213. Prentice-Hall, Upper Saddle River (1999)
2. Dreyfus, G.: Neural Networks: Methodology and Applications, 2nd edn. Springer, Heidelberg (2005)
3. Calderon, G., Draye, J.: Nonlinear Dynamic System Identification with Dynamic Recurrent Neural Networks. In: Proc. of the International Workshop on Neural Networks for Identification, Control, Robotics and Signal/Image Processing, pp. 49–54 (1996) ISBN: 0-8186-7456-3
4. Song, J., Lee, K., Choi, J.: Vibration Control of Two Mass System Using a Neural Network Torsional Torque Estimator. In: Proc. of the 24th Annual Conference of the IEEE, IECON 1998, vol. 3, pp. 1785–1788 (1998) ISBN: 0-7803-4503
5. Yousefi, H., Hirvonen, M., Handroos, H., Soleymani, A.: Application of Neural Network in Suppressing Mechanical Vibration of a Permanent Magnet Linear Motor. Control Engineering Practice (16), 787–797 (2007), doi:10.1016/j.conengprac

6. Bouchard, M.: New Recursive Least-Squares Algorithms for Nonlinear Active Control of Sound and Vibration Using Neural Networks. *IEEE Transactions on Neural Networks* 12(1) (January 2001)
7. Luis, G., Teixeira, R., Ribero, J.: A Neural Network Based Direct Inverse Control for Active Control of Vibrations of Mechanical Systems. In: Proc. of the IEEE VI Brazilian Symposium on Neural Networks, pp. 107–112 (January 2000) ISBN: 0-7695-0856-1
8. Kazemy, A., Hosseini, S.A., Farrokhi, M.: Second Order Diagonal Recurrent Neural Network. In: IEEE International Symposium on Industrial Electronics, ISIE, pp. 251–256 (2007), doi:10.1109/isie.2007.4374607
9. Govindhasamy, J.J., McLoone, S.F., Irvin, G.W.: Second Order Training of Adaptive Critics for Online Process Control. *IEEE Transactions on Systems, Man, and Cybernetics-part B: Cybernetics* 35(2) (April 2005), doi:10.1109/TSMCB.2004.843276
10. Baruch, I.S., Garrido, R.: A Direct Adaptive Neural Control Scheme with Integral Terms. *International Journal of Intelligent Systems, Special Issue on Soft Computing for Modelling, Simulation and Control of Nonlinear Dynamical Systems* 20(2), 213–224 (2005) Castillo, O., Melin, P. (guest eds.) ISSN 0884-8173
11. Baruch, I.S., Mariaca-Gaspar, C.R.: A Levenberg-Marquardt Learning Applied for Recurrent Neural Identification and Control of a Wastewater Treatment Bioprocess. *International Journal of Intelligent Systems* 24, 1094–1114 (2009) ISSN 0884-8173
12. Sira-Ramírez, H., Márquez, R., Rivas-Echeverría, F., Llanes-Santiago, O.: Control of Nonlinear Systems, 1st edn. Pearson Prentice Hall, Madrid (2005) (in Spanish) ISBN 84-205-4449-3

# Towards Automatic Structure Analysis of Digital Musical Content

Adrian Simion and Ștefan Trăușan-Matu

University “Politehnica” of Bucharest  
Computer Science Department  
Splaiul Independenței nr. 313, sector 6, 060042,  
Bucharest, Romania  
simion.adrian@gmail.com, stefan.trausan@cs.pub.ro

**Abstract.** The intuitive mode of structuring melodies by humans is very hard to reproduce in the context of an automated method. The human brain can differentiate between the pitch, timbre and the attack of a musical note even if the listener doesn't have prior knowledge of musical theory; the successions of these notes could easily be the base for recognizing various sections of a song. This paper tries to give some insight in the problem of automatic structuring of musical content, by applying some techniques of machine learning. The experiment followed a TOP-DOWN approach by applying the algorithms on 7 different genres, afterwards on one album of a particular genre and in the end on a single audio file of the same genre. After the automatic structure analysis was performed the accuracy of the results was tested by a performance evaluator and by a human component.

**Keywords:** Machine Learning, Automatic structure analysis, Data mining and data analysis, Neural networks.

## 1 Introduction

The analysis of the large musical corpus that exists today in digital format cannot be done in a feasible manner only by listeners so an automated method has to be found. The various needed tasks could be done with more perseverance by an automated system, as a consequence cheaper and more secure. By letting machines doing automated tasks that could be performed by humans we discover that this task alone reiterates a problem raised by scientists like John McCarthy since the concretization of the term “artificial intelligence” and the electronic means of processing data: “The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.” [9]

We believe that the structure analysis done by the automated system has to resemble closely with the human listener's input. We do not dive into more philosophical issues like “What is music?” or “How can we make a machine to

differentiate Static noise from the Moonlight sonata” and we take well known musical excerpts from various genres and try to apply machine learning techniques in order to obtain relevant structure analysis.

Christopher M. Bishop did an extensive study on machine learning techniques [3]. We chose this study as reference for the algorithms and models used in this paper. Among the algorithms we mention: Random Forest [11], One Rule [4], Naïve Bayes [12], Support Vector Machine [5], Nearest Neighbor [2], Neural Networks [5] and Hidden Markov model [7].

The paper continues with a description of the test set used. Section 3 introduces the experiment and the next section presents the results. Section 5 contains the results and details about the further developments.

## 2 Description of the Test Set

We chose a dataset corpus that is relevant in the eventuality of refining the methods and building an application that can be used online by the MIR [6] community. Thus 21 audio files from the early stages of the MARSYAS [8] framework, were selected, these files being grouped in the following genres (three files each genre): Blues, Country, Disco, Hip-Hop, Metal, Pop and Rock.

The next dataset component contains audio data of a classical rock band AC/DC, specifically the album TNT (1975). This album consists of 9 tracks.

Lastly the algorithms were put to test on another rock track “Angie” by The Rolling Stones.

The particular choice of these tracks was done in order to test the performance of the algorithms from a more general perspective and understand how these algorithms perform when applied using a TOP-DOWN approach.

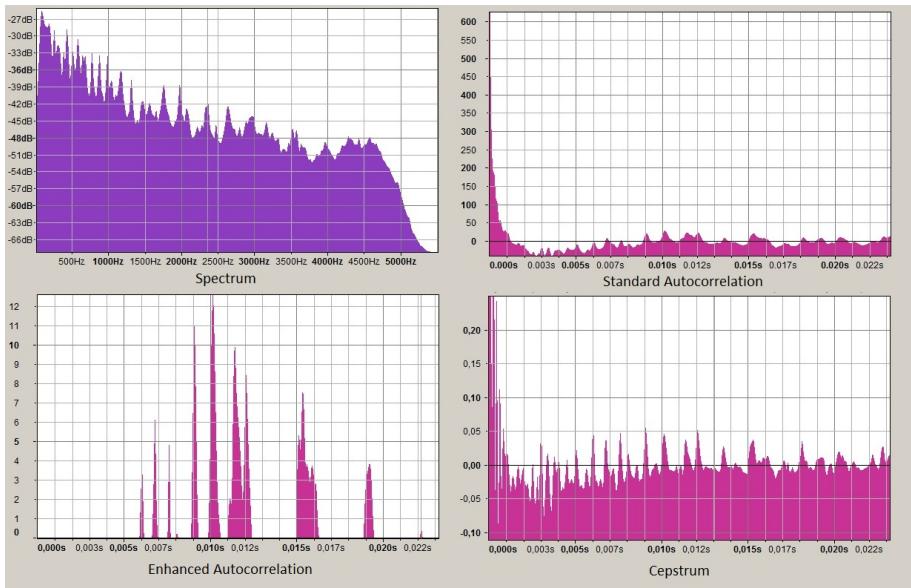
All audio was down-sampled to 22050 Hz, using a 32 bit float range and one channel (mono).

## 3 Experiment

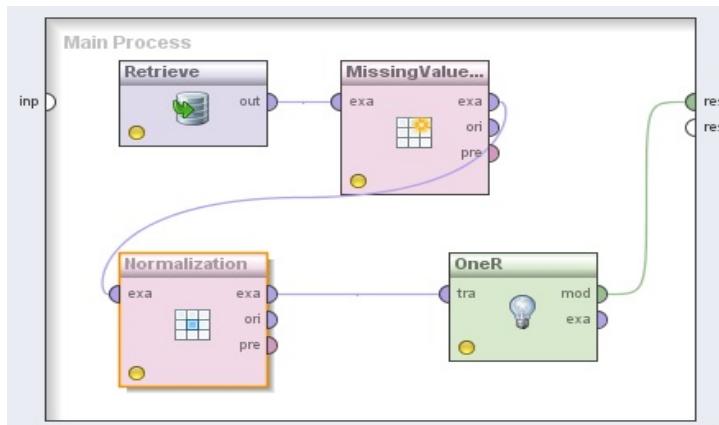
We decided to use the following tools: Audacity [1] and Vamp [13] plug-ins for audio spectral analysis and export of the digital information, and Rapid Miner [10] open-source system for data analysis and machine learning.

The algorithms used in frequency analysis were: Spectrum (normal spectrum envelope), Standard Autocorrelation, Enhanced Autocorrelation and Cepstrum, all with the rectangular window function of size 512.

In Figure 1 you can observe the results of application of the four algorithms to the song “Angie” written by the Rolling Stones band:



**Fig. 1.** Frequency analysis algorithms for the experiment



**Fig. 2.** Example of an experiment using the OneR classifier

After the down-sampling process the data is loaded into the Rapid Miner environment and the following classifiers are used: RandomForest, OneR, NaiveBayes, SVM, NearestNeighbors and NeuralNet.

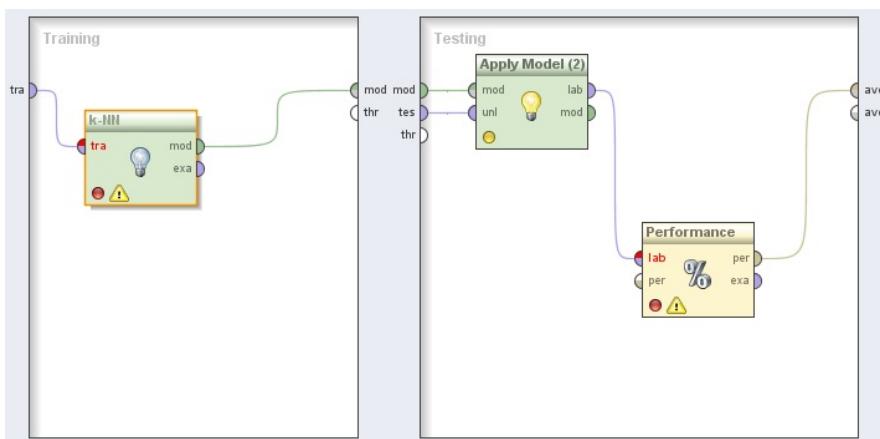
Figure 2 depicts the visual flow of the predicted data in the Rapid Miner system:

For all the classifiers we have calculated the: Root Mean Squared Error in order to have an idea of what classifier best fits a given data sample.

The second part of the experiment was conducted by applying models on the data exported from Audacity. This data was exported in an excel file that was edited by 7 different listeners in order to annotate the structure, more precisely a new column has been added to the sheet, besides the frequency and Level, that marks the change of the musical structure. All listeners heard the musical excerpt two times prior to the actual annotation, which was done by simple signaling. The human input was performed on the first file of the chosen 3 of each genre and the data acquired has been used to train the models.

In this experiment we chose the following models: k-Nearest Neighbor, Naïve Bayes, Feed Forward Neural Network - back propagation algorithm, Support Vector Machine and k-Means Hidden Markov Model.

Of course each model had to be tweaked as the problems of over-fitting and under-fitting occurred. This could be done by adjusting various parameters of each model but a more complex and satisfying solution was given by the cross-validation method. The next figure shows how this method can be applied with a Performance Evaluator in the Rapid Miner environment:



**Fig. 3.** Cross-validation on k-NN model with a Performance Evaluator

In the following section are found the results of these experiments and in the final part some conclusions are presented as well as some suggestions for future improvement of the machine learning techniques applied in these particular cases.

## 4 Results

After applying the algorithms we obtained the following results presented in Tables 1-3.

**Table 1.** Root Mean Squared Error averaged on MARSYAS Genres input

Class Alg.	RForest	OneR	NaiveBayes	SVM	NN	N-Net
Spectrum	0.082	0.072	0.080	0.076	0.055	0.087
Standard AC	0.084	0.080	0.075	0.061	0.077	0.067
Enhanced AC	0.085	0.076	0.039	0.052	0.056	0.055
Cepstrum	0.084	0.083	0.083	0.068	0.917	0.078

**Table 2.** Root Mean Squared Error averaged on ACDC –TNT Album input

Class Alg.	RForest	OneR	NaiveBayes	SVM	NN	N-Net
Spectrum	0.072	0.082	0.077	0.073	0.053	0.081
Standard AC	0.074	0.090	0.074	0.060	0.074	0.062
Enhanced AC	0.082	0.075	0.060	0.040	0.062	0.083
Cepstrum	0.064	0.077	0.096	0.066	0.100	0.077

**Table 3.** Root Mean Squared Error on the track “Angie” by the Rolling Stones

Class Alg.	RForest	OneR	NaiveBayes	SVM	NN	N-Net
Spectrum	0.061	0.080	0.075	0.080	0.054	0.084
Standard AC	0.078	0.077	0.078	0.062	0.072	0.061
Enhanced AC	0.077	0.072	0.068	0.055	0.062	0.085
Cepstrum	0.068	0.073	0.083	0.064	0.85	0.078

By looking at the results in the above tables we can see that from the first stage of the experiment where all the genres are analyzed and the classifiers become more specialized to the second one where we put to test the trained models, the RForest, SVM and the Neural Network have a smaller Root Mean Squared Error which makes them good candidates for further research. Transitioning to the final stage it seems that only the Neural Network keeps the descending trend.

Further experimentation led to the use of the following models on the entire corpus: k Nearest Neighbor, Naïve Bayes, Neural Network, Support Vector Machines, k Hidden Markov Model with the following performance metrics:

**Table 4.** Average model accuracy calculated with X-Validation performance estimator

Model Alg.	k-NN%	NB%	N-Net%	SVM%	k-HMM%
Spectrum	85.61	94.51	94.51	94.51	87.61
Standard AC	87.61	94.51	94.51	94.51	94.51
Enhanced AC	88.24	94.51	93.5	87.61	94.51
Cepstrum	87.61	94.51	94.51	93	93.26

We can draw the conclusion that Naïve Bayes modeling produces constant accuracy which makes it a good candidate in a generic approach, not the TOP-Down presented previously.

Figure 4 depicts the vector table from the analysis made on the blues000 file of the MARSYAS dataset.

counter	label	function value	alpha	abs(alpha)	support vect...	Lag (secon...	Level
0	No	0	0.100	0.100	support vect	-1	1
1	No	0	0.100	0.100	support vect	-0.921	-0.957
2	No	0	0.100	0.100	support vect	-0.898	-0.950
3	No	0	0.100	0.100	support vect	-0.882	-0.978
4	No	0	0.100	0.100	support vect	-0.874	-0.997
5	No	0	0.100	0.100	support vect	-0.835	-0.974
6	No	0	0.100	0.100	support vect	-0.780	-1
7	Yes	0	0.100	0.100	support vect	0.929	-0.973
8	No	0	0.100	0.100	support vect	0.937	-0.974
9	No	0	0.100	0.100	support vect	0.953	-0.961
10	No	0	0.100	0.100	support vect	0.961	-0.964
11	No	0	0.100	0.100	support vect	0.984	-0.969
12	No	0	0.100	0.100	support vect	0.992	-0.963
13	No	0	0.100	0.100	support vect	1	-0.967
14	No	0	-0.100	0.100	support vect	-0.976	-0.890
15	No	0	-0.100	0.100	support vect	-0.819	-0.976

**Fig. 4.** Support Vector Table from the evaluation of blues000.au file

## 5 Conclusions and Future Development

In this paper we tried to test and refine various Machine Learning algorithms in order to achieve an automatic structure analysis of the digital musical content. The results obtained can be used in information retrieval tasks such as genre identification of a particular track or automatic indexing of large audio databases. The novelty consisted in using a TOP-DOWN approach for the training of the various models used and trying to solve the problems of over-fitting and under-fitting as well as testing the results with the aid of a few annotated files by totally different listeners.

The approach presented here can be improved by using cross-validation and by using a larger base of audio input and more thorough testing done by human means. For future development we plan to develop an online platform for these tasks where the community can test and improve the consecrated machine learning methods.

## References

1. Audacity, <http://audacity.sourceforge.net/>
2. Besag, J.E.: Nearest-Neighbor Systems and the Auto-Logistic Model for Binary Data. Journal of the Royal Statistical Society. Series B (Methodological) 34(1), 75–83 (1972)

3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer Science Business Media (2006)
4. Buddhinath, G., Derry, D.: A Simple Enhancement to One Rule Classification, Department of Computer Science & Software Engineering, University of Melbourne Melbourne, Australia 3010 (2007)
5. Cauwemberghs, G., Poggio, T.: Incremental and Decremental Support Vector Machine Learning. In: Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference (2000)
6. Downie, J.S.: Introduction to Music Information Retrieval Research: Past, Present and Future (2007)
7. Eddy, S.R.: Profile hidden Markov models, Department of Genetics, Washington University School of Medicine, 4566 Scott Avenue, St Louis, MO 63110, USA. Oxford University Press (1998)
8. MARSYAS, <http://marsyas.info/>
9. McCarthy, J., Minsky, L.M., Rochester, N., Shannon, C.E.: A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence (1955)
10. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid Prototyping for Complex Data Mining Tasks. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2006 (2006)
11. Montillo, A.A.: Statistical Foundations of Data Analysis, University of Pennsylvania, Radiology Rutgers University, Computer Science (2009)
12. Murphy, K.P.: Naive Bayes classifiers, Technical Report (2006)
13. Vamp Plugins, <http://vamp-plugins.org/>

# Methodology for a New Agent Architecture Based on the MVC Pattern

Yassine Gangat, Denis Payet, and Rémy Courdier

University of La Réunion, LIM

Saint-Denis, Island of La Réunion

{yassine.gangat,denis.payet,remy.courdier}@univ-reunion.fr

**Abstract.** In the last few years, the multiagent system's paradigm has been more and more used in various fields, specially in the simulation field (MAS). Whenever a new application came into being and has been validated by its review board, specialists usually want to reuse it, fully or partially, in order to cut down the time and price of developing similar application.

But this reuse is not as simple as expected. In a previous article, we proposed the DOM modeling to tackle modeling difficulties which arise in a complex system. However this solution has its limits as we will develop here. In this paper, we define a more complete agent modeling, based on the MVC design pattern, in order to push back these limits.

**Keywords:** multi-agent, behaviors, MVC, aMVC, design pattern.

## 1 Introduction

### 1.1 Context

In the last few years, the multiagent system's (MAS) paradigm has been more and more used in various fields, especially in the simulation field. While some applications are used for pedagogic purposes, others are made in order to provide decision-support tool, e-commerce application, etc. which implies a wide range of complexity's level. Like any new paradigm, the MAS's wide-spreading requires new models, new methodologies and new softwares to support development engineers with robust and reliable applications.

Whenever a new application came into being and has been validated by its review board, specialists usually want to reuse it, fully or partially, in order to cut down the time and price of developing similar application. But this reuse is not as simple as expected.

The idea of Dynamic-Oriented Modeling (DOM) [1] was also born of this desire of model's reusing. In that previous article we proposed the DOM modeling to tackle modeling difficulties which arise in a complex system. These difficulties mainly consist in the fact that multiagents systems are becoming more and more hard to model due to the complexity of the system studied. After having implemented DOM on multiple application, we realized that DOM was not enough to

overcome all of these difficulties, especially because it doesn't care about agents' behaviors.

Indeed, to complete this one, in this paper, we study some example and propose a new behavioral model based on the well-attested MVC model. It is an original reuse of this well known pattern to obtain a new modelization approach for modularizing not only the development of environment but the development of agents as well.

As such, we first start by breaking up an agent according to some axis. Then we will recompose this splitted agent in a MVC-like pattern (aMVC) to define our new modeling proposal: the Multi-Behaviors Modelization.

Lastly, we conclude by giving a few perspectives of this work.

## 2 From the Breaking Up of an Agent...

This new approach called Multi-Behaviors Modelization is based on the splitting of the agent into severals pieces. The first step has already been presented in a previous paper [1]. After that first one, the next steps are phases that will allow us to build an MVC based agent.

### 2.1 Splitting of the Agent According to Dynamics

When we say "splitting of the agent according to dynamics", we mean environment splitting reflected on the agents. In a previous paper [1], we have presented a modeling method DOM (Dynamic-Oriented Modeling) based on dynamics, where we started with a basic agent (*c.f.* Figure 1a).

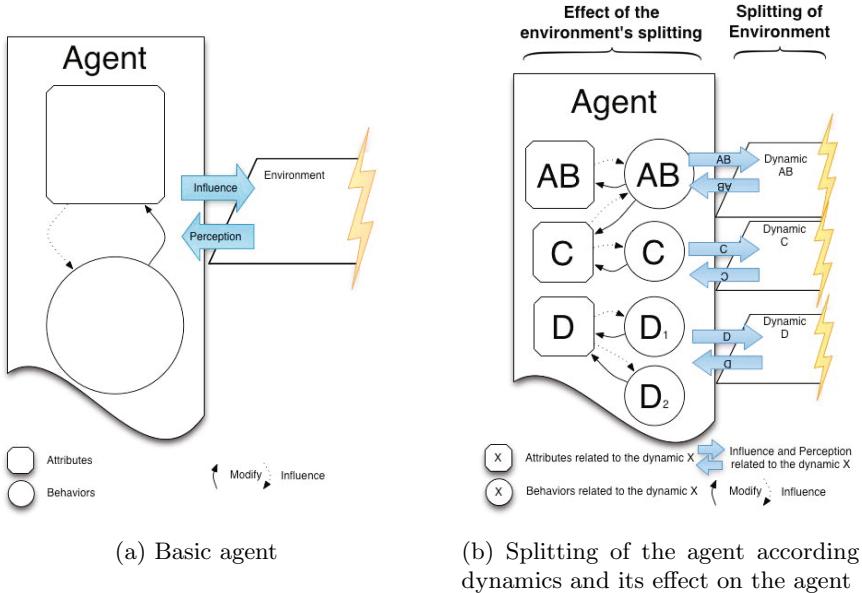
A "**dynamic**" is an association of a set of activities that participate in a major characteristic of a complex problem.

The aim of this modeling method is to break down a complex problem into some less complex parts (*i.e.* dynamics), using the environment (which is the location where agents evolve) as the coupling element for these dynamics in the Figure 1b.

In a few words, DOM is based on the integration of several layers called Mono-Dynamic Model (MDM), where each layer is related to a specific activity (such as population evolution, or flow of energy), into a multi-MDM model.

Let's take a little fictive illustrating study case: We want to make an Agent-Based Simulation (ABS) about wolves (and other animals). We will focus ourselves on wolves. A wolf agent will have some states, behaviors and interactors with environments. The first split we are going to make will through dynamic. We can identify two majors characteristics in this problem: the "individual dynamic" (that will include everything related to the individual such as its emotion, health, age, *etc.*) and the "team dynamic" (that will include everything related to the pack of wolves such as its rank, hunting, migrating, *etc.*).

This DOM methodology has already been applied in previous project such as DS [2] and EDMMAS [3]. The feedback we had on this application showed us that using DOM was a good choice. Indeed it enabled us to easily reuse an "old" simulation and build a new one from it.



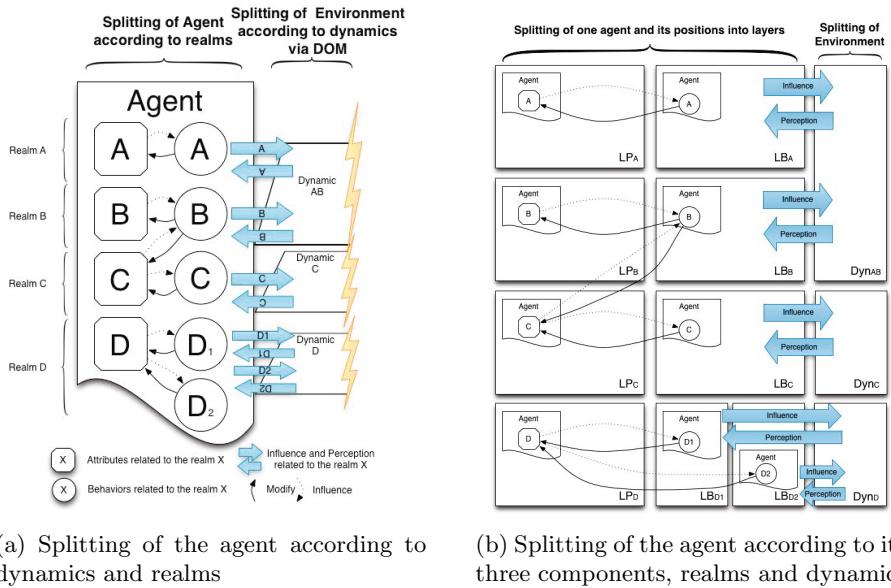
**Fig. 1.** From Basic Agent to the first splitting step

But the DOM methodology, based on the environment splitting, was not enough to fully apprehend modelization of complex system. We are facing some underlying problems that are general to every case of reusability. Whenever someone wants to reuse a MAS model, especially its agents, he will breast the problem of agents' behaviors. Despite the fact that dynamics had been separated, having the agents' states and its behaviors at the same level doesn't facilitate the reusability.

Indeed, DOM does not take into account the agents' behaviors, which is a critical point (as we can see in [4]) if we want to improve the reusability of our model. In this previous article, we presented a collaborative method and a NetLogo prototype focused on green turtles in the South-West Indian Ocean and we showed that each expert has different ways of modeling the turtles according to its interactions with the wind, the surface temperature, the stream, *etc.* Thus, because DOM concentrates its methodology on the environment and cannot tackle the problem of agents' behaviors, we also need to consider a methodology for the agents' modeling, especially its behaviors.

## 2.2 Splitting of the Agent According to Realms

In order to detach ourselves from ambiguous words, we will introduce the world "realm" which means area of behavioral expertise. Each set (states and behaviors) can be split according to the "realm" to which they are referring (*i.e.* in the Figure 2a the A realm, the B realm, the C realm where C is a bit related to B, and the D realm with two different behaviors).



**Fig. 2.** The second and third splitting steps

The "inside" of the agent in the Figure 2a shows us a certain amount of subsets (that could be really huge according to the complexity of the system) that are linked together. Our goal is precisely to organize everything to ease its conception.

The difference between realm and dynamic is that a dynamic can be composed by one or more realms. For example, the "social"'s realm and the "position"'s realm is part of the "communication"'s dynamic. Another example, the dynamic of "energy evolution" is composed by three realms : production of energy by plants, consumption of energy by residential houses and consumption by factories.

*In our previous study case*, we would be able to split the agent according to three realms: Emotional Wolf realm, Survival Wolf realm (both included in the "individual dynamic") and the Social Wolf realm (included in the "team dynamic"). We can notice, that realms of other animals can be included in the same dynamics (*e.g.* Survival Moose realm in the "individual dynamic").

## 2.3 Splitting of the Agent According to Its Three Components

An agent can be identified by three components:

- The agent's states, which contains its attributes.
- The agent's behaviors, which organizes all the actions it can undertake (decisional process).

- The agent's interactors, which allows interaction (influence and perception) with the environment.

In this new approach, we are taking an extra step in our initial DOM partition, in order to separate behaviors to free experts from behaviors unfamiliar to them. In this structure, we will define the whole "world" as an aggregation of several layers of physiognomy ( $LP$ ), several layers of behaviors ( $LB$ ) and interactors (*Influence* and *Perception*).

From this splitting by realms, we then can also split every agents and put them in differents physiognomies and behaviors layers.

If we take the same example, it will be as follows (in the Figure 2b):

- 4 Layers of physiognomies  $LP_A$ ,  $LP_B$ ,  $LP_C$  and  $LP_D$ .
- 5 Layers of behaviors  $LB_A$ ,  $LB_B$ ,  $LB_C$ ,  $LB_{D1}$  and  $LB_{D2}$ .
- 10 Interactors for each agent: 5 for *Influence* and 5 for *Perception*

This cutting of the agent into realms allows us to complete the one obtained by DOM 1 in term of dynamics. In this example, if we supposed that the splitting will result into three dynamics ( $Dyn_{AB}$ ,  $Dyn_C$  and  $Dyn_D$ ), their relation will be like in Figure 2b. In this figure, we can see the differents layers of the system, illustrated with the splitting of one agent. If we have hundreds of agents of the same kind, the same layers will be shared amongst them.

*Note:* In Figure 2b, we choose to simplify by showing the splitting of only one agent; but in fact, every agent will be split by the same realm and sent to the adequate layers. It could be represented as in Figure 3a. The advantage of this technique is that usually in a complex system, there are many agents that can be categorized by "kind". Each "kind" will be defined by the same set of behaviors, e.g. behaviors of an Omega wolf will be the same for every Omega wolf. By taking behaviors away from the agent's state, we are then able to factorize behaviors and reduce the complexity of the model.

**Layers of Behaviors.** One behavior's layer consists in definition of agents behaviors related to one realm.

A layer of behaviors is not supposed to contain the whole behavior of the agent, but its behavior related to one realm of the complex system. It could be defined by known method such as: logic definition, hard-coded definition, color-coded definition, formularized definition, Turing machine definition, tabular definition, matrix definition, etc.

These are only few examples, but it could use and combine a wide set of modelization methods, depending on the way the expert wants to model in his system.

**Layers of Physiognomies.** The physiognomy's layers is in fact a set of dynamic states related to one particular field (c.f. Figure 1a and Figure 2b). We used the word "**physiognomy**" in order to express the "character" or "personality" of the agents (its states and some internal laws related to this field) but

not what is usually called "body", because we did not incorporate the capacity of interaction here (which is usually associated to the body).

As you can see, an agent's states related to a particular realm (such as C) can influence and be modified by behaviors of the same agents, but related to an another realm (such as B). Moreover, a unique layer of physiognomy (*i.e.*  $LP_D$ ) can be linked to two or more layers of behaviors (such as  $LB_{D1}$  and  $LB_{D2}$ ).

*In our previous study case*, let's try this three components' split on the Survival Wolf realm. It would result in a set of physiognomies (such as its health, stamina, age, hunger, *etc.*), behaviors (attacking, calling for help, patrolling, *etc.*) and interactors (walking, running, biting, *etc.*). Wolves are usually not hunting when they are hungry, they usually call their pack in order to organize an attack. This realm is therefore related to the Social Wolf realm.

### 3 ...to the MVC Agent

#### 3.1 Introduction to Design Patterns

Since the introduction of patterns by Christopher Alexander in 1977-1979 in the architectural concept field, the idea of design patterns in software development started in 1987 [5] and gained popularity in 1994 after the book of [6].

Design patterns encourage reusability and can be used as "building blocks" for complex software. Several researches towards the reuse of model have been made in various fields as Software Engineering but also in Artificial Intelligence, *etc.*

Aridor and Lange's paper [7] was one of the first pioneer in applying design patterns to the MAS field. Ideas are emerging like PASSI (Process for Agent Societies Specification and Implementation) [8]. Since, several research has been done as resumed by [9], but most of the work has been focused on patterns for agent-oriented software or for the agent's interaction. Moreover, when the proposal is a pattern-based design methodology, the proposed patterns are usually homemade or specific to one domain. As stated by [10] to maximize the benefits of design patterns, they should be applied uniformly throughout the MAS research community, that would result in spreading MAS solutions and giving valuable feedback to the MAS research community.

As mentioned by [11] and [12], in order to create sets of system components needed to support highly interactive graphical software development, the MVC strategy has been chosen. Isolating components from each other as much as possible helps the application designer to understand and modify each particular unit, without having to know everything about the others.

#### 3.2 Model-View-Controller

If we get back to the basic concept of the MVC paradigm [11], we will see that the **view** manages the graphical and/or textual output to the portion of the bitmapped display that is allocated to its application. The **controller** interprets

**Table 1.** Example of MVC in a GUI component

Designation	Button in Swing
Model	ButtonModel
View	ButtonUI's visual representation
Controller	ButtonUI's handlers

the mouse and keyboard inputs from the user, commanding the model and/or the view to change as appropriate. Finally, the **model** manages the behavior and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller).

MVC implementation in smalltalk [11] inspired many other GUI frameworks. As an example, if we consider the **Button** class in Swing (Java [13]), the class which is used to represent a simple push button, we could see that Swing uses a variant MVC (where V and C are linked together). The **Button** class (see Table 1) is associated with a **ButtonModel** implementor for the **model**. It encapsulates the internal state of a button and defines its behaviors. **Button** class is also associated to a **ButtonUI** for its view, and possibly one or more event handlers for its controller.

Nearly all of the complex GUI elements in Swing use the component-level form of the MVC pattern for a number of excellent reasons, but the most important for us here is that it's highly reusable and it's easier to customize a component and link the components together.

### 3.3 Applying MVC to MAS: aMVC

If we apply this pattern to our MAS of a complex system , we will benefit most of the advantage of the MVC pattern. In order to head toward this, we need a new approach, starting from the bottom. The aim of our discussion is not only to propose a well-known design pattern, that could easily be both comprehended by experts and implemented by any developers. But we also want to underline the parallel between this design pattern and our modelization.

It's important to note that we are not talking about the software's architecture (as we can see in [14-16]) or a methodology in order to help in determining the types of agents needed to build successful MAS (such as in [17]), but about **the agent's architecture** itself. Applying MVC software's architecture would "simply" consist in using MVC in the context of software engineering, *e.g.* in a platform it will be separating visualisation of the world from the internal ABS's engine, usually through Object-Oriented Programming (OOP). Here, we want to import a methodology (MVC) existing in OOP into the world of MAS and use it as a design methodology for the agent. In a nutshell, we are going to use an **aMVC** (agent MVC) pattern.

**Table 2.** One layer of an aMVC agent

Designation	Agent in one realm X
Model	its states & internal laws of X
View	its interactors with the environment related to X
Controller	its behaviors in X

We have to ascertain identity of each concept in our current model: Who is the **model**? Who is the **view**? Who is the **controller**? In order to do that, we will talk for the next three sections of an agent related to one particular realm.

**Defining the Model.** In our modelization, the states' collection mixed with internal evolution laws related to the realm (such as aging of an agent) should be the **model**. We usually tag **model** merely as a database in Software Engineering; but the **model** in MVC is both the data and the domain logic needed to manipulate the data. Thus, identifying this to be the **model** is a good solution: *LP* is a subset of dynamical data (states of agents) which evolves with time due to *LBs* and internals laws of the realms (that make the consistency of the data).

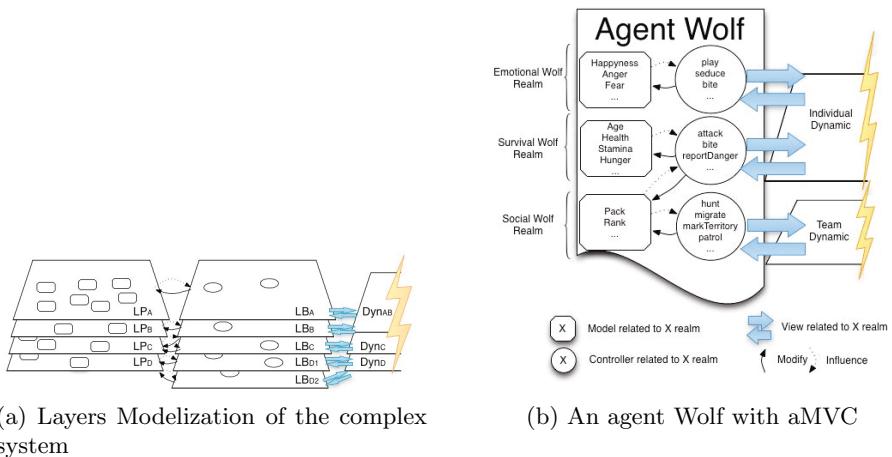
**Defining the Controller.** By adopting this point of view, we have identified the **controller**: the behaviors. The behaviors manage the agent's interaction with the environment and, for this, use its states: either in order to consult the states to take a decision or to influence its modification in order to memorize any experience learnings.

**Defining the View.** Now, last but not least, the **view** has to be defined. In other words, we can say that an agent's perception is similar to a button or a checkbox (in a graphical UI) which allows it to perceive external informations, and its influence is like a textbox or colored gauge through which the component can transmit informations (and do an influence) to the outside. The **view** of an agent is then the agent's interactors (which allow influence and perception) with the environment in a particular realm.

### 3.4 An Agent with aMVC

If we used the upper definition of aMVC, our agent will be a many-layered aMVC component, where each aMVC component is related to a realm. In the Table 2, we can see one layer of an agent in an aMVC form. This way, if we are taking one aMVC layer for each realm, we are then able to make a many-layered aMVC agent such as in Figure 2a.

When we compare the Figure 1a (of a basic agent) and the Figure 2a (of an aMVC agent by following the complement of methodology we proposed in this

**Fig. 3.** Layers Modelization and an example

paper), we could see that we found a way to organize the agent's components in a way that would ease the definition, the use and reuse of it.

*In our previous study case*, we will have an agent Wolf modelled according to aMVC such as in the Figure 3b.

## 4 Conclusion and Perspectives

This new modelization starts from the bottom (the agent) and not from the top (the system). We sliced the agent into a "mille-feuille" (where a layer is a realm) and then again according to its three components : physiognomy, behaviors and environment's interaction. This leads us to split also the environment into dynamic like we did before in DOM.

Due to the real splitting we would be able to give the layer to any experts, and if necessary divide the work among different experts thanks to the aMVC slicing by giving any part of the layer (M, V or C). This modelization help us in the creation of the agent. Additionally using the (a)MVC pattern's property, among other advantages, we would be able to make easy the reuse as well as the customization of any part of any layer of an agent.

This approach allows us to perceive a new field of investigation, particularly in a global level of layers' organization and the potential dynamic evolution of its interconnections, but also in aMVC: How far are the similarities between MVC and aMVC? Would we be able to apply variations of MVC to aMVC? etc. The study of this field will be the subject of further researches.

## References

1. Payet, D., Courdier, R., Sébastien, N., Ralambondrainy, T.: Environment as support for simplification, reuse and integration of processes in spatial MAS. In: IEEE International Conference on Information Reuse Integration, pp. 127–131 (2006)
2. David, D., Payet, D., Botta, A., Lajoie, G., Manglou, S., Courdier, R.: Un couplage de dynamiques comportementales: Le modèle DS pour l'aménagement du territoire. In: JFSMA 2007, pp. 129–138 (2007)
3. Gangat, Y., Courdier, R., Payet, D.: Démonstration: Aménagement énergétique d'un territoire - une approche par simulation multi-agents. In: Journées Francophones Systèmes MultiAgents, JFSMA 2009, pp. 237–240 (2009)
4. Gangat, Y., Dalleau, M., David, D., Sébastien, N., Payet, D.: Turtles are the turtles. In: European Simulation and Modelling Conference, ESM 2010, pp. 439–442 (2010)
5. Smith, R.: Panel on design methodology. SIGPLAN Not. 23, 91–95 (1987)
6. Gamma, E., Helm, R., Johnson, R.E., Vlissides, J.: Design Patterns. Addison Wesley (1995)
7. Aridor, Y., Lange, D.B.: Agent design patterns: elements of agent application design. In: Proceedings of the Second International Conference on Autonomous Agents, AGENTS 1998, pp. 108–115. ACM, New York (1998)
8. Cossentino, M., Burrafato, P., Lombardo, S., Sabatucci, L.: Introducing Pattern Reuse in the Design of Multi-agent Systems. In: Kowalczyk, R., Müller, J.P., Tianfield, H., Unland, R. (eds.) Agent Technology Workshops 2002. LNCS (LNAI), vol. 2592, pp. 107–120. Springer, Heidelberg (2003)
9. Klügl, F., Karlsson, L.: Towards Pattern-Oriented Design of Agent-Based Simulation Models. In: Braubach, L., van der Hoek, W., Petta, P., Pokahr, A. (eds.) MATES 2009. LNCS (LNAI), vol. 5774, pp. 41–53. Springer, Heidelberg (2009)
10. Cruz Torres, M.H., Van Beers, T., Holvoet, T.: (No) more design patterns for multi-agent systems. In: Proceedings of the Compilation of the Co-located Workshops on DSM 2011, TMC 2011, AGERE! 2011, AOOPES 2011, NEAT 2011, VMIL 2011, SPLASH 2011 Workshops, pp. 213–220. ACM, New York (2011)
11. Burbeck, S.: Applications Programming in Smalltalk-80: How to Use Model-View-Controller (MVC). Softsmarts, Inc. (1987)
12. Krasner, G.E., Pope, S.T.: A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System. Journal of Object Oriented Programming 1, 26–49 (1988)
13. Stelting, S., Maassen, O.: Applied Java Patterns. Prentice Hall PTR (2002)
14. Amblard, F., Ferrand, N., Hill, D.R.C.: How a Conceptual Framework Can Help to Design Models Following Decreasing Abstraction. In: SCS-European Simulation Symposium, Marseille, France, pp. 843–847 (2001)
15. Nutaro, J., Hammonds, P.: Combining the model/view/control design pattern with the DEVS formalism to achieve rigor and reusability in distributed simulation. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 19–28 (2004)
16. Nguyen, T.K., Marilleau, N., Ho, T.V.: PAMS – A New Collaborative Framework for Agent-Based Simulation of Complex Systems. In: Bui, T.D., Ho, T.V., Ha, Q.T. (eds.) PRIMA 2008. LNCS (LNAI), vol. 5357, pp. 287–294. Springer, Heidelberg (2008)
17. Mahmoud, Q.H., Maamar, Z.: Applying the MVC design pattern to multi-agent systems. In: Canadian Conference on Electrical and Computer Engineering, CCECE 2006, pp. 2420–2423 (2006)

# An Improved Method of Action Recognition Based on Sparse Spatio-temporal Features

Junwei Zhu, Jin Qi, and Xiangbin Kong

College of Electric Engineering, University of Electronic  
Science and Technology of China,  
611731, Chengdu, China

{jwzhu\_ee, jqichina, uestc\_vincent}@hotmail.com

**Abstract.** Sparse, informative feature representation has become an extremely successful method in action feature detection. Such features make the task more manageable while providing increased robustness to noise and pose variation. As the feature points detected are numerous, thus affect the computational efficiency. In this paper we present an improvement of this idea by decreasing the number of key points. And then we combine the use of 3D SIFT and pLSA in action categories. To test the validation of our method, we show the framework we devise in detail , also present some behavior recognition results on the KTH dataset including boxing, handclapping ,hand waving ,walking, jogging, and running.

**Keywords:** interestpoints, action recognition, spatio-temporal, sparse feature.

## 1 Introduction

Human activity recognition is an important area of computer vision research, recently it is well studied. Consider some of the well known difficulties faced in Human activity recognition. The human body can vary in posture, appearance, moreover, Occlusions and complex backgrounds can multiply the difficulties in recognizing human activity, as a result, the commonly used 2D interestpoints detectors are inadequate, Counterparts to 2D.the direct 3D make up for these disadvantages as it can provide stable 3D local feature representation which is a fundamental component of many video recognition. We propose an improvement of original detection algorithm, and apply semantic analysis model to action category, as a result, the computation and accuracy are taken account of at the same time.

The inspiration for our approach comes from approaches to object recognition that rely on sparsely detected features in a particular arrangement to characterize an object, and the successful usage of latent topic models. Such approaches tend to be robust to pose, image clutter, occlusion, object variation, and the imprecise nature of the feature detectors. In short they can provide a robust descriptor for objects without relying on too many assumptions. As an unsupervised model, pLSA[8] is applied to discover the latent topic that exit in each video document, for each video , a histogram of the number of occurrences of particular visual words is built which is implemented by the “bag of words” representation. Then the “semantic chasm” can be crossed using the pLSA model.



**Fig. 1.** Detect result comparisons of example frames in the KTH dataset. These are sampled from 30 frames in handwaving video. The interest points detected are colored to yellow, The group of pictures above is results using Dollar detector, and below using our proposed method.

**Table 1.** The average runtime and number of interest points detected in 30 frames

Action categories	Dollar detector	Our method
handwaving	5.97 s ; ( 5987 )	5.99 s ; ( 1738 )
handclapping	5.85 s ; ( 5229 )	5.91 s ; ( 1674 )
boxing	5.82 s ; ( 4358 )	5.85 s ; ( 1667 )
jogging	5.84 s ; ( 3387 )	5.87 s ; ( 1571 )

## 2 Related Work

Over the past decades, this problem of sparse interest points detection has received considerable attention. A widely used method is spatio-temporal local features method. In general, this method use local features extracted from 3-D space-time volumes to represent and recognize activities. Laptev [7] computed a saliency value for each voxel and detected the local saliency maxima based on Harris operator. While Dollar et. al. [3] applied the separate linear filters in the spatial and temporal directions and detected the interest points, which have local maxima value in both directions. Niebles et. al. [8] presented an unsupervised learning and classification method for human actions using the feature extractor similar to [3]. Liu [9] presented a methodology to prune cuboid features so as to choose important and meaningful features. Yilmaz and Shah [5] proposed an action recognition approach to extract sparse features called action sketches from a 3-D contour concatenation, which were confirmed to be view and ration-invariant.

After interest points are detected, for the purpose of describing these interestpoints more meaningfully, much work has been done using HOG[18], and SIFT[12] descriptor which has the advantages of scale, affine, viewpoint and rotation invariance, this descriptor is constructed by calculating the gradient around key points extracted in above step, Scovanner extent the 2D to 3-dimension(3D SIFT)[2], the descriptor is similar to the cuboid feature [3], however, the feature descriptors is dimension redundant, so the PCA-based representation for local features (PCA-SIFT)[4] is applied to decrease the descriptor dimension. The final step is clustering and recognizing activity, Ke et al. used segmented spatio-temporal volumes to model human activities, and recognize by searching for a subset of over-segmented spatio-temporal volumes that best matches the shape of the action model [11], recently, bag of words algorithm [6]also received much attention , Sivic et al. [10] perform unsupervised learning and recognition of object classes by applying a pLSA model with the bag of visual words representation.

### 3 Proposed Framework

The major steps of the training phase in our framework are described as follows: The videos are feed into the system, and a appropriate number of 3D interestpoints are extracted from each video. The 3D-SIFT algorithm is applied to descript the Interest Points, then the PCA is adopted to decrease redundant information, finally, pLSA model is trained to category the test action video.

#### 3.1 Feature Point Detection and Representation

In this paper, we adopt the spatio-temporal interest points detector proposed by Dollar[3]. Instead of using a 3D filter on the spatio-temporal domain, it applies two separate linear filters respectively to spatial and temporal dimensions. A response function can be represented as follows:

$$R = (I(x, y, t) * g_\sigma(x, y) * h_{ev}(t))^2 + (I(x, y, t) * g_\sigma(x, y) * h_{od}(t))^2 \quad (1)$$

wher  $g_\sigma(x, y)$  is the spatial Gaussian filter with kernel  $\sigma$  ,  $h_{ev}$  and  $h_{od}$  are a quadrature pair of 1D Gabor filters applied temporally. They are defined as:

$$h_{ev}(t; w, \tau) = -\cos(2\pi w)e^{-t^2/\pi^2} \quad \text{and} \quad h_{od}(t; w, \tau) = -\sin(2\pi w)e^{-t^2/\pi^2} \quad (2)$$

where  $w = 4 / \tau$  . They give a strong response to the temporal intensity changes. The interest points are detected at locations where the response is locally maximum. We run the above detection algorithm on KTH dataset to obtain interest points in handwaving video. Fig.1 shows the detected key points in handwaving videos, however, this detector pick up too much point. In this paper, we introduce an improvement of the detection method, we observed that in each frame, four dozen or less points can represent special action accurately, therefore, we can prune the

detected keypoints by setting a threshold, when the value computed by formulate (1) is below the threshold, the point will be discarded, to detect as many good points as possible, the threshold can't be set too high, then we sample the detected points periodically. We test the performance based on our proposed method, the lower part of Fig.1 show the detect result comparisons. Finally, we compare the performance using the two different methods which is list in Table 1.

To obtain a descriptor for each spatial-temporal interest point ,we choose 3D SIFT as our descriptor, the 3D SIFT descriptor use sub-histograms to encode local time and space information, and is able to robustly describe the 3D nature of the data in a way that vectorization of a 3D volume can't, this method is taken by sampling the surrounding the interest points (2x2x2 pixels regions ) to create the sub-histograms and computing the gradient magnitude and orientation for each pixel involved in each sub-regions, this yield descriptors of length 640 dimensions. When using larger feature vector, the recognition properties tend to be better, but for computation efficiency, we err on the side of using the smaller one, note that we must accumulate the gradient of each pixel in each 3D sub-region, and only orientation is obtained into a histogram. The final descriptor is a vectorization of the sub-histograms.

Given that the descriptors are dimension-redundant, so we adopt PCA to get the dimension of descriptors decreased, we observed that when the order of dimension get to 100, the dimension-lowed descriptors can represent originals effectively. The final descriptor of each interest point is a vector of 1x100.

### 3.2 Action Classification

In this section we will describe the detailed steps involved in classification, the descriptors are gathered from all the interest points and then a subset of them are quantized by clustering them into a pre-specified number of clusters, this step is carried out using k-means algorithm, the cluster centers are named 'words', all words are collected to construct a 'dictionary', the 3D SIFT descriptors from the videos are matched to each 'word' and the frequency of the words in each video is accumulated into a histogram. If there are N training videos and K words, then the collection of these histograms can be viewed as a KxN word frequency matrix M, of which the entry  $m(\omega_i, d_j)$  stores the number of occurrences of a spatial-temporal word  $\omega_i$  occurring in video  $d_j$ ,however, there is a latent topic  $z_k$  variable associated with each occurrence of a spatial-temporal word  $\omega_i$  in a video,. Each topic corresponds to an action category. The bringing in of topic  $z_k$  can be summarized by marginalizing  $P(\omega_i | d)$  over topic  $z_k$  to obtain the conditional probability:

$$P(\omega | d_{train}) = \sum_{k=1}^K P(z_k | d_{train}) P(\omega | z_k) \quad (3)$$

where  $K$  is total numbers of latent topics, the introduction of latent topics make the video document and words Conditional Independent from each other, thus can describe the intricate relationship between them. The objective of pLSA model is to find the mixing coefficients  $P(z_k | d_{test})$  such that the KL divergence between the measured empirical distribution  $P(\omega | d_{test})$  and:

$$P(\omega|d_{test}) = \sum_{k=1}^K P(z_k|d_{test})P(\omega|z_k) \quad (4)$$

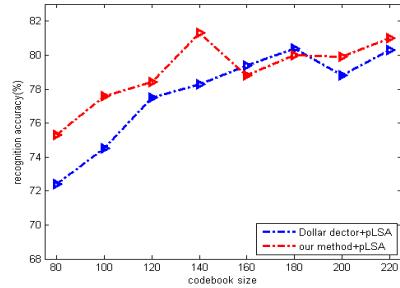
is minimized[20], we can use EM algorithm to find the solution. As the training and testing process are similar, we apply the Euclidean distance measurement between two vectors (specified as  $P(z|d_{train})$  and  $P(z|d_{test})$  respectively) produced by pLSA to be our measurement of similarity, The class of test video can be determined by the following formula:

$$\text{Location} = \arg \min_m D(P(z|d_{test}), P(z|d_{train,m})) \quad (5)$$

where  $P(z|d_{test}) = \{P(z_k|d_{test})\}_{k=1}^K$ , and  $P(z|d_{train,m})$  is the  $m$  th column in the topic-document matrix. The  $m$  th column in the topic-document matrix corresponds to one labeled training video document with known category. Hence the testing video's category can be asserted by judging which action category the value  $m$  calculated by formula (5) belong to.

boxing	.89	.00	.00	.11	.00	.00
handclapping	.14	.81	.05	.00	.00	.00
handwaving	.00	.09	.88	.00	.00	.03
jogging	.00	.00	.00	.67	.28	.05
running	.00	.00	.00	.12	.84	.04
walking	.02	.00	.00	.13	.06	.79

(a)



(b)

**Fig. 2.** (a) Confusion matrix for the KTH dataset using codeword size of 120, rows are ground truth, and columns are model results . (b) Classification accuracy vs. codebook size. The blue curve is the recognition results using Dollar detector +pLSA, and the red stand for results using our proposed framework, the remaining parameters are set equally.

## 4 Experiments

We explore results in the common used KTH dataset. To test the performance of our algorithm, we adopt the leave-one-out testing paradigm (LOO). We learn a model from the videos of 24 subjects (except those videos used to build the codebook in an unsupervised fashion, test the videos of the remaining subject, and compute a confusion table for evaluation. The results are reported as the average confusion table of the 25 runs, The confusion table are shown in Fig.2 (a). From Fig.2(a) we observed that the largest confusion located between “running” and “jogging”, “walking” and “jogging”, this accord to our intuition as these actions are similar and easily be categorized wrongly. Another comparing experiment results are shown in Fig.2 (b). It shows that our method with point pruning technique gets better than Dollar’ method.

## 5 Conclusion

In this paper, we have presented an improvement of the visible action classification algorithm. Keeping in mind that the original Dollar detector extract too much interestpoints, to improve computational efficiency, we only pick up some typical points as our final feature points, we choose 3D SIFT other than HOG algorithm to be our descriptor as it is insensitive to light and invariant to rotation, In order to obtain further insight into the complex relationship between words and videos, we introduce pLSA model into our framework, we also exploited the relationship between the number of interestpoints and recognition accuracy, the experiments have shown the validation and improved performance of our method on the task of action recognition.

## References

1. Aggarwal, J.K., Ryoo, M.S.: Human activity Analysis: A Review. *ACM Computing Surveys* 43(3), Article 16 (2011)
2. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the ACM International Conference on Multimedia, pp. 357–360. ACM, NewYork (2007)
3. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: Proceedings of the IEEE International Conference on Computer Vision. Workshop: VS-PETS (2005)
4. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 506–513 (2004)
5. Yilmaz, A., Shah, M.: Actions sketch: A novel action representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 984–989. IEEE, Los Alamitos (2005)
6. Csurka, G., Dance, C.R., Fan, L.: Visual categorization with bags of key points. In: Proceedings of European Conference on Computer Vision (2004)
7. Laptev, I., Lindeberg, T.: Space-time interest points. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE, Los Alamitos (2003)
8. Niebles, J.C., Wang, H., Fei-fei, L.: Unsupervised learning of human action categories using spatial-temporal words. In: Proceedings of the British Machine Vision Conference (2006)
9. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos “in the wild”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Los Alamitos (2009)
10. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: International Conference on Computer Vision, pp. 370–377 (2005)
11. Ke, Y., Sukthankar, R., Hebert, M.: Spatio-temporal shape and flow correlation for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Los Alamitos (2007)
12. Rapantzikos, K., Avrithis, Y., Kollias, S.: Dense saliency-based spatiotemporal feature points for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Los Alamitos (2009)

# Echelon Platoon Organisation: A Distributed Approach Based on 2-Spring Virtual Links

Madeleine El Zaher, Jean-Michel Contet,  
Franck Gechter, and Abderrafiaa Koukam

IRTES-SeT, UTBM,  
90000 Belfort, France

{madeleine.el-zaher,jean-michel.contet,  
franck.gechter,abder.koukam}@utbm.fr

**Abstract.** This paper presents a reactive multi-agent system for echelon platoon organization. Platoon systems are sets of vehicles that move together while keeping a predefined geometrical configuration without any material coupling. Each vehicle represents an autonomous agent that behaves based only on its own perceptions. The distributed platoon organization problem consists in defining the algorithms to be executed by each vehicle's embedded software, in order to maintain the desired platoon configuration during displacements. Platoon systems found in literature deal generally with column formations adapted to urban or highway transportation systems. Other formations such as line, echelon,... can be encountered in fields like agriculture and the military. In this paper, we focus on the platoon echelon formation.

An approach based on a virtual vehicle-to-vehicle interaction model composed of two springs is proposed. Those virtual springs attach a platoon's vehicle to its local leader, another platoon's vehicle. In this work, five different spring's attachment points are evaluated, to compare them and conclude about the more suitable one, depending on platoon's trajectory. Eventually, from the evaluation results, it can be conceived to make attachment points evolve during platoon operation.

**Keywords:** Platoon system, multi-configurations, multi-agent system.

## 1 Introduction

A platoon system is a set of vehicles that move together while keeping a predefined geometrical configuration without any material coupling. Different platoon configurations can be adopted in different fields. The best known formation is the column, where vehicles are placed one behind the other. This configuration is studied in projects like PATH [8], CRISTAL<sup>1</sup>, [4]... and is well adapted to urban and highway transportation. Other configurations like line, echelon, ... admit other applications such as the military and agriculture. In echelon formations, vehicles form one side of a "V". This configuration seems to be adapted

---

<sup>1</sup> <http://projet-cristal.net/>

to agricultural activities such as harvesting. For any configuration, decentralized platoon organization is based on algorithms to be executed by each vehicle's embedded system, in order to maintain the desired configuration during platoon's displacements along the desired trajectory.

In [7], an approach dedicated to column platoon formation, based on a reactive multi-agent systems is proposed. In this approach each vehicle is considered as an agent that behaves based only on the perceptions of another platoon's vehicle, its nearest predecessor in the direction of movement. To compute its references (a speed vector), each agent uses a virtual physics-inspired interaction model composed of two springs that link a vehicle to its predecessor. This paper suggests to use the same 2-springs interaction model in order to deal with the echelon platoon control. To guarantee an echelon formation the attachment points of both springs can be different from the points considered in column formation. It seems interesting to evaluate five different cases of virtual link attachment.

The main expected advantage of the use of the 2-spring model in the echelon platoon formation, is that this facilitates the transition between configurations (evolving from a platoon configuration to another) during platoon operations. This aspect is to be studied in future works. In fact, if the 2-springs approach is adapted to echelon platoon formations, the transition from echelon formation to column can be made by simply changing the point of attachment of the two springs.

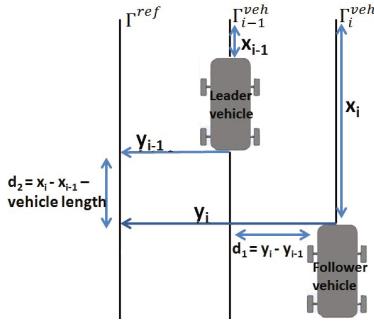
This paper is structured as follows, in section 2 a state of the art about the different platoon configuration and the approach to control these configurations is addressed. Section 3 mentions the coherence between the proposed approach of platoon and the reactive multi-agent system. In section 4, we describe the physical model used in column platoon formation and we consider five different possible attachments of the springs, in order to organize an echelon formation. In Section 5, the five proposed attachment geometries are tested and simulated, then a comparison of these results is presented.

## 2 State of the Art

A platoon is a set of vehicles (generally autonomous) that displace within an environment (rural, urban,...) while maintaining a predefined geometric configuration. This article concentrates on the echelon formation, where vehicles are placed as one side of a "V". The first platoon's vehicle in the direction of displacement, called the leader, can be autonomous or human-driven, other vehicles are autonomous and, depending on the approach, move according to a global leader or to a local one.

Figure 1 shows two vehicles in an echelon formation. Echelon configuration is characterised by lateral distances  $d_1 > 0$  and longitudinal distance  $d_2 > 0$ . The problem of platoon organisation can be compared to the mobile robots formation control problem, where three main strategies have been proposed :

- Behavior based ([1]) : in this strategy for each robot is associated a behavior (static obstacle avoidance, avoidance of collision with other robots...) with



**Fig. 1.** Platoon in an echelon formation

an importance degree. Each behavior produces a different response (orientation, speed ...). The global behavior is deduced by adding the behaviors weighted by their degree, and normalizing the result. The main disadvantage of this strategy, is the difficulty to prove the stability of the system, and the maintenance of the formation.

- Virtual structure ([5], [2]) : in this strategy robots move together as a single rigid body. Each robot tries to maintain a fixed position in this virtual body. The virtual body evaluates as only one entity with its orientation and speed. The main disadvantage of such strategy is that it requires a lot of communication between the different components of the system.
- Leader-follower ([3], [12]) : in this strategy, an entity (robot) is considered as a leader and follows a predefined trajectory. Other robots, follow this leader and try to maintain the desired distance and orientation. The main disadvantage of this strategy, is that it depends heavily on the leader vehicle.

Most of the works done on platoon organisation deal with the column configurations. Two main approaches, global and local, were proposed to organise the column configuration. Global approaches ([1], [13]) are centralized, i.e. there exist an entity that determines required reference information. Global approaches show precise trajectory matching, however they required sophisticated technologies: GPS RTK, reliable wireless communication, ...

Local approaches ([10], [9]) are decentralized, each vehicle computes its own references based on its perceptions. They require less technology, generally distance measurement devices are sufficient. However, since only a local view is accessible, more sophisticated trajectory matching has to be implemented.

The goal of this paper is to propose a local approach for platoon organisation in echelon configuration, where each vehicle perceives a local leader (the nearest vehicle in movement's direction). This strategy allows to avoid the constraint of a unique leader vehicle, and the communication requirement in global approaches. The follower vehicle perceives its local leader and tries to follow it while maintaining the desired lateral and longitudinal nominal distances, in order to conserve the desired echelon formation.

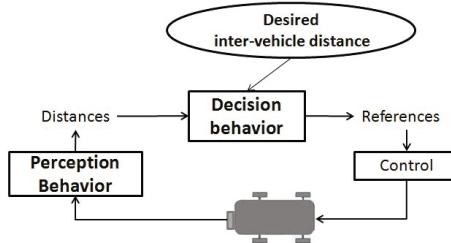
### 3 Multi-agent System for Vehicle Platoons

In local approaches to platoon systems, each vehicle computes its own references autonomously based only on its perceptions. Consequently an adequate embodiment for the local platoon is a reactive multi-agent system, where each vehicle represents an autonomous agent. The platoon system is then the result of a self-organization of a reactive multi-agent system (RMAS). The global behavior of the platoon systems emerges as a result of the individual behavior of each vehicle agent. Two behaviors can be distinguished in a platoon system : the leader behavior, concerns the vehicle (agent) in the first position of the platoon. This agent is either autonomous or human driven, it interacts directly with the road. The follower behavior: concerns the follower agents. These agents perceive their predecessors and act based on their perceptions.

The behavior of each follower agent can be described as a combination of the following sub-behaviors (cf.figure 2) :

- *Perception* : the follower agent, perceives its predecessor and measures the inter-vehicle distances required in the control behavior, using sensors like laser range finder.
- *Decision* : computes the references of the vehicle (acceleration and orientation) using an interaction model.
- *Control* : Transforms the computed references into commands.

The control behavior consists in dealing with three main issues : formation maintaining, collision avoidance and speed regulation.

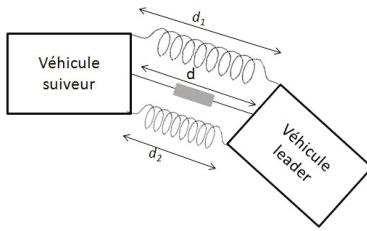


**Fig. 2.** Sub-behaviors of each follower vehicle

### 4 Interaction Model

The virtual physics-inspired interaction model relating a vehicle to its local leader is composed of two springs and a damper, as shown by figure 3 in the case of attachments used for the column configurations. In section 4.2 different possibilities of attachment points in echelon formation are presented.

Parameters involved in this physical interaction model are :



**Fig. 3.** Physical model installed in an echelon platoon formation

- $m$ , the mass of the vehicle.
- *width* and *length*, respectively the width and the length of the vehicle.
- $l_1$  and  $l_2$ , both springs resting length.
- $k_1$  and  $k_2$ , the stiffness of each one of the two springs. We consider also the existence of a stiffness  $K$  equal to the sum of  $k_1$  and  $k_2$ .
- $h$ , the damping coefficient.
- $v$  and  $\gamma$ , the speed and the acceleration of the vehicle.
- $\mu$ , the friction coefficient.

Laser range finders are used to measure three distances:  $d_1$ ,  $d_2$  the length of each one of the springs, and  $d$  the length of the damper. Four forces intervenes in this model :

- Force of first spring:  $\vec{f}_{s1} = k_1 \vec{X}_1$ , with  $\vec{X}_1$  the elongation of the first spring,  $\|\vec{X}_1\| = (d_1 - l_1)$ .
- Force of the second spring  $\vec{f}_{s2} = k_2 \vec{X}_2$ , with  $\vec{X}_2$  the elongation of the second spring and  $\|\vec{X}_2\| = (d_2 - l_2)$ .
- Damping force :  $\vec{f}_d = h \vec{v}$ .
- Friction force :  $\vec{f}_f = -\mu m \vec{g}$ ,  $g$  represents the gravity of earth.

Each follower vehicle computes its references (speed and orientation) according to Newton's second law of motion<sup>2</sup>:

$$\vec{F} = m \vec{\gamma} = \vec{f}_{s1} + \vec{f}_{s2} + \vec{f}_d + \vec{f}_f \quad (1)$$

#### 4.1 Model Parameters

The value of parameters  $K$ ,  $k_1$ ,  $k_2$  and  $h$  has to be determined, as described briefly in this paragraph, for the case of a column formation.  $K$  is constant during the platoon operations. In case of linear trajectories,  $\|\vec{X}_1\| = \|\vec{X}_2\|$  and  $k_1 = k_2 = \frac{K}{2}$ . To facilitate the computation, we replace the two springs by a single one having  $K$  as stiffness and  $\|\vec{X}\|$  as elongation. The speed of the vehicle

---

<sup>2</sup> The acceleration  $\vec{\gamma}$  of a body is parallel and directly proportional to the sum of forces  $\vec{F}$  and inversely proportional to the mass  $m$ , i.e.,  $\vec{F} = m \vec{\gamma}$ .

is written  $\ddot{\vec{X}}$ , and the acceleration will be written as  $\ddot{\vec{X}}$ . So equation 1 could be written like :

$$m \ddot{\vec{X}} = -K \vec{X} - h \dot{\vec{X}} + \mu m \vec{g} \quad (2)$$

Before the vehicle startup we have :  $\vec{0} = -K \vec{X} - h \dot{\vec{X}} + \mu m \vec{g}$ . The follower vehicle must not startup while the elongation of the spring is smaller than the desired elongation. Consequently, we can deduce that :

$$K = \frac{\mu m g}{X_0} \quad (3)$$

where  $X_0$  is the desired spring elongation.

$k_1$  and  $k_2$  vary during the platoon operations proportionally to the variation of the two distances  $d_1$  and  $d_2$ .

To compute the damping coefficient  $h$  a kinematic study of the "spring-damper" system should be done. Equation 2 could be written like :

$$\ddot{\vec{X}} + 2\varepsilon \omega_0 \dot{\vec{X}} + \omega_0^2 = \frac{\mu}{m} m \vec{g} \quad (4)$$

where,  $2\varepsilon \omega_0 = \frac{h}{m}$  and  $\omega_0 = \sqrt{\frac{h}{m}}$ .

The damping coefficient  $h$  could be deduced by solving the equation 4 to be in the critical damping case, so the discriminant is considered to be null.  $\Delta = 0 \Rightarrow \varepsilon = 1$ .

$$\varepsilon = \frac{h}{2 * \sqrt{K m}}$$

Then,

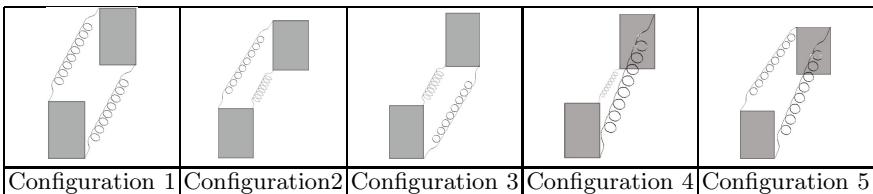
$$h = 2 * \sqrt{K m} \quad (5)$$

A detailed study of this model is presented in [7], were some simulation results that marks the efficiency of this model in column formation of platoon are presented. In [6], a formal verification using the SAL model checker prove that the safety property : *No collision between two follower vehicles* is valid during the platoon circulation.

## 4.2 Platoon Echelon Formation

To organise the echelon configuration, the same physics-inspired model (2 springs and a damper) is used, only the attachment points will change. Main advantages behind the use of this control model are : First, the use of two springs with two different and variable stiffness helps maintaining the formation by adjusting the values of both stiffness when needed. Second, the 2-springs control model is already used in column platoon formation, and it can also be used in other formation, which facilitates the transition between configurations. In fact, this transition could be simply made by changing the attachment points of the springs.

Table 1 shows five different possibilities that can be adapted in echelon configurations. As in column configuration, laser range finders are used to measure

**Table 1.** Five different configurations of the virtual interaction model

the length of the two springs and the damper. Then the physical interaction model is used to compute the acceleration of the system. To compute the parameters of the physical model ( $K$ ,  $k_1$ ,  $k_2$ ,  $h$ ) we can produce like described in [4.1]. Where the two springs are replaced by one having  $K$  as stiffness,  $\vec{X} = \frac{\vec{X}_1 + \vec{X}_2}{2}$  as elongation, and  $l = \frac{l_1 + l_2}{2}$  as spring resting length.

## 5 Simulations and Comparison of Different Configurations

In order to compare the five configurations of echelon platoon, simulations were done using Vivus<sup>3</sup> simulator, a simulator developed in the SeT laboratory<sup>4</sup>. Vivus incorporates 3D Geo-localized models. Vivus also integrates a detailed vehicle's physical model, aware of aspects such as tire-road contact. Simulations involve a 2-vehicle platoon. The leader vehicle is manually controlled, the follower vehicle is equipped with the platoon algorithms described in section 4. Vehicle perception is made by a simulated laser range finder having the same characteristics (range, angle, error, ...) than the real vehicle's sensor. In the five configurations, longitudinal and lateral nominal distances are 3 meters.

Two main evaluations where made :

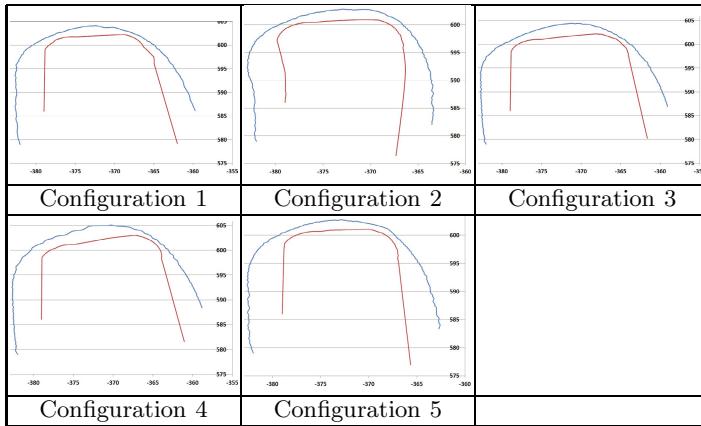
- Evaluation of lateral error : measure of the horizontal spacing between two following vehicles and compare it to the desired lateral distance.
- Evaluation of longitudinal error : measure of the vertical spacing between two following vehicles and compare it to the desired longitudinal distances.

### 5.1 Evaluation of Lateral Error

To evaluate the lateral error, following test is suggested: the leader vehicle will turn around a building in the simulation area. During its trajectory, the vehicle will go through curves with rotation angle equal to 90 degrees. To visualize the results of this test, the trajectories followed by each one of the leader and the follower vehicles are drawn. Table 2 shows these trajectories. We also present in table 3 the variation of the lateral distance during the time.

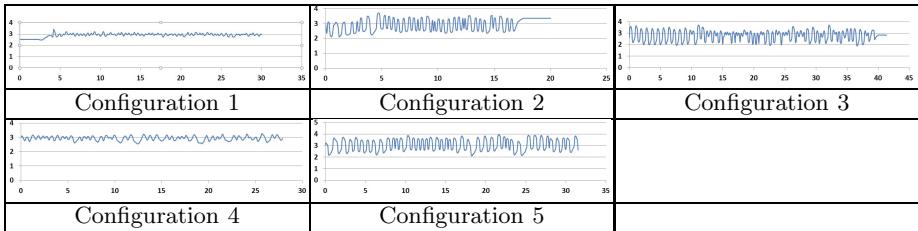
<sup>3</sup> [http://www.multiagent.fr/Vivus\\_Platform](http://www.multiagent.fr/Vivus_Platform)

<sup>4</sup> <http://set.utbm.fr/>

**Table 2.** Trajectory error during a turning case

Results are presented in tables 2 and 3 shows that in the five configurations the lateral distance is always close to the desired lateral distance (3 m).

The maximal errors are produced at the point of inflection. In configurations 1 and 4, the maximal lateral error is 30 cm. The average of this error is 10 cm (less than the width of a tire). In configurations 2,3 and 5, the maximal error reaches 50 cm.

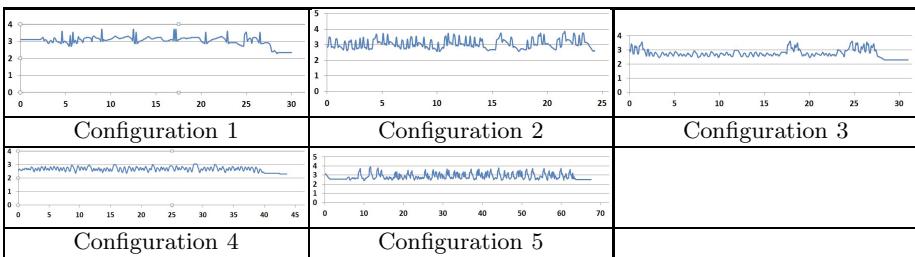
**Table 3.** Results of the lateral deviation. Horizontal axe represents the time in seconde, vertical axe represents the distance in meter.

**Conclusion of the Test:** Results produced by this tests give confidence to the configuration 1 and 4

## 5.2 Evaluation of Longitudinal Distance

In this test, for each one of the five configurations we measure the longitudinal distance between the leader and the follower vehicle during an arbitrary trajectory, with a vehicle speed of 7 km/h. The measured longitudinal distance is

**Table 4.** Results of the longitudinal deviation. Horizontal axis represents the time in second, vertical axis represents the distance in meter.



compared to the desired one (3 m). Table 4 dresses the results of the tests made on the five configurations.

The graphs presented in table 4 show that in the five configurations the longitudinal distance is always close to the nominal lateral distance (3 m).

In configurations 1, 3 and 4 the maximal longitudinal deviation is 50 cm. While in configurations 2 and 5 this maximal deviation reaches the 80 cm. Adding to this configurations 1, 3 and 4 show more stable graphs, while in configurations 2 and 5, oscillations of longitudinal distance are more greater.

**Conclusion of the Test:** The longitudinal test shows that the configurations 1, 3 and 4 produce more advantageous results and also close to the desired ones.

### 5.3 Conclusion of the Results

The results presented by these tests, show that among the five configurations proposed in table 4 the first one (configuration 1) is the most suitable to address the echelon problem.

Configurations 3 and 4 show also some promising results. However, in the lateral tests configuration 1 seems more reliable than these two configurations . Consequently, we can say that the configuration 1 is the most suitable configuration to be used in the echelon platoon formation as a control model.

## 6 Conclusion

The goal of this paper was to find an efficient method to organize an echelon platoon formation. The proposed method bases on reactive multi-agent system where each vehicle is considered as an agent that behaves based on its perceptions. To compute its speed and direction references each agent uses a virtual physical interaction model composed of two springs that attaches a follower vehicle to its predecessor. This control approach is already used to control platoon in column formation. To adapt it to the echelon formation, we evaluate five interaction model that differ by their attachment points. Simulations were done to evaluate the quality of each one of the five proposed configurations. After a comparison between the results produced by the five configurations, we concluded by identifying the most suitable configuration for the echelon formation.

Future works will be devoted to evaluate the 2-springs interaction model from the point of view of dynamic transition between formations (passing, for instance, from linear to echelon formation).

## References

1. Avanzini, P., Thuilot, B., Martinet, P.: A control strategy taking advantage of inter-vehicle communication for platooning navigation in urban environment. In: IROS 2011 International Workshop on Perception and Navigation for Autonomous Vehicles in Human Environment (2011)
2. Beard, R.W., Lawton, J., Hadaegh, F.Y.: A feedback architecture for formation control. In: American Control Conference, vol. 6, pp. 4087–4091 (2000)
3. Das, A.K., Fierro, R., Kumar, V., Ostrowski, J.P., Spletzer, J., Taylor, C.J.: A vision-based formation control framework. *IEEE Transactions on Robotics and Automation* 18(5), 813–825 (2002)
4. Daviet, P., Parent, M.: Longitudinal and lateral servoing of vehicles in a platoon. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 41–46 (1996) Automatic driving;Platooning techniques
5. Do, K.D., Pan, J.: Nonlinear formation control of unicycle-type mobile robots. *Robotics and Autonomous Systems* 55(3), 191–204 (2007)
6. El-Zaher, M., Contet, J.M., Gruer, P., Gechter, F.: Towards a compositional verification approach for multi-agent systems: Application to platoon system. In: First International workshop on Verification and Validation of Multi-agent Models for Complex Systems, V2CS (2011)
7. El-Zaher, M., Gechter, F., Gruer, P., Hajjar, M.: A new linear platoon model based on reactive multi-agent systems. In: The 23rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI. IEEE Computer Society (2011)
8. Hedrick, J.K., Tomizuka, M., Varaiya, P.: Control issues in automated highway systems. *IEEE Control Systems* 14(6), 21–32 (1994)
9. Jean-Michel, C., Franck, G., Pablo, G., Abder, K.: Physics inspired multiagent system for vehicle platooning. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2007, pp. 184:1–184:3. ACM (2007)
10. Moskwa, J.J., Karl Hedrick, J.: Nonlinear algorithms for automotive engine control. *IEEE Control Systems Magazine* 10(3), 88–93 (1990)
11. Scharf, D.P., Hadaegh, F.Y., Ploen, S.R.: A survey of spacecraft formation flying guidance and control. Part ii: control. In: American Control Conference, July 30–July 2, vol. 4, pp. 2976–2985 (2004)
12. Tanner, H.G., Pappas, G.J., Kumar, V.: Leader-to-formation stability. *IEEE Transactions on Robotics and Automation* 20(3), 443–455 (2004)
13. Woo, M.J., Choi, J.W.: A relative navigation system for vehicle platooning. In: Proceedings of the 40th SICE Annual Conference, SICE 2001 International Session Papers (IEEE Cat. No.01TH8603), pp. 28–31 (2001)

# Multi-agent Task Division Learning in Hide-and-Seek Games

Mohamed K. Gunady<sup>1</sup>, Walid Gomaa<sup>1,\*</sup>, and Ikuo Takeuchi<sup>1,2</sup>

<sup>1</sup> Egypt-Japan University of Science and Technology, New Borg El-Arab, Alexandria, Egypt

{mohamed.geunady, walid.gomaa}@ejust.edu.eg

<sup>2</sup> Faculty of Science and Engineering, Waseda University, Tokyo, Japan

nue@nue.org

**Abstract.** This paper discusses the problem of territory division in Hide-and-Seek games. To obtain an efficient seeking performance for multiple seekers, the seekers should agree on searching their own territories and learn to visit good hiding places first so that the expected time to find the hider is minimized. We propose a learning model using Reinforcement Learning in a hierarchical learning structure. Elemental tasks of planning the path to each hiding place are learnt in the lower layer, and then the composite task of finding the optimal sequence is learnt in the higher layer. The proposed approach is examined on a set of different maps and resulted in convergece to the optimal solution.

**Keywords:** Hide-and-Seek, task division, multi-agent systems, Q-learning, task sequence learning.

## 1 Introduction

Most of the real life applications of multi-agent systems and machine learning consist of highly complicated tasks. One learning paradigm for solving such complex tasks is to learn simpler tasks first then increase the task complexity till reaching the desired complexity level, similar to the learning paradigm followed by human children. Various practical planning problems can be reduced to Hide-and-Seek games in which children learn the basic concepts of path planning, map building, navigation, and cooperation while working on teams. Robotics applications in various domains such as disaster rescuing, criminal pursuit, and mine detection/demining are typical examples. One important aspect when dealing with cooperative seekers is how to divide the searching environment among them for a better seeking performance, namely the territory division, or in more abstract term, task division.

Previous researches have been done on similar fundamental problems like the problem of multi-agent coverage. A survey of the coverage problem was done by Choset in [1]. Choset discussed some heuristic, approximate, and exact coverage algorithms based on cellular decomposition. Some complete coverage algorithms for multi-robot were also discussed. Another idea is using spanning trees as a base for

---

\* Currently on leave from Faculty of Engineering, Alexandria University, Egypt.

efficient coverage paths in multi-robot coverage. Agmon et al. exploited this idea in [2] by introducing a polynomial-time tree construction algorithm in order to improve the coverage time. However, they assume that the robots are initially distributed over the terrain and they build the spanning tree so that the robots do not pass by each other throughout the coverage plan. The problem of multi-agent coverage is a little different from the territory division problem in Hide-and-Seek games. In seeking tasks it is commonly the case for the seekers to start at adjacent points and their seeking trajectories might intersect. Furthermore, it is not necessary to cover all the terrain in Hide-and-Seek; not only scanning the possible hiding places is sufficient, but scanning the good hiding places first is also preferable. These are the main differences by which our approach is motivated to overcome using Reinforcement Learning. In this paper, we propose a new learning model to solve the problem of territory division using a hierarchical learning structure.

The following Section 2 gives more details about the problem of seeking task division in Hide-and-Seek games. Section 3 provides a quick review of the Q-Learning algorithm adopted in this work. Section 4 illustrates the proposed learning model for both single and multiple seekers. Experimental results and analysis of the proposed approach is presented in Section 5. Finally, Section 6 concludes the paper.

## 2 Task Division in Hide-and-Seek

Consider a simple form of the Hide-and-Seek games in which there are multiple seekers and a hider. Before the game starts, the hider chooses a hiding place based on a pre-determined hiding probability distribution over the environment based on an early learning process. Afterwards, it is the task of seekers to search for the hider and find his hiding place in minimum time. A game terminates as soon as one seeker finds the hider, in other words neither escaping nor chasing is involved. Seekers cannot just scan the whole environment for the hider, or they will suffer from a poor seeking performance. They should learn the possible hiding places and perform an optimized scan that minimizes the average time for finding the hider. For a game with multiple seekers, a good seeking strategy is to cooperate to employ the concept of dividing seekers' territories to increase the seeking efficiency and thus decrease the seeking time. Nevertheless such cooperation requires physical communication channel between the seekers. In this paper it is assumed that a full communication is provided in the learning phase, while after learning seekers can work without communication.

To achieve a good territory division strategy there are three possible approaches: first and simplest is to adopt some fixed heuristics, e.g. East-West, North-South, etc. or any similar partitioning strategy. However, this approach is too naive to achieve well load-balanced performance with different map topologies. The second approach is to solve the territory division problem algorithmically. To construct an algorithm that finds the optimal seeking strategy is a hard task that requires a rigorous understanding for the problem model and its details. The  $k$ -Chinese postmen problem is a close example, which is classified as an NP-complete problem.

The third approach is learning; the learning approach can be considered a moderate solution. By learning it is easy for the system to be adapted to different map topologies and hiding probability distributions. In the next sections, a learning model

will be illustrated for solving the multi-seekers territory division problem by applying a simple Reinforcement Learning algorithm, Q-Learning.

### 3 Q-Learning

Reinforcement Learning (RL) is a machine learning paradigm in which an agent learns over time through repetitive trial-and-error interaction with the surrounding environment. Rewards are received depending on the agent's actions attached to the observed states of the environment. RL aims to learn an optimal decision-making strategy that gains the maximum total (discounted) reward received from the environment over the agent's lifetime. Most RL algorithms are based on the Markov Decision Process (MDP) model in which the agent needs to make a decision to choose an action  $a$  depending only on the current state  $s$  and resulting in a new state  $s'$  while a cumulative reward function  $R$  is used to quantify the quality of the temporal sequence of actions. One of these algorithms is the Q-Learning of Watkins [3].

The agent's task is to learn an optimal policy  $\pi^*: S \rightarrow A$ , that tells the agent which action  $a \in A$  to take given the current state  $s \in S$  with the goal of maximizing the cumulative discounted reward. Given a policy  $\pi$  and a state  $s_t$  at time step  $t$ , the value of  $s_t$  under  $\pi$  (assuming infinite horizon) can be calculated as follows:

$$V^\pi(s_t) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \quad (1)$$

where  $0 < \gamma < 1$  is the discount factor and  $r_{k+1}$  is the immediate reward at time  $k$ . Thus by calculating  $V^*(s)$ , the optimal value function, an optimal action policy  $\pi^*$  is readily available. Unfortunately, in many practical problems the reward function (and generally the environment model) is unknown in advance which means equation (1) cannot be evaluated. Let  $Q(s, a)$  be the maximum discounted cumulative reward achieved by taking action  $a$  from state  $s$  and then acting optimally. In other words, the  $Q$  value equals the immediate reward given by taking action  $a$  from state  $s$ , plus the value gained by following an optimal policy afterwards as follows:

$$Q(s, a) \equiv r + \gamma V^*(s') \quad (2)$$

Notice that  $V^*(s) = \max_a Q(s, a)$  which allows us to rewrite Equation (1) in a recursive form as follows:

$$Q(s, a) = r(s, a) + \gamma \max_{a'} Q(s', a') \quad (3)$$

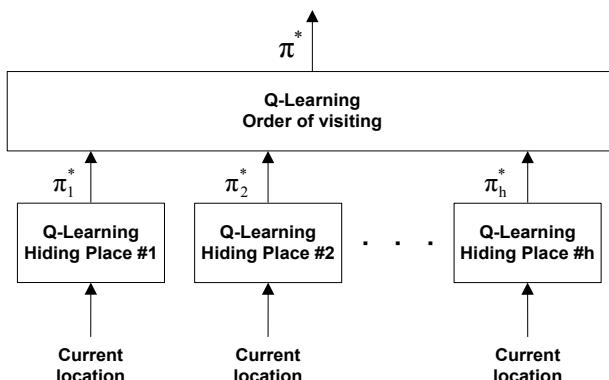
where  $s'$  is the next state reached from state  $s$  after taking action  $a$  (this formula assumes deterministic environment dynamics). The Q-Learning algorithm constructs a lookup table for its estimated  $Q$  values and observes the experience tuple  $\langle s, a, r, s' \rangle$  for each iteration. After visiting every state-action pair infinitely often, the constructed  $Q$  table values are guaranteed to asymptotically converge to the true  $Q$  values [4].

## 4 Proposed Learning Model

This section discusses our proposed model to solve the problem of territory division learning. The first subsection explains the overall idea of the model and its structure. Then followed by two subsections: one for a single-agent case and second for multiple agent case with a generalized model.

### 4.1 Sequential Task Learning

In [5], Singh introduces an elegant module for an efficient learning for the task of a sequence of multiple subtasks for single agent. It considers a learning agent deals with an environment of finite-state and discrete-time space in order to perform a sequence of tasks. Each task requires the agent to take a sequence of actions to bring it to the next desired goal. The territory division problem in a Hide-and-Seek game can be modeled similarly. If a seeker is assigned a part of the environment as his territory, then the seeker has to perform a sequence of elemental tasks each of which corresponds to visiting one of the possible hiding places and the total tasks sequence is completed when the seeker visit all the possible hiding places in his assigned territory. RL algorithms such as Q-learning or Sutton's TD( $\lambda$ ) [6] can be used to solve each elemental task independently, or by applying some graph-theoretic algorithms such as Dijkstra's shortest path algorithm [7] if applicable. Consequently, if we assume that the seeker knows the best strategy to visit each hiding place in the environment given its current position, then the territory division problem can be reduced to be the problem of learning the best order of visiting all the hiding places in the minimum total time. The hierarchical learning system can be formulated as shown in Figure 1. The basic block Q-Learning Hiding place  $#i$  is responsible for learning the best strategy to reach the hiding place number  $i$  given any current location. If the elemental tasks can to be solved by shortest path techniques, like the case of feasible small problem space, these learning blocks can be replaced with other shortest path computation blocks.



**Fig. 1.** Hierarchical structure of a sequential task learning for  $h$  hiding places

## 4.2 Single-agent Learning

Elemental tasks can be easily solved by an early stage application of any standard RL algorithm. Each of the learning blocks in the lower level is solved by our earlier work in [8]. In this subsection we develop a model for the higher level learning: learning the task of territory division from the view point of a sequence of elemental tasks. Consider a simple Hide & Seek game with a single seeker. After some early learning stages in which the seeker learned the hiding probability distribution of the hider and learned how to find the shortest path from any location to each of the possible hiding places. Now it is the seeker's task to learn what optimal visiting order is for these possible hiding places that minimizes the expected time to find the hider over the games.

As for the state space, each state is defined as two parts: the set of already visited hiding places  $V$  (an unordered set; as the actual order in the history does not affect the choice of the future actions), and the current location  $loc$  of the seeker which represents the location of the last visited hiding place, i.e.  $s := \langle V, loc \rangle$ , where  $V$  can be represented as a bit string of size  $h$  such that each hiding place is represented by a bit that takes 1 if visited and 0 if unvisited; and  $h$  is the size of  $H$ , the set of possible hiding places in the environment. As for the action space, given state  $s_t$  at time  $t$ , we define the action to be the next unvisited hiding place to be visited by the seeker, i.e.  $a_t \in H - V_t$ , where  $V_t$  is the set of visited hiding places at time  $t$ . The transition function from the current state  $s$  and action  $a$  is defined as  $\delta(s, a) = s'$ ,  $s' := \langle V + a, loc' \rangle$ . The reward function is defined as a function of the hiding probability for the resulting location, i.e.  $R(s, a) = cP_{hiding}(loc')$ , where  $c$  is constant.

It should be noted that this action definition leads to unequal time steps for executing of different actions because the length of the actual actions taken to reach the next unvisited hiding place is different depending on the current location of the seeker and the target hiding place. That requires the standard updating rule of Q-Learning in equation (2) to be revised. Consider at time  $t$  it is required to perform the transition  $\delta(s_t, a_t) \xrightarrow{\tau} s'_{t+\tau}$  that takes duration  $\tau$  for execution. Given Equation (1) and Equation (2), we propose the  $Q$  value to be evaluated as follows:

$$Q(s_t, a_t) \equiv \sum_{k=0}^{\tau-1} \gamma^k r_{t+k+1} + \gamma^\tau V^*(s'_{t+\tau}) \quad (4)$$

From the reward function definition all intermediate transitions have zero reward, therefore,  $r_{t+k+1} = 0$ , for  $k = 0, 1, \dots, \tau - 2$ . Thus the new updating rule will be as follows:

$$Q(s, a) = \gamma^{\tau-1} R(s, a) + \gamma^\tau \max_a Q(s', a') \quad (5)$$

The impact of the discount factor  $\gamma$  reflects the effect of the distance to the hiding places which is absent from the reward function. Notice that the distance factor cannot be simply included as a term in the reward function because that would affect only the reward for the current action and not the next actions, whilst each action should be evaluated according to the total path length from the starting state to the

current state. The new updating rule successfully preserves the total history length through the recursive formula which is crucial for convergence to the optimal strategy.

### 4.3 Multi-agent Cooperative Learning

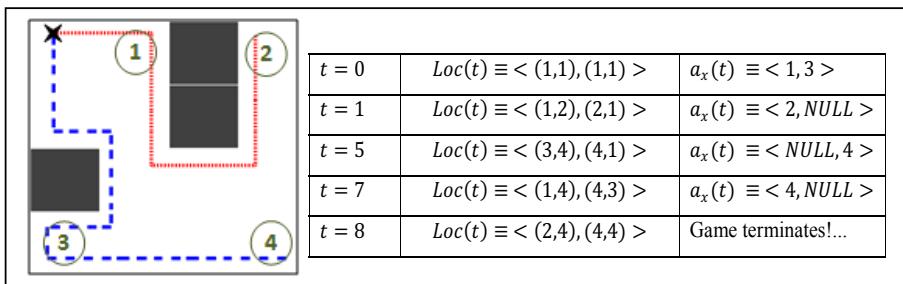
We can extend the model in the previous subsection for the case of multiple agents. First an assumption is required for the learning phase. A global view for the whole system state must be kept, which can be realized by providing full communications between the seekers in the learning phase. Consequently, the state definition would be  $x := \{V, Loc\}$  where  $Loc$  is a vector of the current locations of the seekers, e.g.  $Loc \equiv <loc^1, loc^2, \dots>$ . In addition the action definition would be  $a_x \equiv <a^1, a^2, \dots>$ , where the action  $a^i$  of seeker  $i$  at time  $t$  is given by:

$$a_t^i = \begin{cases} a \in H - V ; P_{hiding}(loc^i) > 0 \\ NULL & ; \end{cases} \quad o.w. \quad (6)$$

where the *NULL* clause means that the agent  $i$  is in an intermediate state and he does not need to take any new action at time  $t$ . A seeker in an intermediate state is performing a set of required actions to transfer from the previous hiding place to the next target one. The essence of *NULL* actions appears only in the case of multiple seekers; the time steps are not equal anymore, so at time  $t$  the time required to perform  $a_t^i$  may not be equal to the time required by  $a_t^j$ ;  $i \neq j$ . That means one seeker will reach a next state while the others are still in intermediate states. Figure 2 shows an example of action interpretation with time steps for two seekers in a  $4 \times 4$  map. Both the reward function and the updating rule for multi-agent case can be generalized from the single-agent case as follows:

$$R_{joint}(x, a_x) = \sum_{seeker i} R(s^i, a^i)$$

$$Q(x, a_x) = \gamma^{\tau-1} R_{joint}(x, a_x) + \gamma^\tau \max_{a'_x} Q(x', a'_x) \quad (7)$$

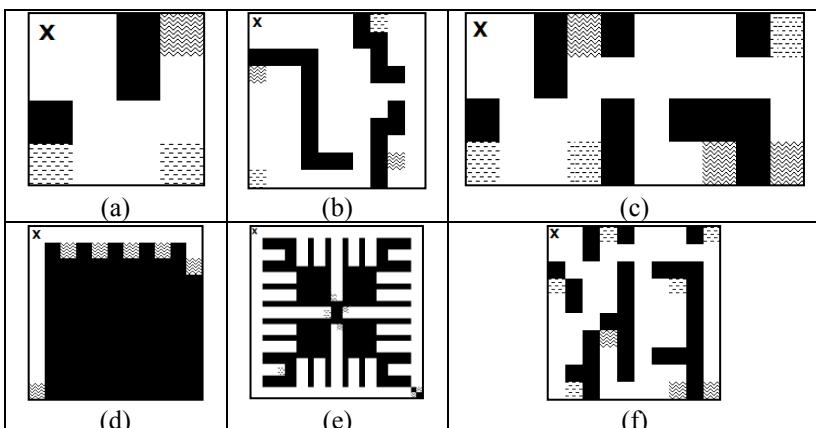


**Fig. 2.** An example of location-action interpretation with time steps for two seekers in a  $4 \times 4$  map. The dashed lines represent the seeking trajectories of the two seekers. Circles represent four possible hiding places. Seekers start from the map origin (1,1) in the top-left corner.

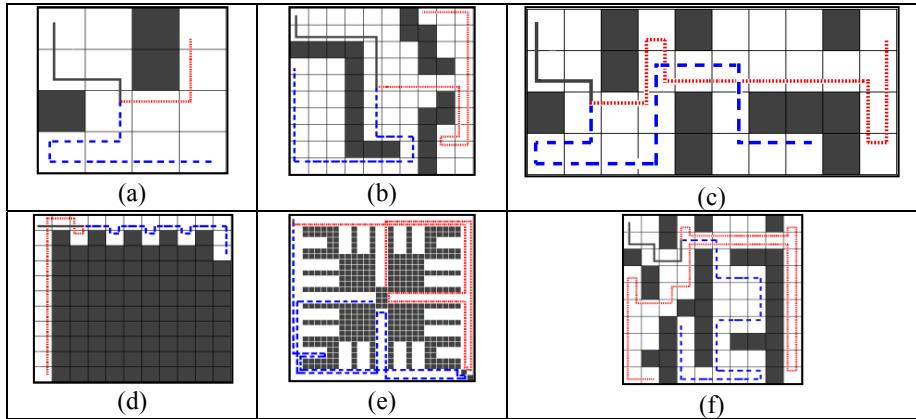
## 5 Experiments

This section discusses some experiments to examine the proposed model. A set of 2D grid maps were used as a testbed for the proposed approach. A seeker has physical action space  $\{N, S, E, W\}$  used in the lower level of learning. Results are to be shown, criticized, and compared with the real optimal solution. Figure 3 shows a set of different maps used for the experiments with different sizes, topologies, and different hiding probability distributions. The X on each map marks a starting position for the adjacent seekers, while black cells represent blocks that seekers cannot go through. Zigzagged and dashed cells mark possible hiding places with different hiding probabilities, for instance a zigzagged cell has hiding probability three times higher than a dashed one. Experiments are done for two seekers on each map. Figure 4 shows the seeking trajectories of two seekers that the learning procedure converges to. It should be noted that the optimal seeking strategy does not require the seekers to visit all the cells, but rather only those related to the possible hiding places. Furthermore, it is not necessarily preferred to visit the nearest hiding place first, because it may have low hiding probability which increase the expected time to find the hider. The results of map (e) illustrate an example of such case. Seekers prefer to visit far but high probability hiding places and then backtrack to lower probability hiding places.

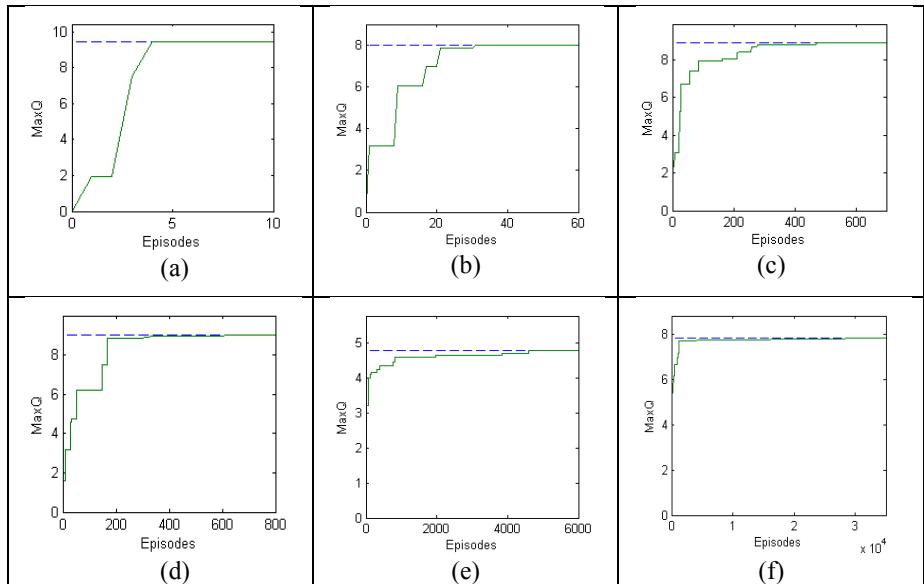
Over time the maximum  $Q$  value for the initial state with all its possible actions gives an indication for the progress of learning, as it converges to the optimal value gained by applying the optimal policy. To check the completeness of the results, they are compared to the results produced by performing an exhaustive search in all the solution space. The comparison showed that the proposed system converges to the real optimal solution for all cases listed here. Figure 5 shows the convergence progress towards the optimal solution. It compares the maximum  $Q$  value of the initial state learnt so far with the actual cumulative discounted reward gained by the optimal solution.



**Fig. 3.** Different maps used for experiments. X marks the starting point of the seekers, zigzagged and dashed cells are possible hiding places.



**Fig. 4.** Results of experiments for two seekers. The dashed lines represent the optimal seeking trajectories while the solid line is an intersection between the two trajectories.  $\gamma = 0.99$ .



**Fig. 5.** The convergence progress over time. It compares the maximum  $Q$  value learnt so far (solid curve) for the initial state, versus the optimal cumulative discounted reward (dashed line) gained by the optimal policy obtained by an exhaustive search. After some learning episodes the maximum  $Q$  value converges to the optimal value. In the action selection method, actions are chosen probabilistically based on the estimated  $Q$  value of the state-action pairs.

As discussed in Section 2, the problem of multi-agent territory division is similar to the  $k$ -Chinese postman problem which is classified NP-complete. So it is expected for the learning approach to suffer from similar problems. But the learning approach has an advantage over a brute force searching algorithm. First it doesn't require a rigorous understanding for the whole problem details, but our proposed model can be applied for different variations of task decomposition with some adjustments. Second and more important, the learning process can give a good sub-optimal solution within a reasonable time. This sub-optimal solution gets closer to the optimal solution by increasing the learning time limit with a reasonable increase rate; and this is not the case with a brute force searching algorithm. This can be observed from the results in Figure 5. With increasing the problem size in the number of hiding places and the map size, the number of learning episodes required till convergence highly increases. For example, map (a) of size  $4 \times 4$  has only 3 hiding places and it required only 5 episodes till convergence. Whereas map (f) of size  $10 \times 10$  has 8 hiding places and it required 30000 episodes, whilst a good sub-optimal solution which is very close to the optimal one is found within 1500 episodes.

The problem of time and space complexity that increase with the problem size can be tackled by the techniques of generalization such as state aggregation. It is like adding another layer in the hierarchical structure discussed in Section 4. We plan to apply our previous work discussed in [8] about state aggregation for Reinforcement Learning. It exploits the idea that agents can group some similar states together into one abstract state and then learn on the abstract level which is much reduced in size. As for the problem of Hide-and-Seek some hiding places reside close to each other, so that it is obvious that a seeker should search all of them together. These close hiding places can form a group that can be used for an abstract level of learning. Since state aggregation is an approximation technique, a slight compromise may be needed between the space reduction efficiency and the absolute optimality.

## 6 Conclusion

In this paper, the problem of multi-agent territory (or task) division is discussed. This problem appears in various real robotics applications that can be modeled as a simple Hide-and-Seek game. The target is not to perform a full coverage of the terrain, but scanning only the possible hiding places so that the total expected time for finding the hider is minimized. That requires finding the optimal sequence of visiting to the hiding places according to the hiding probability. We present a new learning approach to solve this problem using Reinforcement Learning with an illustration of how the learning system is built in a hierarchical structure. In which, the elemental tasks of path planning for each hiding place is learnt first in the lower layer, then the composite task of finding the optimal visiting sequence is learnt in the higher layer. The full module is described including the state, action definitions, the reward function for both single and multiple agents. A revised version of the standard updating rule of Q-Learning is also discussed. This new equation is essential to deal with the problem of unequal time steps for agents' actions. The proposed approach is

examined on a set of different maps in which the learning process converged to the complete solution. The impact of the problem size and the complexity of the system are also studied. The result discussion show that the proposed learning approach has advantages over the algorithmic techniques in that, it is easier to design, can be applied for different variations of task decomposition, and gives a good near-optimal solution within a reasonable time for problems of large space.

**Acknowledgements.** This work has been funded by a scholarship grant from Mitsubishi Corporation International Scholarship to E-JUST which is highly acknowledged.

## References

1. Choset, H.: Coverage for robotics a survey of recent results. *Annals of Mathematics and Artificial Intelligence* 31(1-4), 113–126 (2001)
2. Agmon, N., Hazon, N., Kamink, G.A.: Constructing Spanning Trees for Efficient Multi-robot Coverage. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA (2006)
3. Watkins, C.J.C.H., Dayan, P.: Q-learning. *Machine Learning* 8(3), 279–292 (1992)
4. Singh, S., Jaakkola, T., Littman, M.: Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning* (1998), 287–308 (2000)
5. Singh, S.P.: The Efficient Learning of Multiple Task Sequences. In: Advances in Neural Information Processing Systems, vol. 4, pp. 251–258 (1992)
6. Sutton, R.S.: Learning to predict by the methods of temporal differences. *Machine Learning* 3(1), 9–44 (1988)
7. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 269–271 (1959)
8. Gunady, M.K., Gomaa, W.: Reinforcement learning generalization using state aggregation with a maze-solving problem. In: Proceedings of IEEE Japan-Egypt Conference on Electronics, Communications and Computers (JEC-ECC), pp. 157–162 (2012)
9. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. *Arxiv preprint cs/9605103*, pp. 237–285 (1996)
10. Mitchell, T.M.: Machine Learning book. McGraw-Hill Science/Engineering/Math (1997)
11. Watkins, C.J.C.H.: Learning from Delayed Rewards. PhD thesis, Cambridge Univ., Cambridge, England (1989)
12. Singh, S.P.: Transfer of learning by composing solutions for elemental sequential tasks. *Machine Learning* (1992)
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)

# Modeling Actions Based on a Situative Space Model for Recognizing Human Activities

Juan Carlos Nieves, Dipak Surie, and Helena Lindgren

Department of Computing Science

Umeå University

SE-901 87, Umeå, Sweden

{jcnieves,dipak,helena}@cs.umu.se

**Abstract.** Human activities usually have a motive and are driven by goal directed sequence of actions. Recognizing and supporting human activities is an important challenge for ambient assisted living of elderly in their home environment. By understanding an activity as a sequence of actions, we explore action specification languages for recognizing human activities. In this setting, we analyze the role of the situative space model for modeling indoor human activities in terms of an action specification language.

## 1 Introduction

Ambient intelligence envisions a novel paradigm of computation that is human-centered and occurs in everyday environments augmented with multiple networked computers with the primary purpose of enhancing human experience [1]. Advancements in technology (sensing, computing, communication and interaction), miniaturization of hardware, and the reduction in their cost has brought us within striking distance from experiencing ambient intelligence. Context-awareness [4] is an important feature of ambient intelligence where the networked computers are expected to have an implicit understanding of the environmental situations which improves the richness of communication between humans and computing systems, and enables providing services that make sense in the current situation. Context-awareness that is centered on human agents and their activities enable the development of personalized and adaptive smart environments [12]. Such environments facilitate enhanced activity performance useful for elderly people to lead an independent life at home without (or prolong) the need to move to a care center bringing in economical and humane benefits. Also, monitoring in-home activities provide valuable insights into the health status of the occupant useful in providing health care services.

Understanding and inferring human activities and the context in which they take place is an interesting research challenge. Human activities take place at multiple levels simultaneously: from the level of body and body-part movements, to the interaction with situated objects, to performing goal-directed actions to performing complex activities with clearly defined motives [12]. While there are

several approaches to activity recognition in a smart home [18,21,20,10] that are of importance, such approaches are not based on a formal definition of an activity and their associated context thereby answering “*what activity was performed?*” but are insufficient in answering “*how the activity was performed?*” and “*how can we be sure that the activity was performed?*” A smart home worthy of its name in providing activity support requires additional knowledge about an activity and better qualitatively describe the activity performed [3].

We take a logic declarative approach in defining and describing human activities that improves the quality of inferring activities by complementing existing approaches. An egocentric interaction [16] framework is used for modeling human interaction in an environment, and for framing the context associated to a human agent and their activities using a situative space model [15] that maintains the human agent’s bodily situation in that environment. The situative space model is inspired from how biological agents frame their environment and survive in it based on perception and action [9].

Several candidate approaches in knowledge representation techniques for managing actions and change are available [2,11]. Given the situative space model’s inclusion of human actions as a corner stone, action specification languages are suitable among other specification languages for supporting reasoning about human activities [2,7,6]. Among the different action specifications which can be found in the literature, those which are based on non-monotonic formalism have shown to be expressive enough for capturing sophisticated domains such as: Space Shuttle [13], Biological Networks [5] and Social Norms [14]. The formalization of these languages are based on different formal methods. To maintain simple specifications, we consider a basic language of actions specification called  $\mathcal{A}$  and its semantics has been defined in terms of Answer Set Programming [8]. An analysis of human activities using the situative space model to support human activity reasoning and the importance of human perception for human activity analysis are presented inspired from our earlier work [20]. In particular, we focus our analysis on a breakfast scenario taking place in a home environment. The breakfast scenario is taken since preparing food and eating well removes malnutrition prevalent among elderly people living alone. This scenario takes place in a smart kitchen equipped with kitchen appliances and objects augmented with different types of sensors capable of delivering streams of relevant data.

The rest of the paper is divided as follows: In Section 2, some background about action specifications languages is introduced. In Section 3, a human activity scenario is analyzed in terms of the situative space model which is used in identifying some exemplary fluents and actions to support a declarative specification of a human activity like preparing coffee. In Section 4, some potential specifications based on  $\mathcal{A}$  are discussed followed by the conclusions.

## 2 Background

In this section, we introduce some basic concepts of action specification languages and take the language  $\mathcal{A}$  as a starting point also introduced in [8] and has been

extended in several direction [2]. The alphabet of the language  $\mathcal{A}$  consists of two nonempty disjoint sets of symbols  $\mathbf{F}$  and  $\mathbf{A}$ . They are called the set of fluents  $\mathbf{F}$  and the set of actions  $\mathbf{A}$ . A *fluent* expresses the property of an object in a world, and forms part of the description of states of this world. A *fluent literal* is a fluent or a fluent preceded by  $\neg$ . A *state*  $\sigma$  is a collection of fluents. We say a fluent  $f$  holds in a state  $\sigma$  if  $f \in \sigma$ . We say a fluent literal  $\neg f$  holds in  $\sigma$  if  $f \notin \sigma$ .

*Situations* are representations of the history of action execution. In the initial situation no action has been executed: we represent this by the empty list  $[]$ . The situation  $[a_n, \dots, a_1]$  corresponds to the history where action  $a_1$  is executed in the initial situation followed by  $a_2$ , and so on until  $a_n$ . There is a simple relation between situation and state. In each situation  $s$  certain fluents are true and certain others are false, and this is a *state of the world*. Usually, the syntax of  $\mathcal{A}$  is presented in three sub-languages:

**Domain description:** The domain description language is used to succinctly express the transition between states due to actions. It consists of *effect propositions* of the following form:

(1) *a cause f if*  $p_1, \dots, p_n, \neg q_{n+1}, \dots, \neg q_r$

where  $a$  is an action,  $f$  is a fluent literal, and  $p_1, \dots, p_n$  and  $q_{n+1}, \dots, q_r$  are fluents. Intuitively, the above proposition means that if the fluent literals  $p_1, \dots, p_n, \neg q_{n+1}, \dots, \neg q_r$  hold in the state corresponding to a situation  $s$  then in the state corresponding to a situation reached by executing  $a$  (denoted by  $[a|s]$ ) a fluent literal  $f$  must hold. If both  $n$  and  $r$  are equal to 0 in (1) then we simply write

(2) *a cause f*

**Observation language:** A set of observations  $O$  consists of value propositions of the following form:

(3) *f after*  $a_1, \dots, a_m$

where  $f$  is a fluent literal and  $a_1, \dots, a_m$  are actions. Intuitively, the above value proposition means that if  $a_1, \dots, a_m$  would be executed in the initial situation then in the state corresponding to the situation  $[a_m, \dots, a_1]$ ,  $f$  would hold. When  $a_1, \dots, a_m$  is an empty sequence, we write the above as follows:

(4) *initially f*

In this case the intuitive meaning is that  $f$  holds in the initial state corresponding to the initial situation.

**Query language:** Queries consists of value propositions of the form (3)

The role of effect propositions is to define a *transition function* from states and actions to states. Given a domain description  $D$ , such a transition function  $\Phi$  should satisfy the following properties. For all actions  $a$ , fluents  $f$  and states  $\sigma$ :

- if  $D$  includes an effect proposition of the form (1) where  $f$  is the fluent  $g$  and  $p_1, \dots, p_n, \neg q_{n+1}, \dots, \neg q_r$  hold in  $\sigma$  then  $g \in \Phi(a, \sigma)$ ;
- if  $D$  includes an effect proposition of the form (1) where  $f$  is a negative fluent literal  $\neg g$  and  $p_1, \dots, p_n, \neg q_{n+1}, \dots, \neg q_r$  hold in  $\sigma$  then  $g \notin \Phi(a, \sigma)$ ; and
- if  $D$  does not include such effect propositions, then  $g \in \phi(a, \sigma)$  iff  $g \in \sigma$ .

If such a transition function exists, then we say that  $D$  is *consistent* and refer to its transition function by  $\Phi_D$ . Given a consistent domain description  $D$  the set of observations  $O$  is used to determine the states corresponding to the initial situation, referred as the initial states and denoted by  $\sigma_0$ . While  $D$  determines a unique transition function, and  $O$  may not always lead to a unique initial state. We say  $\sigma_0$  is a initial state corresponding to a consistent domain description  $D$  and a set of observations  $O$ , if for all observations of the form (3) in  $O$ , the fluent literal  $f$  holds in the state  $\Phi(a_m, \Phi(a_{m-1}, \dots, \Phi(a_1, \sigma_0) \dots))$ . We will denote this state by  $[a_m, \dots, a_1]\sigma_0$ . We say that  $(\sigma_0, \Phi_D)$  *satisfies*  $O$ .

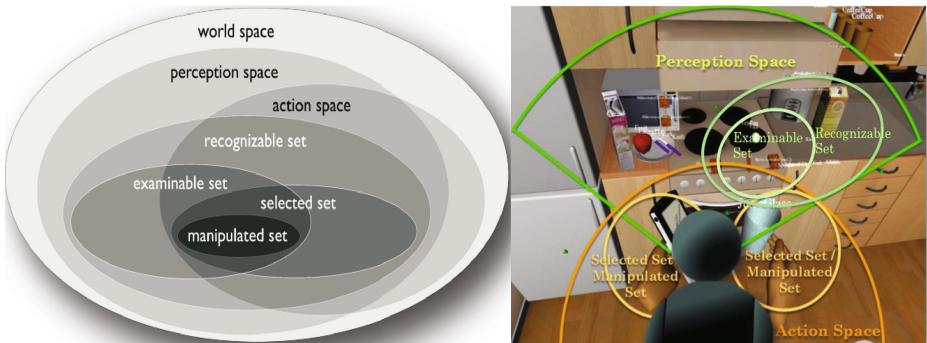
Given a consistent domain description  $D$  and a set of observation  $O$ , we refer to the pair  $(\Phi_D, \sigma_0)$ , where  $\Phi_D$  is the transition function of  $D$  and  $\sigma_0$  is an initial state corresponding to  $(D, O)$ , as a model of  $(D, O)$ . We say that  $(D, O)$  is *consistent* if it has a model and say it is *complete* if it has a unique model. We say a consistent domain  $D$  in the presence of a set of observations  $O$  entails a query  $Q$  of the form (3) if for all initial states  $\sigma$  corresponding to  $(D, O)$ , the fluent literal  $f$  holds in the state  $[a_m, \dots, a_1]\sigma_0$ . We denote this as  $D \models_O Q$ .

### 3 Human Activity Scenario

In this section, we introduce *the situative space model* as a reference model for analyzing and designing an action specification of human activities. In this context, the issue of how relevant is the concept of *perception* of a human with respect to the world in which he/she is situated is addressed. We motivate seven sets/spaces [15][18], *i.e.*, *Word Space*, *Perception Space*, *Recognizable Set*, *Exam-  
inable Set*, *Action Space*, *Selected Set*, and *Manipulated Set* for identifying fluents and actions which are relevant in recognizing human activities.

In our activity scenario, Björn is a 68 years old widow living alone in his smart home. During the breakfast scenario, he is performing the activity of preparing coffee among other activities like preparing bread toast, boiling an egg, and preparing oats meal. The breakfast scenario is taken since preparing food and eating well removes malnutrition prevalent among elderly people living alone. Björn's smart kitchen is equipped with kitchen appliances and objects augmented with different types of sensors including temperature, humidity, force, infrared distance, light and RFID readers capable of delivering streams of data relating to Björn's current situation captured using the situative space model described in detail elsewhere [15].

The situative space model is centered on Björn (or any other human agent) and varies continuously according to his body and body part movements: it is intended to capture what Björn can perceive and not perceive, affect and not affect at a given moment in time, being inspired by his situatedness and proximity to nearby objects in the kitchen that facilitates him in performing activities. The situative space model represents perception and action possibilities in a unified manner acknowledging the different levels and multiple modalities. The breakfast scenario is described using an operational definition of the situative spaces limited to visual and touch modalities:



**Fig. 1.** The situative space model and its representation in an immersive virtual reality kitchen environment

**World Space (WS):** A space containing the set of all objects that are part of Björn’s smart home.

**Perception Space (PS):** The part of the space around Björn that can be visually perceived at each moment, like a cone as a rough approximation. Refer to Fig. 1. Objects may occlude other objects and thus create (temporary) holes in this space. During the breakfast scenario, objects like a stove, coffee machine, coffee powder, coffee filter, bread toaster, refrigerator, sink, coffee cup, milk packet, etc. enter and leave Björn’s perception space as a driving factor in performing breakfast activities. In some sense, the location context framed by Björn’s perception capabilities is included in this space.

**Recognizable Set (RS):** The set of objects currently within Björn’s perception space that are within his recognition distances, i.e., it is possible for Björn to recognize the type of objects and what can be done with it. In this scenario, Björn is able to recognize that he can use the coffee machine for preparing coffee, however is not able to recognize the new coffee filter shopped by his caregiver thereby leaving it out of the recognizable set. The recognizable set includes coffee machine (recognized as for preparing coffee), stove (recognized as for preparing oats) and refrigerator (recognized as for storing food cool).

**Examinable Set (ES):** The set of objects currently within Björn’s perception space that are within his examination distances, i.e., it is possible for Björn to not only recognize the objects but also their states. Normally, the examinable set is a proper subset of the recognizable set. In this scenario, Björn is used to preparing coffee using the coffee machine, and can not only recognize it, but can also examine its state to realize the lack of coffee powder that drives him to bring some coffee powder and fill-up the machine while performing the activity of preparing coffee. The examinable set includes coffee machine (examined to have no coffee powder) and bread toaster (examined to contain a piece of bread).

**Action Space (AS):** The part of the space around Björn’s body that is currently accessible to perform actions without locomotion, while small body part movements like lifting the hand up to take the sugar jar from the cupboard or bending to take out the coffee machine from a shelf is allowed. Objects within this space can be directly acted on while some of the objects in the perception space need human locomotion before they enter the action space. While preparing coffee, Björn’s action space is expected to be filled with objects like the coffee machine, coffee powder, coffee cup, milk packet, sugar jar and a spoon among other objects based on physical proximity.

**Selected Set (SS):** The set of objects currently being handled by Björn with his hands (touched, gripped; or selected in the virtual sense). In the scenario, Björn opens the coffee powder slot and pours coffee powder into the machine, thereby including the coffee machine and the coffee powder packet in his selected set.

**Manipulated Set (MS):** The set of objects whose states (external as well as internal) are currently in the process of being changed by Björn. Normally, the manipulated set is a subset of the selected set. In this scenario, Björn first selects the coffee machine before turning it on (internal state), and takes the milk packet from the refrigerator changing the external state of the milk packet from *inside the refrigerator* to *on the counter* while performing the activity of preparing coffee.

Since many actions require perception to be effective, the current shape of the perception space qualitatively affects the action space. Perception space can change with actions like mobility where a coffee cup comes into the perception space when Björn moves to the cupboard, opens it and fetches a coffee cup. Also, the perception space (specifically the examinable set) changes due to objects that change their state like the coffee machine that turns off automatically after making coffee. At this point, let us observe that traditional action and change models, *e.g.*, *situation calculus* and *event calculus* are oriented to identify fluents and actions in the selected and manipulated sets while ignoring perception, recognizable and examinable spaces/sets. An important issue *w.r.t.* these sets is that these sets suggest fluents and actions that can consider incomplete information *w.r.t.* the world. Hence, the uncertainty which is attached to the objects that take part or momentarily interrupt an activity can be dynamic with respect to time and human perception information could be used to address the emergent behavior of human agents while performing activities.

### 3.1 Fluents and Actions

In Table 1, we concretize some exemplary fluents and actions which are relevant for action specification based on the situative space model and useful in modeling activities [16]. For the individual objects, we have identified actions as: *move\_towards* and *move\_away*. These actions do not change the state of

**Table 1.** Actions and Fluents

Object - Area	Fluent	Action
Coffee machine ( $O_1$ ) - cooking	is_on {true, false} is_on_time {min} has_coffee_powder {true, false} has_water {true, false}, <i>in_situative_spaces</i> {world space, perception space, recognizable set, examinable set, action space, selected set, manipulated set}	[activate_O <sub>1</sub> , deactivate_O <sub>1</sub> ] [activate_O <sub>1</sub> , deactivate_O <sub>1</sub> ] [add_coffee_powder_O <sub>1</sub> , remove_coffee_powder_O <sub>1</sub> ] [add_water_O <sub>1</sub> ] [move_towards_O <sub>1</sub> , move_away_O <sub>1</sub> ]
Stove ( $O_2$ ) - cooking	is_on {true, false} is_on_time {min} has_heat_level {very low, low, medium, high, very high} contains_object {true: object_id, false: null} <i>in_situative_spaces</i> {world space, perception space, recognizable set, examinable set, action space, selected set, manipulated set}	[activate_O <sub>2</sub> , deactivate_O <sub>2</sub> ] [activate_O <sub>2</sub> , deactivate_O <sub>2</sub> ] [increase_heat_O <sub>2</sub> , decrease_heat_O <sub>2</sub> ] [add_object_O <sub>2</sub> : object_id, remove_object_O <sub>2</sub> : object_id] [move_towards_O <sub>2</sub> , move_away_O <sub>2</sub> ]
Bread toaster ( $O_3$ ) - cooking	is_on {true, false} has_heat_level {low, medium, high} <i>in_situative_spaces</i> {world space, perception space, recognizable set, examinable set, action space, selected set, manipulated set}	[activate_O <sub>3</sub> , deactivate_O <sub>3</sub> ] [increase_heat_O <sub>3</sub> , decrease_heat_O <sub>3</sub> ] [move_towards_O <sub>3</sub> , move_away_O <sub>3</sub> ]
Refrigerator & Freezer ( $O_4$ ) - storage	is_open_r {true, false} is_open_f {true, false} r.contains_objects {[list of object_ids, null]} f.contains_objects {[list of object_ids, null]} object_added_r (object_id) object_added_f (object_id) object_removed_r (object_id) object_removed_f (object_id) r.has_temperature_level {very low, low, medium, high, very high} f.has_temperature_level {very low, low, medium, high, very high} r.in_situative_spaces {world space, perception space, recognizable set, examinable set, action space, selected set, manipulated set} f.in_situative_spaces {world space, perception space, recognizable set, examinable set, action space, selected set, manipulated set}	[open_O <sub>4</sub> .r, close_O <sub>4</sub> .r] [open_O <sub>4</sub> .f, close_O <sub>4</sub> .f]  [add_object_O <sub>4</sub> .r: object_id] [add_object_O <sub>4</sub> .f: object_id] [remove_object_O <sub>4</sub> .r: object_id] [remove_object_O <sub>4</sub> .f: object_id] [increase_heat_O <sub>4</sub> .r, decrease_heat_O <sub>4</sub> .r] [increase_heat_O <sub>4</sub> .f, decrease_heat_O <sub>4</sub> .f] [move_towards_O <sub>4</sub> .r, move_away_O <sub>4</sub> .r]  [move_towards_O <sub>4</sub> .f, move_away_O <sub>4</sub> .f]
Sink ( $O_5$ ) - washing	tap.open {true, false} water_temperature {hot, warm, medium, cool, cold} dishing_liquid_used {true, false} dishing_brush_used {true, false} <i>in_situative_spaces</i> {world space, perception space, recognizable set, examinable set, action space, selected set, manipulated set}	[open_tap_O <sub>5</sub> , close_tap_O <sub>5</sub> ] [increase_temperature_O <sub>5</sub> , decrease_temperature_O <sub>5</sub> ] [select_dishing_liquid_O <sub>5</sub> , deselect_dishing_liquid_O <sub>5</sub> ] [select_dishing_brush_O <sub>5</sub> , deselect_dishing_brush_O <sub>5</sub> ] [move_towards_O <sub>5</sub> , move_away_O <sub>5</sub> ]
Coffee powder, cup, milk, sugar... ( $O_{i=6,7,8,...n}$ ) - Tagged objects	is_used {true, false} is_placed_on {object_id, null}	[add_O <sub>i</sub> _to_object: object_id, remove_O <sub>i</sub> _to_object: object_id] [add_O <sub>i</sub> _to_object: object_id, remove_O <sub>i</sub> _to_object: object_id]

the objects themselves but changes the perception (perception space) and the action possibilities (action space) of an intelligent agent that can eventually change the state of the object. These actions are conditioned by fluents such as *in\_situative\_spaces*<sup>1</sup>.

## 4 Reasoning in $\mathcal{A}$ Language

Now that the basic elements such as fluents and actions have been identified for supporting an action specification of human activity, this section will introduce some domain basic descriptions about the breakfast scenario, especially for the context of preparing coffee.

The domain description will describe changes of states of a coffee machine. The effect of the actions `remove_coffee_powder`, `add_coffee_powder`, `add_water`, `coffee_machine_on` and `coffee_machine_off` can be expressed by the following effect propositions:

<sup>1</sup> For the practical point of view, fluents such as *in\_situative\_spaces* are modified by sensors which are augmented within the object.

remove_coffee_powder cause $\neg$ has_coffee_powder if has_coffee_powder, old_coffee_powder, coffee_machine_in_action_space.
add_coffee_powder cause has_coffee_powder if $\neg$ has_coffee_powder, coffee_machine_in_action_space.
add_water cause has_water if $\neg$ has_water, coffee_machine_in_action_space.
coffee_machine_on cause coffee_machine_is_on if $\neg$ coffee_machine_is_on, has_water, has_coffee_powder, coffee_machine_in_action_space.
coffee_machine_off cause $\neg$ coffee_machine_is_on if coffee_machine_is_on, $\neg$ has_water, coffee_machine_in_action_space.
coffee_machine_on cause old_coffee_powder if has_coffee_powder.

Let us observe that all the actions have as precondition that the coffee machine is in the action space. From the perception point of view, one can say that the coffee machine is in the action space of an intelligent agent whenever the intelligent agent move towards to the coffee machine:

move_towards_coffee_machine cause coffee_machine_in_action_space
--

Another issue to observe in the activity of preparing coffee is that even though a situation like [add\_coffee\_powder, remove\_coffee\_powder] suggests some changes in the coffee machine, this situation does not mean that coffee was prepared recently. For inferring that coffee was prepared recently, one need to explain sequence of actions of the form:

[coffee_machine_off, coffee_machine_on]
[coffee_machine_off, coffee_machine_on, add_coffee_powder].
[coffee_machine_off, coffee_machine_on, add_water].
[coffee_machine_off, coffee_machine_on, add_water, add_coffee_powder].
[coffee_machine_off, coffee_machine_on, add_coffee_powder, add_water].

By keeping in mind that an activity comprises of an history of events (as the previous sequence of actions) which we want to explain, it is important to explain the changes of fluents and the effects of actions in a given environment. Hence, the observation languages introduced in Section 2 has to be extended in order to capture a given history of events. In this setting, we follow the approach introduced in [2]; therefore, propositions of the following form are considered:

<i>a occurs_at k</i>	<i>f occurs_at k</i>
----------------------	----------------------

where *a* is an action symbol, *f* is a fluent symbol and *k* is a non-negative number. The intuitive meaning of the first proposition is that the action *a* occurs at time point *k* and the intuitive meaning of the second proposition is that the fluent *f* is observed to be true at time point *k*.

Given the semantics of the *A* language, one can perform different types of queries with respect to an activity:

- To explain the current state of a world space, *e.g.*, Has Björn prepared breakfast in the morning?
- To predict potential activities in the near future, given the current perception and action spaces which are activity drivers inspiring Björn to take-up an activity or two?

While the problem of sensing the situative spaces is addressed elsewhere [17,19], in the current implementation, we have been mapping our specification into logic normal program and are using answer set solvers like CLASP<sup>2</sup>. For mapping our specifications, we are using the mapping suggested in [2]. In the long version of this paper we are going to present the full implementation of the reasoning process.

## 5 Conclusions

Understanding and inferring human activities and the context in which they take place is an important research challenge. The relevance of this research challenge has been increased by the need for novel technologies that support older adults with health care and daily living. Accurate sensing and reasoning of human activities, especially in their home allows for the prevention of critical health issues that might occur in the future, while monitoring their activities despite privacy issues could contributes to the empowerment of older adults in enjoying a safe, secure and independent life. In this paper, we have argued for supporting human activity reasoning using a situative space model that suggests aspects of the environment captured as a perception space can enrich the specification of activities in terms of action specification languages.

**Acknowledgement.** This research is partly funded by VINNOVA (Sweden's innovation agency) and the Swedish Brain Power.

## References

1. Aarts, E., de Ruyter, B.: New research perspectives on Ambient Intelligence. *J. Ambient Intell. Smart Environ.* 1(2), 5–14 (2009)
2. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press, Cambridge (2003)
3. Cook, D., Schmitter-Edgecombe, M.: Assessing the quality of activities in a smart environment. *Methods of Information in Medicine* 48, 480–485 (2009)
4. Dey, A.K.: Understanding and using context. *Personal Ubiquitous Comput.* 1, 4–7 (2001)
5. Dworschak, S., Grell, S., Nikiforova, V.J., Schaub, T., Selbig, J.: Modeling Biological Networks by Action Languages via Answer Set Programming. *Constraints* 13(1–2), 21–65 (2008)
6. Eiter, T., Faber, W., Leone, N., Pfeifer, G., Polleres, A.: A logic programming approach to knowledge-state planning, ii: The DLV<sup>k</sup> system. *Artif. Intell.* 144(1–2), 157–211 (2003)
7. Gebser, M., Grote, T., Schaub, T.: Coala: A Compiler from Action Languages to ASP. In: Janhunen, T., Niemelä, I. (eds.) *JELIA 2010. LNCS*, vol. 6341, pp. 360–364. Springer, Heidelberg (2010)

---

<sup>2</sup> <http://www.cs.uni-potsdam.de/clasp/>

8. Gelfond, M., Lifschitz, V.: Representing Action and Change by logic programs. *Journal of Logic Programming* 17(2,3,4), 301–323 (1993)
9. Gibson, J.J.: The Ecological Approach to Visual Perception. Houghton Mifflin (1979)
10. Kim, E., Helal, S., Cook, D.: Human activity recognition and pattern discovery. *IEEE Pervasive Computing* 9, 48–53 (2010)
11. McCain, N., Turner, H.: Causal Theories of Action and Change. In: Proc. AAAI 1997, pp. 460–465 (1997)
12. Nardi, B.: Context and consciousness: activity theory and human computer interaction. MIT Press (1996)
13. Nogueira, M., Balduccini, M., Gelfond, M., Watson, R., Barry, M.: An A-Prolog Decision Support System for the Space Shuttle. In: Ramakrishnan, I.V. (ed.) PADL 2001. LNCS, vol. 1990, pp. 169–183. Springer, Heidelberg (2001)
14. Panagiotidi, S., Nieves, J.C., Vázquez-Salceda, J.: A framework to model norm dynamics in answer set programming. In: Proceedings of FAMAS 2009 (Multi-Agent Logics, Languages, and Organisations Federated Workshops). CEUR WS Proceedings, vol. 494 (2009)
15. Pederson, T., Janlert, L.-E., Surie, D.: A situative space model for mobile mixed-reality computing. *IEEE Pervasive Computing* 10(4), 73–83 (2011)
16. Surie, D.: Egocentric Interaction for Ambient Intelligence. PhD thesis, Dept. of Computing Science, Umeå University (2012)
17. Surie, D., Jackel, F., Janlert, L.-E., Pederson, T.: Situative space tracking within smart environments. In: International Conference on Intelligent Environments. IEEE Computer Society (2010)
18. Surie, D., Lagriffoul, F., Pederson, T., Sjölie, D.: Activity Recognition Based on Intra and Extra Manipulation of Everyday Objects. In: Ichikawa, H., Cho, W.-D., Satoh, I., Youn, H.Y. (eds.) UCS 2007. LNCS, vol. 4836, pp. 196–210. Springer, Heidelberg (2007)
19. Surie, D., Laguionie, O., Pederson, T.: Wireless sensor networking of everyday objects in a smart home environment. In: International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp. 189–194 (2008)
20. Surie, D., Pederson, T., Lagriffoul, F., Janlert, L.-E., Sjölie, D.: Activity Recognition Using an Egocentric Perspective of Everyday Objects. In: Indulska, J., Ma, J., Yang, L.T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 246–257. Springer, Heidelberg (2007)
21. Wang, L., Gu, T., Tao, X., Lu, J.: Sensor-Based Human Activity Recognition in a Multi-user Scenario. In: Tscheligi, M., de Ruyter, B., Markopoulos, P., Wichert, R., Mirlacher, T., Meschterjakov, A., Reitberger, W. (eds.) AmI 2009. LNCS, vol. 5859, pp. 78–87. Springer, Heidelberg (2009)

# Management of Urban Parking: An Agent-Based Approach

Nesrine Bessghaier<sup>1,2</sup>, Mahdi Zargayouna<sup>1</sup>, and Flavien Balbo<sup>1,2</sup>

<sup>1</sup> Université Paris-Est, IFSTTAR, GRETIA, Noisy le Grand, France

hamza-mahdi.zargayouna@ifsttar.fr

<sup>2</sup> Université Paris Dauphine, Laboratoire Lamsade-CNRS, Paris, France

{nesrine.bessghaier,balbo}@lamsade.dauphine.fr

**Abstract.** In the context of road urban traffic management, the problem of parking spots search is a major issue because of its serious economic and ecological fallout. In this paper, we propose a multi-agent system that aims to decrease, for private vehicles drivers, the parking spots search time. In the system that we propose, a community of drivers shares information about spots availability. Our solution has been tested following different configurations. The first results show a decrease in parking spots search time.

## 1 Introduction

The growing density of urban populations and the complexity of cities infrastructure are challenges that our societies must face. In the domain of transportation, the management of urban traffic growth is one of the important issues. For instance, inside the European union, the congestion cost represents several billions of Euros, and 1% of the Union GDP. Besides the financial consequences, transport has serious ecological fallouts since it is responsible for a large part of carbonic gas emission. The pollution - including noise - generated by traffic has become a serious obstacle to the quality of life and to the health of urban populations. An important part of these emissions is due to “search traffic”, that is the traffic generated by drivers looking for parking spots. A study [8] performed in the French cities of Grenoble, Lyon and Paris in 2005 reveals that, in average 10 % of the moving vehicles at a moment are looking for a parking and until 60% in the small parisian streets. This search traffic leads to 10 millions hours lost per year in France. The design of an intelligent transport system (ITS) dedicated to the management of parking spots should allow to limit the cost of this useless activity. An ITS for urban parking is different from those designed for the management of parking in reserved spots (car park, road infrastructure, etc.). The issue here is to adapt to a complex problem necessitating the consideration of a dynamic and open environment. The solution provided to this problem has to use minimal information on a shared, volatile and uncontrollable resource. The available information is minimal because, without a dedicated infrastructure for parking, there is no information resource about the availability of spots. In addition, the size of the vehicles and the space between them are characteristics that condition locally the number of available spots in a given space. Since the context is urban parking, the resource is shared and accessible to everyone

in an uncontrolled manner. The availability of a spot is volatile and depends totally of the activity of the transport network. It is then necessary to provide a solution in which the information management adapts to the characteristics of this resource. Thus, an ITS has to be able to function without initial information and ensure to its users to have an information that is the most up-to-date possible. Finally, in an important agglomeration, the management of the volatility of this information might need important information flows. Indeed, in order to have up-to-date information of the parking evolution, we have to update the data for each vehicle that leaves or takes a spot.

In this paper, we propose an agent-based transport information system that helps to find parking spots in an urban agglomeration. The multi-agent paradigm facilitates an approach by analogy in the transportation domain, one of the objectives of which is the coordination of distributed entities to insure the safety and efficiency of transport systems. This is why the multi-agent approach is often chosen to model, solve and/or simulate transportation problems. This approach is particularly relevant for the management of parking spots, since the problem is indeed to take into account human behaviors that interact in a complex, dynamic and open environment. We propose a MAS that facilitates the information sharing relative to parking for a community of drivers. In our system, agents communicate to exchange information concerning the parking spots availability. The information comes from the users and their collaboration has to ensure an information of good quality.

There exists commercial transport applications that use an information sharing between users, for the localization of mobile speed cameras for instance. These systems use information which volatility is not very high and for which a centralized architecture could be adequate. In our case, we choose a totally decentralized architecture with an inter-vehicular communication (V2V) to allow vehicles to receive and broadcast information to the other vehicles of the same community. This communication support opens new perspectives in terms of services and has been used for instance to improve road safety [15] and the management of traffic lights [5]. Our solution supports a collaborative process grounded on mutual awareness. Each agent receives the information within its reach and participates to its update. The local information processing avoids centralization, which is costly and makes the scalability more difficult.

The remainder of this paper is organized as follows. In section 2 we describe our multi-agent model. We describe our simulations setup and report our results in section 3. We discuss related work in section 4 before to conclude and describe the perspectives of this work.

## 2 System Design

### 2.1 Agents Model

Our system for the search of spots in an urban area is modeled by a type of agent designated by *assistant* agent. An assistant agent assists one driver of a vehicle by helping him choosing among the available spots that suits him the most. The internal architecture of the assistant agent is composed of three modules: a *Communication* module, an *Itinerary* module and a *Decision* module. The first module enables the agent to communicate with its neighbors in the community. This communication is based on messages

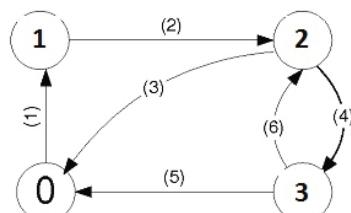
and allows to exchange information about the availability of parking spots. The itinerary module ensures the calculation of the route to a particular parking spot starting from the driver's current position, but also monitors its movement. Finally, the decision module takes care of the decision making. This module proposes a parking spot to the driver. The latter must meet the criteria specified by the driver, which may concern for instance its distance, the time since its release, or the safety of its location. In addition to proposing parking spots, the decision module manages a memory containing information related to the spots. This knowledge evolves over time with information acquired through the exchange of messages with different assistant agents and to the perception of the agent. This memory is composed of two disjoint lists:

- The FS list (for Free Spot): It is a set of pairs  $\{<spot, time>\}$ , where each one refers to a specific spot: its geographic position and the moment since which it was released.
- The OS List (for Occupied Spots) : It contains the spots that were in FS but which turned out to be occupied with the moment since which this information was known.

Both lists are exchanged by the assistant agents and are updated gradually by the knowledge of each one. Their combined use provides a dynamic update of the system information. Indeed, one consequence of the volatility of information regarding the availability of spots is illustrated when an agent chooses a spot on its list FS - supposed to be free - but, once there, it finds it occupied. In this case, the FS lists contain incorrect information about this spot. In the absence of a centralized component that would filter this information from the memory of agents, we need a mechanism to limit the spread of information. Thus, the OS list enables agents to filter the information received and to have the best information possible. Besides, to allow an update of the list without specific information, the decision module of each assistant agent shall filter outdated information after a time  $\theta$ , i.e the spots in FS and OS with an associated time that is inferior to the current time minus  $\theta$ . This parameter takes into account the network activity. Thus, a low value reflects a high volatility as the case may be in rush hours in downtown, while a high value keeps a longer sharing of information and reflects, for instance, the lower volatility in a residential area.

The assistant agent passes through four states as indicated by the automata of Fig 1.

- state 0: The agent is inactive, he does not participate in the community (that means that the vehicle is parked).



**Fig. 1.** Assistant agent state diagram

- state 1: The agent is active, he participates in the community, while the driver is moving (without looking for a spot).
- state 2: The agent looks for a parking spot to propose it to the driver of the concerned vehicle. Meanwhile, the driver looks for a spot on its own.
- state 3: The agent has proposed a spot and the corresponding itinerary to the driver. He stays aware of possible spots alternatives which would be more suitable than the proposed one. Meanwhile, the driver moves towards the proposed spot.

Starting from state 0, the assistant agent goes to state 1 when the driver releases a parking spot(arc (1) in Fig. 1). When the driver is near his destination, the assistant agent switches to state 2 (arc (2)) and starts looking for an available spot. If the agent cannot propose spot (the FS list is empty for example), the driver keeps on driving and looking by himself for a place while the assistant agent tries to update its knowledge in order to find a free parking spot(i.e. it remains in state 2). In this case, if the assistant agent cannot offer places but the driver finds one on its way, then agent returns to state 0(arc (3)). However, if the agent proposes a place to the driver, together with its itinerary, it proceeds to state 3(arc (4)) and the driver goes to the chosen spot. Finally, from state 3, it can go to:

- state 0, if the driver finds a spot on his way that suits him better than the one proposed or when he arrives at the chosen spot that stills free. So the driver parks his vehicle and the agent will be stopped.
- state 2 and the search cycle starts again. This happens when the driver arrives at the place and find it already taken (for instance, a driver from outside the community would have found it)

We have chosen to make the agents communicate via an inter-vehicular network. This choice allows the information exchanged to move following two vectors. The first is specific to the communication. Indeed, the messages exchange takes between each two neighboring vehicles in the same community, and by transitivity agents can be informed of the availability of spots. The second vector concerns the movement of vehicles that mechanically move their information. However, the broadcast of information within the community can lead to a deterioration in the quality and in the effectiveness of the system. There is quality degradation if an isolated agent cannot access or share its information. The effectiveness of the system can be challenged by a very large number of communications. Indeed, the information update is based on a restricted broadcast that depends on the location, but this communication is systematic. The density of the network can generate a large number of messages. However, the communications take place very locally between vehicles.

## 2.2 Cooperation Model

Our MAS is based on the cooperation of agents to share information regarding the availability of spots. This cooperation uses two types of broadcast. The first type concerns all the information that the agent has when not looking for a parking spot. Otherwise, it only broadcasts information that are not interesting. The messages exchanged between

assistant agents from the same community include their lists ( $FS_A$  and  $OS_A$ ) which contain, respectively, the spots that are possibly free and those probably taken.

The communication module of the assistant agent extracts the lists  $FS_B$  and  $OS_B$  from each received message and forwards it to the decision module. The decision module updates both lists by aggregating the various received lists ( $FS_B$  and  $OS_B$ ) with its own ( $FS_A$  and  $OS_A$ ) by using the algorithm following bellow.

```

 $LP_A = \{\dots \langle pl_{i_1}, t_{i_1} \rangle, \dots\}$ 
 $LP_B = \{\dots \langle pl_{i_2}, t_{i_2} \rangle, \dots\}$ 
 $BL_A = \{\dots \langle pl_{i_3}, t_{i_3} \rangle, \dots\}$ 
 $BL_B = \{\dots \langle pl_{i_4}, t_{i_4} \rangle, \dots\}$ 
for all  $\langle pl_i, t_i \rangle \in LP_B$  do
  if  $(pl_i \in BL_A)$  then
    if  $(t_i^B \geq t_k^A)$  then
      eliminate( $pl_i$ ,  $BL_A$ )
      add( $pl_i, t_i^B, LP_A$ )
    end if
  else
    if  $(pl_i \in LP_A)$  then
      if  $(t_k^A < t_i^B)$  then
        update( $pl_i, t_i^B, LP_A$ )
      end if
    else
      add( $pl_i, t_i^B, LP_A$ )
    end if
  end if
end for
for all  $\langle pl_i, t_i \rangle \in BL_B$  do
  if  $(pl_i \in BL_A)$  then
    if  $(t_j^A < t_i^B)$  then
      update( $pl_i, t_i^B, BL_A$ )
    end if
  else
    if  $(pl_i \in LP_A)$  then
      if  $(t_j^A < t_i^B)$  then
        eliminate( $pl_i$ ,  $LP_A$ )
        add( $pl_i, t_i^B, BL_A$ )
      end if
    end if
  else
    add( $pl_i, t_i^B, BL_A$ )
  end if
end for

```

The idea is to browse each received list ( $FS_B$  and  $OS_B$ ) and update the local list ( $FS_A$  and  $OS_A$ ) with the date associated with the spots. If there are two conflicting informations, then the newest one is kept, since the last driver who has visited this spot

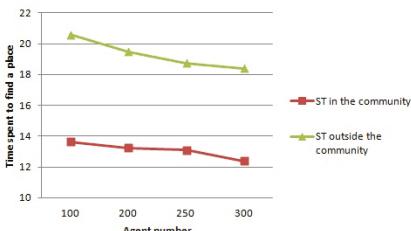
has the information that is most probably correct about its availability. After updating the two lists  $FS_A$  and  $OS_A$ , the decision module refers to the communication module, which is responsible for its dissemination to other neighbors in the community. This message processing is done in every state of the agent assistant, with the exception of the state 0.

When the driver is looking for a parking spot, he may request help from the system. The corresponding assistant agent updates its lists FS and OS from the received messages. Then, the decision module sends to the itinerary module the entire FS containing the list of spots known to be free. The itinerary module calculates the routes for each spot on this list and forwards the result to the decision module. Based on the selection criteria set by the driver, the decision module proposes a spot that meets the needs of the driver. Then it deletes the information corresponding to the proposed spot from its FS list. Finally, it sends the rest of the list and the OS list to the communication module which takes care of their distribution to the neighbors. The removal of the information about this spot will reduce its spread within the community. Thus, the assistant agent increases the driver's chances of finding the spot free. In addition, during the movement of the driver to the chosen spot, the assistant agent can suggest an alternative spot that best meets her needs. If the driver finds the spot the proposed by the system is occupied, then the corresponding agent updates the list BL and begins to look him for another spot.

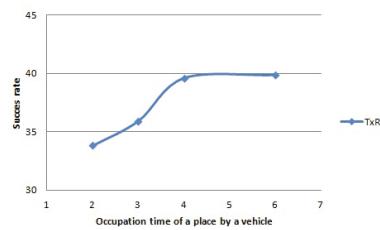
### 3 Experiments

For the validation, it is necessary to compare the effectiveness of the process of finding spots of drivers who use the system with drivers who don't. We have chosen the proximity to the current position as the decision criterion to choose a spot. The parameters that are selected for the system are as follows. First, the number of agents within and outside the community. Then, the time spent by an agent on a spot (OT, for occupation time). The third and final parameter is the lifetime of the information on the availability of a spot. We wish to verify the impact of this parameter of information dissemination on the results and to find the thresholds beyond which there is no point in maintaining this information.

To evaluate the different scenarios, we choose the following criteria. The first one is the success rate (SR, or effective use rate of the system) by the agents of the community which represents the ratio between the number of drivers who have found a parking via



**Fig. 2.** Relevance of the system

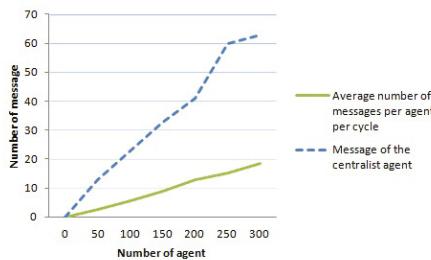


**Fig. 3.** Impact of spot rareness

the system by the total number drivers. The second criterion is the average spent time to find a spot per agent (ST). All time variables are expressed in number of execution cycles.

In the first series, we vary the number of agents in the community (NbA). This allows us to see the impact of system use on search time. In the second one, we study the impact of the rareness of spots on the success rate (SR). Finally, we compare the average number of exchanged messages per agent in our system and in centralized approach. In the graph shown in Fig. 2, we represent the average time spent to find a spot in and outside of the community according to the number of agents. The abscissa axis gives the number of agents that are taken into account. For instance, the value 100 means that 100 agents into the community spend on average 13.62 cycles to find a free spot whereas 100 agents outside community spend 20.57 cycles on average.

We can notice that, the more agents in the community we have, the less time they spend searching spots. This result is due to the fact that spots availability information is better propagated in the community, when the number of its members is important. Therefore, vehicles using the system spend less time to find a parking spot. Moreover, we can also note that the average time to find a spot (ST) for an agent of the community is much lower than that of an agent outside the community. The difference varies from one to seven cycles when all agents are in the community. For example, if there are 100 agents in the community, the average ST is equal to 13.62 cycles, whereas the average ST of an agent outside the community (i.e. 200 agents) is 19.48 cycles. According to these results, we can conclude that our proposal is useful and effective, especially when the community size is large enough. In a centralized approach, when the number of requests increases, the cost of communication rises and the quality of the service degrades because the response time increases while the information is volatile and can be obsolete meanwhile.



**Fig. 4.** Comparison of the number of messages per agent

In the next series of simulations, we fixed the number of spots in the network and varied the occupation time of a spot by a vehicle (OT). The Figure 3 illustrates the variation of the success rate according to the rareness of spots. For example the SR is 33.82 % when the OT is equal to 2 cycles. However this rate increases to 39,85 % when OT is 4 cycles. At the end, it stabilizes as the number of spots is limited. These results prove that more spots are rare more the system is useful, until a certain limit due in the limitation of the resources. Indeed, when a driver perceives several free spots, he does not really need help to find one. However, if they are rare the proposed system turns out to be very useful.

In Fig. 4, we report the number of messages handled by each agent in each cycle in our proposal, that we compare with a centralized solution. In the centralized approach, there are two messages exchanged with the central agent for each parking spot search (request and response). There is also a message informing that the chosen spot is taken. Finally, when the driver leaves a spot, there is a message informing about the spot release. This agent is unique, which limits the total number of messages but is a bottleneck. Fig. 4 shows that, even when the number of agents increases, the average number of messages that everyone should process remains reasonable (60 messages).

## 4 Related Work

In this section, we compare our proposal with the approaches of the literature. In the first subsection, we present previous solutions to the problem of parking spots, and in the second, we compare our cooperation model with the state of the art.

### 4.1 Application

Several studies such as [10] have identified the importance of better parking systems to improve the quality of life. Several works have proposed solutions to help drivers find a place as soon as possible either in car parks or in urban areas. In the first case, the e-car presented in [6] offers the driver to book online a parking space. SPARK [11] is an application of smart car that offers a new way of parking for large parking lots using communication in a VANET<sup>1</sup>. Another work in this context is presented in [4]. The authors employ inter-vehicular communication where a driver releasing a parking place disseminates information to her neighbors and assigns the resource to one of them. Thus, this solution assumes that the driver remains in the vicinity of the place and nearby vehicles that are interested until the allocation is made.

Other solutions have also been proposed for parking in an urban environment. San Francisco is the first city to adopt this type of intelligent system. Indeed, it has deployed SFpark<sup>2</sup>. This is an innovative project management of parking places in real time. The collection of real-time information is done through ground sensors. They are connected to a wireless network and thus allow to indicate at all times the presence or absence of cars on a place and relay this information to a centralized database. Drivers can be informed of the availability through various means such as electronic road signs placed along the streets, dynamic maps on the web site dedicated to mobile applications. In France, the first experiment took place in Lyon in early 2009, under the Predit<sup>3</sup>. This is a system for analyzing the presence of vehicles and calculating the length of the parking lot with sensors under the roadway every 2.5 meters that communicate via RFID. Information is updated every 10 seconds and disseminated to clients on their PDAs or via SMS. Another work in this context is presented in [3] where the authors propose an algorithm using a VANET and based on the dissemination of information regarding the state of urban

---

<sup>1</sup> Vehicle Ad Hoc Networks.

<sup>2</sup> <http://sfspark.org/>

<sup>3</sup> French Research Program and innovation in land transport <http://www.predit.fr>

parking fee. They produce such information and disseminate them to nearby vehicles. Then, the vehicles exchange information through inter-vehicular communication.

We note that the majority of solutions in the literature are centralized and use sensors to collect information, they require significant investments in contrast to the solution we propose in this paper. A less expensive option is explained in [14], where the authors have proposed to use an architecture called ParkNet based on a network of mobile sensors positioned in the vehicle that collects information about the availability of parking places in an urban area, along with the movement of the vehicle. They have proposed two architectures: a centralized architecture and a distributed architecture. In the distributed architecture, another vehicle-related sensors act as detectors of places and disseminators of information to other sensors.

## 4.2 Cooperation by Mutual Awareness

There are two main solutions to support mutual awareness in multi-agent systems: the creation of a dedicated channel and the distribution. The first solution is based on an architecture that intermediates between agents. This architecture might be such the MAS environment [12] or thematic channels [2]. As part of our application, the use of inter-vehicular communication to ensure the locality of interactions does not allow this type of approaches as there is no common channel to all agents.

The broadcast solution is the easiest to handle mutual awareness [79]. It has been used in [9] as support for a dynamic organization of autonomous vehicles and is similar to our approach. Mutual awareness is limited by the scope of communication and allows dynamic update of the representation of the world (groups, agents) through broadcasted messages. We have extended this result to a large number of agents and to the management of very volatile information thanks to the implementation of an epidemic spread of information [1]. Indeed, information from a vehicle is spreading in the network as it is valid (valid and not temporally questioned) by the movement of vehicles and their spontaneous interactions. The management of information quality that we propose has implications for the efficiency of the solution by mechanically limiting the dissemination of information to where it is useful.

## 5 Conclusion and Perspectives

In this paper, we propose a solution for the management of parking spots in an urban area. It is based on a multi-agent approach for the design of a community of drivers that interact to keep up to date information regarding the availability of parking spots. Communication between agents is supported by an inter-vehicular network with a radius of restricted broadcast, ensuring the consideration of local information. Our system works without prior information on the spots and no central storage of information. We have focused our validation on the average search time and showed a decrease regardless of the density of the vehicular network. The first perspective is to expand our testing protocol to take into account the particular hazards of data transmission inherent to this type of network and data traffic using the system Claire-Siti [3]. The second, is to study the definition of two architectures that re-centralize some of the processing. The objective is to compare the two architectures with the one presented here.

## References

1. Becker, C., Bauer, M., Hahner, J.: Usenet-on-the-fly: supporting locality of information in spontaneous networking environments. In: Proceedings of CSCW 2002 (2002)
2. Busetta, P., Merzi, M., Rossi, S., Legras, F.: Intra-role Coordination Using Group Communication: A Preliminary Report. In: Dignum, F.P.M. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 231–253. Springer, Heidelberg (2004)
3. Caliskan, M., Graupner, D., Mauve, M.: Decentralized discovery of free parking places. In: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, VANET 2006, pp. 30–39. ACM (2006)
4. Delot, T., Cénérario, N., Ilarri, S., Lecomte, S.: A cooperative reservation protocol for parking spaces in vehicular ad hoc networks. In: Mobility 2009 Proceedings of the 6th International Conference on Mobile Technology, Application and Systems, pp. 1–8. ACM (2009)
5. Gradinescu, V., Gorgorin, C., Diaconescu, R., Cristea, V., Iftode, L.: Adaptive traffic lights using car-to-car communication. In: IEEE Vehicular Technology Conference, pp. 21–25 (2007)
6. Hodel-Widmer, T.B., Cong, S.: Psos, parking space optimization service. In: The 4th Swiss Transport Research Conference (2004)
7. Kaminka, G., Pynadath, C., Tambe, M.: Monitoring teams by overhearing: A multi-agent plan-recognition approach. Journal of Artificial Intelligence Research 17, 83–135 (2002)
8. Lefauconnier, A., Gantelet, E.: La recherche d'une place de stationnement: Stratégies, Nuisances associées, Enjeux pour la gestion du stationnement en France. L'ingénierie du stationnement, bureau d'études et de conseil, Sareco (2005)
9. Legras, F., Tessier, C.: LOTTO: Group Formation by Overhearing in Large Teams. In: Dignum, F.P.M. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 254–270. Springer, Heidelberg (2004)
10. Nangeroni, C.: Stationnement intelligent: à la bonne place au bon moment. Ville, Rail et Transport 507, 30–33 (2010)
11. Rongxing, L., Xiaodong, L., Haojin, Z., Xuemin, S.: Spark: A new VANET-based smart parking scheme for large parking lots. In: IEEE INFOCOM, pp. 1413–1421 (2009)
12. Saunier, J., Balbo, F.: Regulated multi-party communications and context awareness through the environment. International Journal on Multi-Agent and Grid Systems 5, 75–91 (2009)
13. Scemama, G., Carles, O.: Claire-siti, public and road transport network management control: A unified approach. In: IEE Road Transport Information and Control Conference, pp. 11–18 (2004)
14. Suhas, M., Sanjit, K., Gruteser, M., Trappe, W.: Parknet: A mobile sensor network for harvesting real time vehicular parking information. In: Proceedings of the 2009 MobiHoc S3, pp. 25–28 (2009)
15. Yang, X., Liu, J., Zhao, F.: A vehicle-to-vehicle communication protocol for cooperative collision warning. In: Mobile and Ubiquitous Systems: Networking and Services, pp. 114–123 (2004)

# Modeling Smart Home Using the Paradigm of Nets within Nets

Yacine Kissoum<sup>1</sup>, Ramdane Maamri<sup>2</sup>, and Zaidi Sahnoun<sup>2</sup>

<sup>1</sup> Faculty of Sciences, Computer Science Department,  
Skikda University, Skikda, Algeria

<sup>2</sup> Lire Laboratory, Mentouri University, Constantine Algeria  
`{kissoumyacine, rmaamri, sahnounz}@yahoo.fr`

**Abstract.** Smart home is a sub branch of ambient intelligence technology. It was initially used to control environmental systems such as lighting and heating; but recently the use of smart technology has been strengthened and expanded such that the resulting technologies promise to revolutionarize daily human life by making people's surroundings flexible and adaptive. This paper proposes a technique that uses the paradigm of nets within nets to model the behavior of a mobile robot which is sensitive, adaptive and responsive to the presence of a human person and which is able to provide one or more homework.

**Keywords:** Ambient intelligence, Agents, Nets within Nets, Reference Net.

## 1 Introduction

Ambient Intelligence (AmI) [4] is a new technology which proposes to create an environment that is sensitive, adaptive and responsive to the presence of people, in order to create the desired atmosphere and functionality. It implies a seamless environment of computing and advanced networking technology that is aware of human presence, personalities, needs and which is capable of responding intelligently to spoken or gestured indications of desire, and even in engaging in intelligent dialogue.

A sub-branch of ambient intelligence is the design of smart home. Much of processes used for smart home design are ad-hoc. By time, all the gritty details of implementation are taken care with the original system description has pretty much been lost, causing a lack of design oversight and a surplus of one-time-only design artifacts [1].

To cope with the ad-hoc nature of smart home systems design process, it's required to use a suitable formal model that is able to handle smart home domain specific nature. Since Petri nets [2] have a long tradition to describe and analyze concurrent processes, they were excellent candidates. However Colored Petri Nets (CPN) [5] combines the best of classical Petri nets and high level programming languages [3], and are for that very popular. But, Köhler et al [9] have proposed a model that deals with complex systems in an elegant and intuitive manner without losing formal

accuracy. A model that is capable of expressing the different kind of mobility and which is built upon a formalism that has a formal semantics to support verification and execution. The proposed approach for modeling smart home proposed so far is based on such a model called Reference net [10]. Reference nets are based on the nets within nets paradigm that generalizes token to data types and even nets [12].

The rest of this paper is organized as follow: in section 2 we give a short introduction on the paradigm of reference net and show how this paradigm can be used to model the mobility concept in multi-agent special case. After a short description of the smart home case study in section 3, we present how the basic components of such a case study are modeled using the reference net paradigm. Section 5 gives a general insight on how verification a smart home can be conducted. Finally, the section 6 concludes with discussion about issues and future work.

## 2 Reference Nets and Mobility

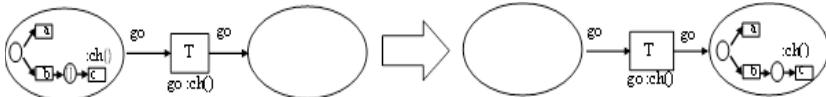
### 2.1 Reference Nets

The paradigm of nets within nets due to Valk [12] is based on the former work on task-flow nets. The paradigm formalizes the aspect that tokens of a Petri net can also be data types and even nets. Taking this in consideration it is possible to model hierarchical structures in an elegant way. A short introduction of the implementation of certain aspects of nets within nets called Reference net [11] will be given in subsequently.

Reference nets [10] are a graphical notation that is especially well suited for the description and execution of complex, concurrent processes. Reference nets extend black and colored Petri nets by means of net instances, nets as token objects, communication via synchronous channels, and different arc types. Beside this, they are quite similar to colored Petri nets as defined by Jensen [5]. Definitions of these extensions are given in [7,8].

### 2.2 Modeling Mobility

Intuitively and like black or colored token in ordinary Petri nets, the reference net paradigm admits place to host token nets. Look to Fig. 1. As it is stated before, in the reference net terminology, the net token is called object net and the surrounding net is called system net. The object net in the left place can be bound to the arc inscription  $go$ . So transition  $T$  is activated with this possible binding. In addition this transition is inscribed with a synchronous channel ( $go:ch()$ ). This means that for the object net  $go$  to become an actual binding of transition  $T$  a counterpart has to be found within  $go$ . This precondition is an enabled transition inscribed with the channel  $:ch()$ . This precondition satisfied by the transition  $c$  on the object net, so we can say that the synchronous firing of the object and system net can take place and leads us to the situation depicted in the right net of Fig. 1. The object net is moved to the right side place. Synchronously the marking of the object net changed and another firing of transition  $c$  is never possible.



**Fig. 1.** System and object net firing

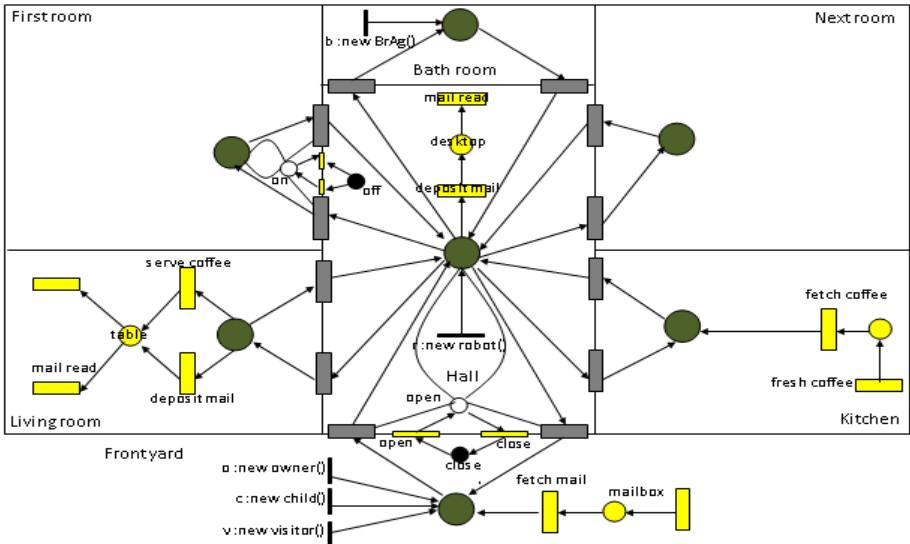
This example gives an idea how the interplay between object net and system net can be used to model mobile entities moving through a system net. Such a net offers or denies possibilities to move around, while the mobile object net moves at the right time by activating respectively, the transition that is inscribed with the precondition of the channel, and the move transition in the system net. Without the viewpoint of nets as tokens, the modeler would have to encode the mobile entity as a data structure for example. As a consequence, the inner actions of the mobile entity cannot be modeled directly, so they have to be lifted up to the system net, which seems quite unnatural. By using nets within nets we can investigate the concurrency of the system and the mobile entity in one model without losing the needed abstraction [9].

To investigate this modeling method, below we introduce the smart home case study. Then we show how we model this example by means of reference net paradigm.

### 3 The Smart Home Case Study

In this section we discuss how we modeled the smart home case study by means of reference nets paradigm. We lift the general insights of how mobility can be modeled to a special form of multi agent systems. That is, the modeling of human persons and a mobile robot (object nets) that cohabit a smart home composed of several rooms (system net) allows for an intuitive reproduction of real-world scenarios. So, the overall system will be designed according to MULAN [6] architecture, which is used to describe the natural hierarchies in an agent system. MULAN is implemented in RENEW [10], the IDE (Integrated Development Environment) and simulator for reference nets.

So, let us imagine a house with several rooms in which a mobile robot is designed as sensitive, adaptive and responsive to the presence of a human person and which is able to provide one or more services. Fig. 2 shows the system net of such case study. Places are locations (rooms) in or in front of the house: hall, living room, kitchen, first room, next room, bath room and the front yard. Transitions model possible movements between the rooms. Each room offers special services to the robot: it can fetch coffee in the kitchen, serve it in the living room, fetch mail in the front yard, and deposit it on the desk (in the hall) or on the table (in the living room) depending on the fact that the owner is out or at home. Service transitions are supplemented with additional information showing for instance if new mail has arrived, coffee is available, etc.



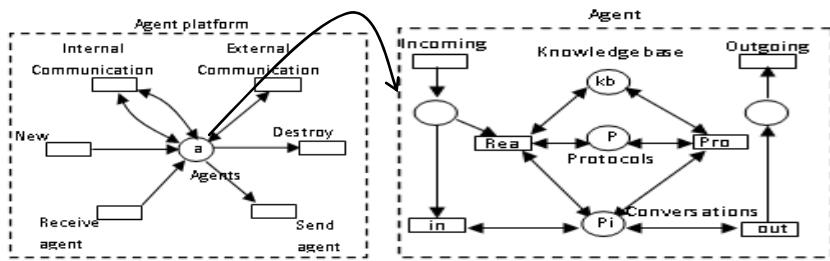
**Fig. 2.** The smart home system net

Rooms are also sensible to the human presence at home. Being sensible demands recognizing the user, learning or knowing her/his preferences, and the capability to exhibit empathy with or react to the user's mood and the prevailing situation. For this reason, we have modeled three kind of person that may enter such a house, namely: the owner, his child and a visitor. The door of the house and the lighting system of the first room are modeled to show another possibility of viewing special parts of the system: the state of the door (open/closed) and the light of the first room (on/off) are directly modeled. Also, a stationary agent can be associated to each room. This is visible by the thin-lined transition attached to the bathroom platform.

#### 4 Modeling the Basic Components

When modeling a complex system it is often undesirable to see the overall complexity at every stage of modeling and/or execution. Therefore the notion of a system view is introduced. Several views on an agent system are possible, for example the overall multi-agent agent systems, the set of platforms that host agents, the agent itself or simply its behavior. Using reference nets as a modeling paradigm allows for the direct use of system models at execution time. The difference between this proposal and a visualization tool that shows some activities of a program running in background is twofold [9]: (1) whereas a normal modeling process requires at least three stages to reach an executable program: (a) model the system, (b) implement the model and (c) write the visualization for the program, using the nets within nets paradigm, the modeling process concludes with a running system model. (2) The visualization of the system model at execution time is indeed the implementation of the system. This eliminates several potential sources of errors shifting from model to implementation to visualization in an ordinary software design process.

This model is filled with life by implementing the appropriate agents and defining the desired services for the platforms. Let us start with the structure of the platform. We recall here that places in the system net can host nets. So, by zooming into an arbitrary place, its structure becomes visible, shown in Fig. 3. The central place agents host all agents, which are currently on this platform. Each platform offers services to the agents laying on such platform, some of which are indicated in the figure. Agent can be created (transition *new*) or destroyed (transition *destroy*). Two agents of the same platform can communicate by the transition *Internal communication*, which binds two agents, the sender and the receiver, to pass one message over a synchronous channel. *External communication* binds one agent, since the other one is bound on a second platform somewhere in the system net. Also mobility facilities are provided on a platform: agents can leave the platform via the transition *send agent* or enter the platform via the transition *receive agent* [7].



**Fig. 3.** The structure of the platform and the agent system net (adapted from [9])

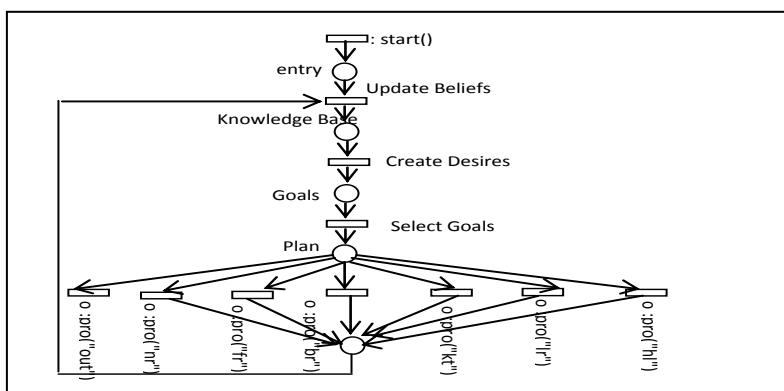
The agent net becomes visible if we zoom into place *Agents*. As mentioned above, we have associated to each room (platform) an agent, but here only the agent that lies on the bathroom platform is showed. Unlike such stationary agent, the other agents modeled are mobile. These agents are named respectively the owner, the child, the visitor and the robot. They have all the same agent's structure. They are encapsulated, since the only way of interaction is by message passing. Agents are intelligent, since they have access to a *Knowledge base* place. The behavior of the agents is described in terms of protocols, which are again nets. Protocols are located as templates on the place *Protocols*. Protocol templates can be instantiated, which happens either if a message arrives or if the agent undertakes an action pro-actively. An instantiated protocol is part of a conversation and lies in the place *Conversations*. Each agent can control an arbitrary number of such protocols, possesses however only one net, that represents its interface to the agent system and therewith its identity [8].

The central point of activity of a protocol-driven agent is the selection of protocols and therewith the commencement of conversations. The protocol selection can basically be performed pro-actively (dynamic autonomy) or reactively (deterministic autonomy). This distinction corresponds to the bilateral access to the place holding the protocols: *Protocols*. The only difference in enabling and occurrence of the transitions reactive and proactive is the arc from the place incoming messages to the transition *Reactive*. So it may only be enabled by incoming messages. Both the reaction to arriving messages and the commencement of a new conversation is influenced by the *knowledge base* place.

A selected and activated protocol is also called a conversation because it usually includes the exchange of messages with other agents. A conversation can however also run agent internal, therefore without message traffic. The transition *in* passes incoming messages to the corresponding conversation protocol in execution. If the sending of messages to other agents is required during the run of a conversation, these messages are passed from the protocol net over the transition *out* to the agent's main page and are handed over to the message transport mechanism by the transition *outgoing*. The communication between protocol net (conversation) and the agent's main net takes place via synchronous channels [9].

We modeled the owner as an agent having the structure of the right net in Fig. 3. Its knowledge base is a subnet consisting of a Knowledge base place, a Goals place and a Plan place, Fig. 4. The Knowledge base place corresponds to the owner's beliefs, which describes the knowledge of the environment and other agents. The Goals place corresponds to the owner's desires, which describe some desired final states and consists of a goal set. The desires of the owner represent its motivation and are the main source for the owner's actions. The desires may be associated with a value of a variable, a record structure, or a symbolic expression in some logic so that desires can be prioritized. The Plan place corresponds to the owner's intentions, which are a list of plans and describes the actions achieving the goal values of the owner. In simple cases the knowledge base, the goals and the plan places can be implemented for example as a subnets. Advanced implementations as the connections to an inference engine are also possible.

At home, the owner behaves fairly instinctive. Indeed, imagine the following scenario: as soon as he returns home, he checks for mails arrived during his absence. If it is the case, he will read them (in the hall). Otherwise, the owner will sit comfortably in the living room and zap some digital channels. Subsequently, he will move between rooms: it can, for example, moves to the kitchen and drink a soda, spend some minutes in the first room, take a shower in the bathroom or just sleep in the next room.

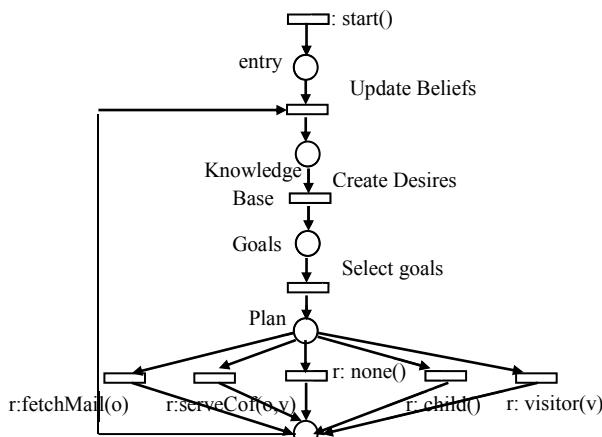


**Fig. 4.** The owner knowledge base net

The owner is endowed with a number of operations in order to act in such environment. It can perceive information and use it instantly or register it for later use. All actions the owner undertakes are driven by a set of goals. The goals of the owner are quite simple and depend on the room where he is. If, for example, he is in the hall he will update his beliefs (actual position and mail presence). Depending on such update, he will create one or more desired final states (take a shower or watch a TV program). Driven by such a final states (goals), the owner selects a plan to execute. This plan consists of a move to one of the rooms and some actions to perform in such a room. Actually, a plan is an instantiation and an execution of one of the owner's protocols. This cycle repeats until the owner decides to leave home. As soon as this later protocol is executed, the process is stopped until our human person reenters home.

Although the child and the visitor agents have the same structure as the owner agent, their knowledge base sub nets are different. For instance, the visitor cannot enter the house only if the owner is at home. Also, he is not authorized to enter first room, next room or read the mail. More precisely, to access home, the robot will ask the owner authorization for such visitor. Agents that lie on rooms will prohibit him access to such private locations. Finally, the robot will serve him coffee only after an explicit instruction (message) from the owner. As well as the visitor, the child has a knowledge base sub net that restrict his behavior. Unfortunately such sub nets cannot be deepened here any further.

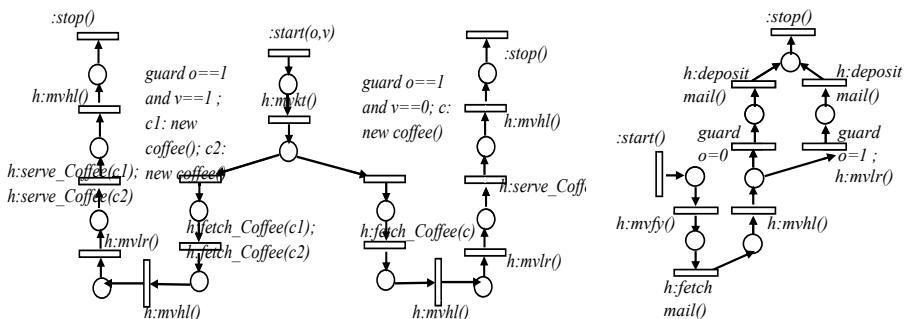
Before discussing the owner protocols, let us talk first about the robot modeling. It is again an agent having the structure of Fig. 3. Its knowledge base is also a subnet, Fig. 5. The actions that the robot should undertake depend closely on the fact that persons are inside or outside home. Beside this, if the robot hands are free it will check repetitively the human's presence at home. In the case of absence, it will wait for the arrival of mail and/or a visitor. As soon as the mail is available, the robot moves to the front yard, holds it, looks for the right destination (the desktop because it is home alone) and deposits it there.



**Fig. 5.** The agent knowledge base net

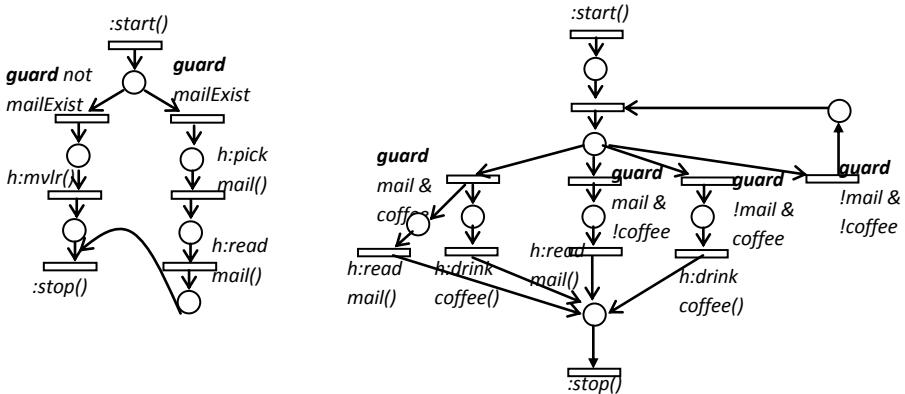
However, if the owner is at home, the robot will either serve him coffee or deliver him mail. In general, this means that the robot must take into account its own state and the state of the home. In our model the robot maintains the humans presence variables (the owner, the visitor and the child) and their positions (in which room), its own position and whether it actually carries some thing or not (transition *Update beliefs*). These beliefs contain information that the robot has collected in the past. It is clear that some of this information is volatile. A desktop position for example will never change, but the human location at a certain room might change after a while. In our model, the robot knows which beliefs unconditionally can be trusted and which are not trustable. The robot revises suspicious beliefs as soon as it has information about them from its percept update (transition *start*). Further, the robot must decide what it should do (transition *Create desires*). Driven by this goal, the robot selects a plan to execute (transition *Select goals*). This plan consists of set of moves between rooms and some actions to perform. Specifically, the robot can fetch mail and deposit it on desk or table depending on the owner presence at home (transition *fetchMail(o)*), serve coffee to owner and/or the visitor (transition *serveCof(o,v)*), treat the arrival of a visitor (transition *visitor(v)*) or the child (transition *child()*) or simply do nothing (transition *none()*).

The protocol nets that show the robot moves to deliver services are shown in Fig. 6. On the first one, the *:start* channel serves as a means to pass necessary parameters to the protocol. It is called on the agent main page by transition *proactive*. The channels *:mvkt*, *:mvhl*, *:mvfy* and *:mvlr* are responsible for moving the robot between rooms. The *:new coffee()* and *:fetchCoffee* transitions permit to prepare and pick up the coffee to deliver it in the living room (channel *:serveCoffee*). Here we use transitions with guard inscriptions. Such transition may only fire if all of its guard inscriptions evaluate to true. The *:fetchMail* permits to pick up the mail to deliver it either in the living room or in the hall (channel *:depositMail*). When a protocol is terminated, the transition inscribed with channel *:stop* is enabled. By calling of this channel the robot may delete the protocol or, more correctly, the protocol instance.



**Fig. 6.** The serve coffee and the deliver mail protocols

Let us back to then owner protocols. The net protocol executed when the owner lies in the hall is shown in the Fig. 7. Such protocol starts by checking mail presence. If mail exists, channels *:pickmail* and *:readmail* are activated subsequently, otherwise the transition inscribed by *:mvlr* is executed which means that the owner moves to the living room. Hence, the owner’s protocol associated to the living room is instantiated and executed. In such a room, the owner will continuously check the presence of mail and/or coffee. Once available, he will either read mail and/or drink coffee.



**Fig. 7.** The protocol nets associated to hall and to the living room

## 5 Towards Smart Home Verification

Using reference nets as a modeling paradigm concludes not only with a running system model but also supports verification. Which approach is well suited for such purpose? Several approaches try to lift verification styles designed for object oriented paradigm up to the agent context neglecting the special needs of multi agent systems namely: mobility, adaptation and cooperation. Distributed markings describe the mobility aspect of nets within nets in an algebraic way. They are especially suited for the formalization of agent groups in a distributed system. Adaptation is based on the reconfiguration of composed assumption/commitment Petri nets. Cooperation takes place in form of conversations, consisting of ordered assumption/commitment agent protocols [7].

Due to size limitations, any of these concepts are deepened here. Further publication will show in details how the verification of a smart home model using reference nets can be conducted.

## 6 Conclusion

In this paper we have presented the paradigm of “nets within nets” for the modeling of mobile agents in ambient intelligence environment. An important argument for using the paradigm of nets within nets is its strong expressiveness in capturing

mobility in an elegant and intuitive manner without losing formal accuracy. Indeed, using such a paradigm, the modeling process concludes with a running system model. Also, this paradigm is selected not only for mobility purposes in multi-agent systems, but mainly because it is built upon a formalism that has a formal semantics to support verification and execution. We think that such robust and easy-to-use tool reduces considerably the large initial effort in term of man-hours required mainly in constructing and validating smart home models. In fact, the reference net-based multi-agent system architecture MULAN(Multi Agent Nets) structures a multi-agent system in four layers, namely infrastructure, platform, agent and protocol. In this way the modeling process and the model validation are easier because of the smaller areas of such layers.

Additional services such as fire system, curtain control, climate control system, multimedia control system and so on, will be directly integrated in the design of the Petri net based multi agent system architecture MULAN.

## References

1. Nabih, A.K., Gomaa, M.M., Osman, H.S., Aly, G.M.: Simulation and control of smart home Using Petri Nets. *International Journal of Smart Home* 5(3) (July 2011)
2. Petri, C.A.: Communication with Automata, vol.1. Applied Data Research, Princeton (1966)
3. Barkaoui, K., Boucheneb, H., Hicheur, A.: Modelling and Analysis of Time-Constrained Flexible Workflows with Time Recursive ECATNets. In: Bruni, R., Wolf, K. (eds.) WS-FM 2008. LNCS, vol. 5387, pp. 19–36. Springer, Heidelberg (2009)
4. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.: Scenarios for ambient intelligence in 2010. Technical report, office for Official Publications of the European communities (2001)
5. Jensen, K.: Colored petri nets, basic methods, analysis methods and practical use. EATCS monographs on theoretical computer science, vol. 1. Springer (1992)
6. Duvigneau, M., Moldt, D., Rölke, H.: Concurrent Architecture for a Multi-agent Platform. In: Giunchiglia, F., Odell, J.J., Weiss, G. (eds.) AOSE 2002. LNCS, vol. 2585, pp. 59–72. Springer, Heidelberg (2003)
7. Köhler, M., Rölke, H.: Towards a unified approach for modeling and verification of multi-agent systems. In: Moldt, D. (ed.) Workshop on Modelling of Objects, Components, and Agents (MOCA 2001), pp. 85–104 (2001)
8. Köhler, M., Moldt, D., Rölke, H.: Modelling the Structure and Behaviour of Petri Net Agents. In: Colom, J.-M., Koutny, M. (eds.) ICATPN 2001. LNCS, vol. 2075, pp. 224–241. Springer, Heidelberg (2001)
9. Köhler, M., Moldt, D., Rölke, H.: Modelling Mobility and Mobile Agents Using Nets within Nets. In: van der Aalst, W.M.P., Best, E. (eds.) ICATPN 2003. LNCS, vol. 2679, pp. 121–139. Springer, Heidelberg (2003)
10. Kummer, O.: Simulating synchronous channels and net instances. In: Desel, J., Kemper, P., Kindler, E., Oberweis, A. (eds.) Workshop Algorithmen und Werkzeuge für Petrinetze, pp. 73–78 (1998)
11. Kummer, O., Wienberg, F.: Reference net workshop (Renew), <http://www.renew.de>
12. Valk, R.: Petri Nets as Token Objects: An Introduction to Elementary Object Nets. In: Desel, J., Silva, M. (eds.) ICATPN 1998. LNCS, vol. 1420, pp. 1–25. Springer, Heidelberg (1998)

# A Formal Model for Constructing Semantic Expansions of the Search Requests about the Achievements and Failures

Vladimir A. Fomichov and Anton V. Kirillov

Department of Innovations and Business in the Sphere of Informational Technologies,  
Faculty of Business Informatics, National Research University Higher School of Economics,

Kirpichnaya str. 33, 105187 Moscow, Russia

[vfomichov@hse.ru](mailto:vfomichov@hse.ru), [vfomichov@gmail.com](mailto:vfomichov@gmail.com)

**Abstract.** The paper describes a new method of constructing semantic expansions of search requests about the achievements and failures of active systems (organizations, people) for improving the results of Web search. This method is based on the theory of K-representations (knowledge representations), proposed by V.A. Fomichov - a new theory of designing semantic-syntactic analysers of natural language texts with the broad use of formal means for representing input, intermediary, and output data. The method uses an original formal model of a goals base – a knowledge base containing the information about the goals of active systems. The stated approach is implemented with the help of the Web programming language Java: an experimental search system AOS (Aspect Oriented Search) has been developed and tested.

**Keywords:** semantic transformation of search request, requests about achievements and failures, theory of K-representations, SK-languages, goals base, semantic transformation pattern, AOS search system.

## 1 Introduction

Every day the amount of information stored on the Internet is considerably increased. The format of presented information is heterogeneous, and its is unstructured; most often, the information is expressed by means of natural language (NL) – English, Russian, etc. A number of different approaches to recognizing a syntactic correspondence of a document to a search request have been proposed: vector space model, the ranking functions Okapi BM25 and BM25F (taking into account the various weight factors of the words from a document), and other. These approaches solve the problem of syntactic search, but a semantic correspondence of the found documents to the search request is not considered.

In order to solve this problem, several language systems describing semantic components of the documents have been developed, first of all, RDF, RDFS, OWL. Semantic description of a document provides the possibility to more exactly recognize

its content and respectively the relevancy as concerns a search request. Since meta-information most often is inaccessible, the focus of the methods of finding the document relevance has shifted to the semantic analysis of information stored in a natural language form.

During last years, many systems based on semantic analysis of the contents of requests and documents have been developed, in particular, SemSearch [1], Semantic Crystal [2], AquaLog [3]. However, there are such classes of search requests that the available scientific literature proposes no approach to their semantic processing. In particular, this applies to the requests about the achievements and failures.

For example, suppose that a businessman would like to get a certain information about the company X in order to consider the possibility of starting a collaboration with this company. In this connection, the questions about the achievements of the company during last year would be quite natural. For instance, the user may ask the questions "What achievements did the company X have last year?" or "What failures did the company X have last year?".

The problem is that, very likely, no one document being relevant to the first or second request includes the word "achievement" or "failure" respectively: such documents usually describe concrete facts, e.g., the launch of a new device by the company X, the increase (or diminishment) of benefit, etc.

This paper proposes a solution improving the work of traditional syntactic search systems by means of semantic analysis and expanding the natural language input requests about achievements and failures. Taking into account the calculating power of the biggest existing systems fulfilling the key words based search, it is proposed to shift the focus from the detailed semantic analysis and indexation of the content of electronic documents to the analysis of the inputted search requests and generation of a set of semantically expanded requests that will be transmitted to a syntactic search system. The results of the search corresponding to each request from this semantically expanded set will be analysed and compared with the aim of increasing semantic relevancy of the search results.

The central idea of our approach is the use of a special goals base for constructing semantic expansions of the search requests. This base contains the formal expressions indicating the achievements or failures of concrete active systems (on the example of organizations – the companies, etc.). A mathematical model of a goals base is proposed, this model uses the expressive mechanisms of SK-languages (standard knowledge languages) determined by the theory of K-representations (knowledge representations), stated, in particular, in [4]. This paper continues the line of the work [5].

## **2 A Method of Searching for Information of Interest**

Let's consider a method of looking for the information being of interest for the user under the framework of the proposed approach. A generalized algorithm consists of five main steps, two of them are unique for the considered type of questions.

**Step 1.** The inputted search request is analysed for finding its type. It is necessary to distinguish the primary and secondary objects of interest of the search request W. Suppose that the request W = “What achievements did the company Intel have in the year 2010?”. Then the primary object of interest is W1 = “achievements”, and secondary object of interest is W2 = “the company Intel”. The object W1 enables us to classify the search request W as an element of the class of questions about achieving a goal.

**Step 2.** After finding the type of the request, it is possible to go to creating a set of secondary search requests generated by the request W, that is, to forming a semantic expansion of the inputted request. The construction of the semantically expanded set of requests is being fulfilled with the help of a knowledge base containing the information needed for forming new requests.

**Step 3.** As soon as the expanded set of requests has been formed, it is transmitted to the traditional search system, the latter returns a set of documents which syntactically correspond to the generated requests. Dependent on the preferences of the user, i.e. dependent of the user’s behavior and selection of certain results of the search, the weights of the substitutions and the order of generating the requests (during the previous step) will be calculated.

**Step 4.** The documents received from the search system are analysed and filtered with the help of a knowledge base (in order to calculate the number of occurrences of the indicators of interest in the document) and with the help of the indicators of documents’ syntactic relevancy (the documents having the values of these indicators below a certain border will be excluded as non-relevant). The indicators will be understood as such natural language expressions that their occurrence in the text allows for judging about the correspondence of the document to the initial search request. First of all, the documents with the big amount of duplications will be considered. The reason is as follows: if a document more often occurs in the results of search proceeding from different requests, this document contains more indicators and, hence, contains more information corresponding to the initial request.

**Step 5.** The analysed and filtered documents are then returned to the user.

### 3 Theory of K-Representations as a Methodological Basis of the Study

Below we propose an original method of constructing semantic expansions of the search requests concerning the achievement of certain goals by the companies or persons. The method is underpinned by a formal model of a goals base. The main ideas for building this model were provided by the theory of K-representations (knowledge representations). It is a new theory of designing semantic-syntactic analysers of NL-texts with the broad use of formal means for representing input, intermediary, and output data [4]. This theory also contributes to the development of logic-informational foundations of (a) Semantic Web of a new generation, (b) E-commerce, and (c) multi-agent systems theory (agent communication languages). The structure of this theory is as follows.

The *first basic constituent* of the theory of K-representations is the theory of SK-languages (standard knowledge languages). The kernel of the theory of SK-languages is a mathematical model describing a system of such 10 partial operations on structured meanings (SMs) of natural language texts (NL-texts) that, using primitive conceptual items as "blocks", we are able to build SMs of arbitrary NL-texts (including articles, textbooks, etc.) and arbitrary pieces of knowledge about the world. The analysis of the scientific literature shows that today the class of SK-languages opens the broadest prospects for representing SMs of NL-texts in a formal way.

The *second basic constituent* of the theory of K-representations is a broadly applicable mathematical model of a linguistic database [4]. The *third basic constituent* of the theory of K-representations is several complex, strongly structured algorithms carrying out semantic-syntactic analysis of texts from some practically interesting sublanguages of NL. The algorithm *SemSynt1* transforms a NL-text in its semantic representation being a K-representation [4]. The input texts can be from the English, German, and Russian languages.

## 4 A Formal Model of a Knowledge Base for Representing Goals

Let us say about the questions about achieving a goal in case of the queries where one asks about information reflecting the results of functioning of an object, a system. In other words, these are *the questions about the achievements and failures*.

The success of functioning (or existing) of an object or a system is determined by achieving the formulated goals. By a goal of a company we'll understand the final desirable result that is set in the process of planning. An example of the questions about achieving a goal is as follows: "What failures did the company Sun experience in the year 2010?".

For fulfilling a detailed analysis of questions about achieving a goal, we've selected, studied, and divided into several groups the goals associated with the activity of the enterprises. The examples of such goals are as follows: "The launch of a new product", "Starting a new office by a company", "The increase of benefit", "The joining of a company".

The data of the kind should be stored in a special knowledge base, it will be called a *goals base*. This base is used for the generation of NL expressions showing the availability in the documents of the information about success or failure. The goals base model is constructed with the help of the theory of K-representations.

**Definition 1.** A *knowledge signature* is an ordered tuple *Ksign* of the form

$$(X, Concepts, Individuals, Bin-rel, Events-concepts, Numbers, ref), \quad (1)$$

where *X* is a countable set of symbols, *Concepts*, *Individuals*, and *Bin-rel* are non-intersecting subsets of *X*, the set *Individuals* includes the symbol *#Object-of-interest*, *Events-concepts* is a subset of *Concepts*, *Numbers* is a subset of *Individuals*, *ref* is a distinguished element of *X* not belonging to the sets *Concepts*, *Individuals*, *Bin-rel*, the set *Bin-rel* includes, in particular, the elements *Agent1* and *Role2*.

The set  $X$  will be called the *primary informational universe*, the elements of the sets *Events-concepts* and *Numbers* will be respectively called the *event-qualifying concepts* and the *rational number representations*, the element *ref* will be called the *referential quantifier*.

**Example.** The sets *Concepts* and *Individuals* may include respectively the elements *firm1*, *person* and *IBM*, *A.Turing*. The set *Bin-rel* may include the elements *Location*, *Director*, *Time*, *Agent1*, *Role2*, the subset *Events-concepts* may contain the elements *benefit-increase*, *joining-comp*, and the subset *Numbers* may include the elements -1 and 12.3.

The elements of the sets *Concepts* and *Individuals* are to be interpreted respectively as the designations of the considered notions (or concepts) and of concrete objects (the company IBM, etc.). The subset *Events-concepts* contains the semantic items qualifying the events (joining a company, opening of an office, etc.). The set *Bin-rel* contains the designations of binary relations and functions with one argument. The elements of the set *Numbers* are to be interpreted as the designations of rational numbers. The informational item *ref* denotes the meaning of the word combination “a certain” (*ref = certain* in the examples below).

**Definition 2.** Let  $Ksign$  be an arbitrary knowledge signature of the form (1). Then  $Lang1(Ksign)$  is the union of the set *Individuals* and of the set consisting of all strings of the form *ref cpt*, where *cpt* belongs to the set *Concepts*.

**Example.** The language  $Lang1(Ksign)$  may include the expressions *IBM* and *certain firm1*.

**Definition 3.** Let  $Ksign$  be an arbitrary knowledge signature of the form (1). Then  $Lang2(Ksign)$  is the set consisting of all strings of the form

$$\text{ref cpt } * (\text{rel}_1, d_1) \dots (\text{rel}_n, d_n),$$

where *ref* is the referential quantifier, *cpt* belongs to the set *Concepts*, *n* is not less than 1, for  $k = 1, \dots, n$ , the element *rel<sub>k</sub>* belongs to *Bin-rel* and *d<sub>k</sub>* belongs to  $Lang1(Ksign)$ .

**Example.** There is such knowledge signature  $Ksign$  of the form (1) that  $Lang2(Ksign)$  includes the expression *certain firm1 \* (Location, France)*.

**Definition 4.** For arbitrary knowledge signature  $Ksign$  the language  $Lobjects(Ksign)$  is the union of  $Lang1(Ksign)$  and  $Lang2(Ksign)$ .

**Definition 5.** Let  $Ksign$  be an arbitrary knowledge signature of the form (1). Then  $Lgoals(Ksign)$  be the finite set consisting of all expressions of the form

$$\langle \text{concept } * (\text{Agent1}, d1)(\text{Role2}, d2), \text{estimation} \rangle,$$

where *concept* belongs to the set *Events-concepts*, *d1* and *d2* are different elements of the language  $Lobjects(Ksign)$ , where either *d1* or *d2* is the symbol *#Object-of-interest*, and *estimation* is an element of the set *Numbers* denoting a rational number from -1 to 1 being distinct from 0.

**Example.** The set  $Lgoals(Ksign)$  may include the string

$$<\text{joining-comp} * (\text{Agent1}, \#\text{Object-of-interest})(\text{Role 2, certain firm1}), 1>. \quad (2)$$

The idea of considering the formal expressions of the kind stems from the theory of K-representations.

**Definition 6.** An ordered tuple  $Gbase$  of the form  $(Ksign, Goals)$  is called a *goals base* iff  $Ksign$  is an arbitrary knowledge signature, and  $Goals$  is a finite subset of the language  $Lgoals(Ksign)$ .

The expressions of the form (2) enable us to represent the facts about the achievements or failures of the companies and persons. However, it would be insufficient to proceed from only these data for building semantic expansions of search requests. It is necessary to transform such facts into the word combinations that most often occur in electronic documents. The transformation must take into account the morphology and syntax of the considered language in order most flexibly transform the facts into NL-expressions. That is why let's introduce the concept of a semantic transformation pattern.

**Definition 7.** Let  $Gbase = (Ksign, Goals)$  be an arbitrary goals base, and  $Morph-values$  be a finite set of symbols interpreted as the values of various morphological properties (noun, verb, singular, plural, passive-voice, active-voice, etc.). Then a *semantic transformation pattern* generated by the goals base  $Gbase$  and the set  $Morph-values$  is an arbitrary five-tuple of the form

$$(\text{sem-pattern}, X, Y, Z, \text{prop-chain}), \quad (3)$$

where  $\text{sem-pattern}$  is an element of  $Goals$ , the triple  $(X, Y, Z)$  is an arbitrary permutation composed by the symbols  $\#\text{A}\#$ ,  $\#\text{Pred}\#$ ,  $\#\text{B}\#$ , and  $\text{prop-chain}$  is a string of the form  $v_1 * v_2 * \dots * v_k$  where  $k$  is not less than 1, and  $v_1, \dots, v_k$  are the elements of the set  $Morph-values$ .

**Example.** Let  $\text{sem-pattern}$  be the string

$$<\text{joining-comp} * (\text{Agent1}, \#\text{Object-of-interest})(\text{Role 2, certain firm1}), 1>.$$

Then the five-tuple  $(\text{sem-pattern}, \#\text{A}\#, \#\text{Pred}\#, \#\text{B}\#, \text{verb} * \text{Past-Simple})$  is one of the possible semantic transformation patterns.

## 5 A Method of Transforming the Requests about the Achievements and Failures into an Expanded Form

The descriptions of the achievements and failures (let's call them the facts) are stored in a goals base and are used for the generation of word combinations being the indicators of the document fragments mentioning these achievements or failures. Consider in more detail a method of transforming a fact into a word combination – a lexical indicator of this fact.

The construction is being fulfilled with the help of the transformation rules being unique for each class of facts. A transformation rule indicates the order of the words

in the word combination and the forms of combinations. These combinations will enable a traditional search system realizing the search on key words to find all documents mentioning the relevant facts. The collection of the documents returned by a search system will be analysed from the standpoint of calculating the quantity of occurrences of various combinations – indicators, that is, the indicators of a reference in the document to a fact.

The method stated below postulates the existence of a linguistic knowledge base (or database) establishing a correspondence between the informational (or semantic) items and the lexical expressions. In other words, let's suppose that there is such function *Lex-forms* with the domain being the set of informational items *Inf-items* that for each element *sem* from this set the value of the function *Lex-forms(sem)* is the set of all lexical expressions corresponding to the informational item *sem*. E.g., if *sem* = *firm1*, then *Lex-forms(sem)* = {*the firm, a firm, the company, a company*}.

### **A Method of Generating the Resulting Indicators**

- [1] As a result of processing a search request, the variable *Studied-object* receives the value being a designation of the secondary object of interest of the input request (for instance, *Studied-object := Microsoft*).
- [2] If the search request is about the achievements, the variable *Estim-orient* receives the value 1 else (if the request is about the failures) *Estim-orient* := -1.
- [3] For each semantic transformation pattern of the form (3), where the component *estimation* of the string *sem-pattern* has the same sign as the value of *Estim-orient*, the string *X Y Z* is constructed.
- [4] In the string *X Y Z*, the symbol *#A#* is replaced by arbitrary lexical item corresponding to the value of the variable *Studied-object*. That is, *#A#* is replaced by arbitrary lexical item from the set *Lex-forms(Studied-object)*.
- [5] Let *sem-pattern* be the string of the form *<event-concept \* (Agent1, d1)(Role2, d2), estimation>*. Then in the string obtained during Step 3, the symbol *#Pred#* is replaced by arbitrary lexical item *pred-word-form* from the set *Lex-forms(event-concept)*, where the lexical item *pred-word-form* should possess the values of morphological properties determined by the string *prop-chain*.
- [6] In the string obtained during Step 4, the symbol *#B#* is replaced by arbitrary lexical item corresponding to such semantic item from the set {*d1, d2*} that is distinct from the symbol *#Object-of-interest* in the considered semantic transformation pattern.

**Example 1.** Let *Studied-object := firm-Oracle*, *sem-pattern* be the string *<joining-comp \* (Agent1, #Object-of-interest)(Role 2, certain firm1), 1>*. Then the processing of the semantic transformation pattern (*sem-pattern, #A#, #Pred#, #B#, verb \* Past-Simple*) in accordance with the stated method may lead to the construction of the resulting indicator “*Oracle joined the company*”.

**Example 2.** Let the search request W= "What are the achievements of the company Oracle?". After finding the type of request and its object of interest it is possible to build an expanded request being a set of concrete requests.

The stated method considerably details the authors' ideas of processing the search requests about the achievements or failures stated in [5]. This method underpins the developed algorithm of constructing a set of transformed requests *ExtSet* for the questions about achievements and goals.

## 6 The Program Complex AOS Engine

The principal idea underpinning the architecture of a program complex realizing the proposed approach is the maximal availability and independence of its components and also the possibilities of using them independently. Let's consider the process of transforming and analysing a search request from the standpoint of the used logical components. This process can be described by the following scheme:

$$\begin{aligned} \text{Search request} &\Rightarrow \text{Subsystem-1} \Rightarrow \text{Data-structure-1} \Rightarrow \text{Subsystem-2} \\ &\Leftrightarrow \text{Data-structure-2} \Rightarrow \text{Subsystem-3} \Rightarrow \text{Data-structure-3} \Rightarrow \\ &\qquad\qquad\qquad \Leftrightarrow \text{Subsystem-4} \Rightarrow \text{Data-structure-4}, \end{aligned}$$

where Subsystem-1 is Analyser of requests, Data-structure-1 is the result of analysing the input request (its type and the objects of interest), Subsystem-2 is the component of expanding the requests, Data-structure-2 is the set of transformed requests, Subsystem-3 is the component of interacting with a search system (or systems), Data-structure-3 is the set of retrieved documents, Subsystem-4 is the component of ranging, Data-structure-4 is the set of retrieved relevant documents.

Subsystem-1 (Analyser of requests) interacts with the Stemming Component and with the vocabularies. Subsystem-2 (the component of expanding the requests) interacts with (a) a goals base, (b) a base of knowledge about the changes of sets (the management boards of companies, etc.), (c) a linguistic database (LDB), (d) an aspect-oriented knowledge base (AOKB), including the knowledge about the structure and functions of various objects and systems. Subsystem-4 (the component of ranging) interacts with LDB and AOKB.

The analyser of requests finds the type and the objects of interest of the processed request. For transforming the words into their stem-form, the component of stemming is used, it includes the logic and libraries for obtaining the normal forms of the words. The output of the analyser is an expanded form of the input request, this form includes the type of request, its object of interest, and possible additional objects of interest (e.g., a time interval).

The constructed semantic expansion of the processed search request (i.e., a set of generated requests) is transmitted to the component of interacting with syntactic search system (or several systems). This component fulfils a connection with a particular syntactic search system (the search is based on key words), transmits to it each generated search request, and aggregates the received documents.

The designed program complex is called AOS Engine (Aspect-Oriented Search Engine) and consists of three principal subsystems: AOS Engine System (Aspect-Oriented Search System), LDB (Linguistic Database), and AOKB (Aspect-Oriented Knowledge Base), see <http://www.aosengine.ru>.

It should be mentioned that all subsystems interacting with databases and knowledge bases use the System of Object-Relational Mapping (ORM) and database connection pool (DBCP). The employment of ORM allows for diminishing the time for the cycle design – implementation, and the use of DBCP gives the possibility to reduce the time being necessary for creating new connections with a database, this diminishes the time for access to data.

Taking into account these requirements, the availability of various Web-components and several databases, the programming environment Java SE 6 was selected as a platform for the implementation of the elaborated architecture.

It should be noted that the code of the elaborated program complex AOS Engine was integrally tested, and all components and subsystems of the complex correctly interact. For testing the algorithms of finding the type and objects of interest of the input search requests, the method of white box in automatic regime was used with the help of the testing library JUnit. A set consisting of over 300 unique search requests of various types was formed. Most often, the quantity of retrieved documents containing the details obtained as a result of using the constructed knowledge bases exceeds 50%.

## 7 Conclusion

The stated formal approach to constructing semantic expansions of search requests is based on the ideas of the theory of K-representations. We considered above the questions concerning the achievement of certain goals or concerning the failures. Our approach is implemented with the help of the Web programming language Java: an experimental search system AOS (Aspect Oriented Search) has been developed and successfully tested.

## References

1. Uren, V.S., Lei, Y., Motta, E.: SemSearch: Refining Semantic Search. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 874–878. Springer, Heidelberg (2008)
2. Kaufmann, E., Bernstein, A.: How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users? In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 281–294. Springer, Heidelberg (2007)
3. Lopez, V., Uren, V., Motta, E., Pasin, M.: AquaLog: An Ontology-driven Question Answering System for Organizational Semantic Intranets. Web Semantics: Science, Services and Agents on the World Wide Web 5, 72–105 (2007)
4. Fomichov, V.A.: Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms. Springer (2010)
5. Fomichov, V.A., Kirillov, A.V.: Semantic Transformation of Search Requests for Improving the Results of Web Search. In: Pre-Conference Proceedings of the Focus Symposium on Intelligent Information Management Systems (August 2, 2011, Focus Symposium Chair: Jens Pohl) in Conjunction with InterSymp-2011, 23rd International Conference on Systems Research, Informatics and Cybernetics, Germany, Baden-Baden, August 1-5, pp. 37–43. Collaborative Agent Design Research Center, California Polytechnic State University, San Luis Obispo, CA (2011)

# Parallel CHC Algorithm for Solving Dynamic Traveling Salesman Problem Using Many-Core GPU

Patryk Filipiak and Piotr Lipiński

Institute of Computer Science,  
University of Wrocław, Wrocław, Poland  
`{patryk.filipiak, lipinski}@ii.uni.wroc.pl`

**Abstract.** This paper presents a massively parallel evolutionary algorithm with local search mechanism dedicated to dynamic optimization. Its application for solving Dynamic Traveling Salesman Problem (DTSP) is discussed. The algorithm is designed for many-core graphics processors with the Compute Unified Device Architecture (CUDA), which is a parallel computing architecture for nVidia graphics processors. Experiments on a number of benchmark DTSP problems confirmed the efficiency of the algorithm and the parallel computing model designed.

## 1 Introduction

Evolutionary computation is being applied successfully to address numerous recent dynamic optimization problems (DOPs), i.e. those including time-dependent variations of the objective function. Naturally, DOPs are tailored to model crucial real-life processes that scarcely remain fixed or well-defined, e.g. vehicle routing, load balancing, funds investment, etc. Such practical DOPs are frequently dealt with Evolutionary Algorithms [3][4][10][11] or Artificial Neural Networks [12].

Traveling Salesman Problem (TSP) is the commonly known  $NP$ -hard combinatorial problem frequently used for benchmarking various optimization techniques. Its dynamic version (DTSP) [7] is even harder to solve, since it introduces time-varying environmental changes that can for instance result in turning a nearly optimal solution at the moment into a far from optimal one few seconds later.

Lately, a modification of CHC algorithm [2] was suggested for solving DTSP in [14]. Not only the original CHC handles many static optimization problems [16], yet additionally it can deliver promising results in DOPs after slight modifications emphasising its ability to maintain diversity within a population [14].

It is worth noticing, that diversified population is a key factor in any dynamic optimization problem. Otherwise, an algorithm would most probably converge to some optimum at given time step and lost the ability of tracking changes in the environment due to stagnation of individuals [1].

Since evolutionary computations are based on biological processes that usually take place simultaneously, it is tempting to introduce parallelism into algorithms solving DOPs aiming to speed them up. A use of Compute Unified Device Architecture (CUDA) [6] appears particularly suitable for this purpose. A number of CUDA-based approaches

were proposed for static optimization problems [8,9]. In this paper we focus on dynamic optimization where an effective many-core parallelization is even more desired due to speed of changes taking place during the time of computation.

## 2 Dynamic Traveling Salesman Problem

Above many equivalent definitions, a classical TSP can be simply described as the problem of finding Hamiltonian cycle in a complete weighted graph. Let  $G = (U, V)$  be such graph with  $U$  representing the set of edges and  $V$  – the set of  $n > 0$  vertices. Additionally, for each pair  $(v_1, v_2) \in U$  where  $v_1, v_2 \in V$ , let  $\delta(v_1, v_2) > 0$  encode the distance from vertex  $v_1$  to  $v_2$ . If  $(v_1, v_2) = (v_2, v_1)$  for each  $(v_1, v_2) \in U$ , a TSP is called *symmetric*, otherwise – *asymmetric*. The aim is to find a shortest path for visiting all  $n$  vertices precisely once, i.e. a permutation  $\sigma : \{1, \dots, n\} \longrightarrow \{1, \dots, n\}$  that minimizes

$$\sum_{i=1}^{n-1} \delta(v_{\sigma(i)}, v_{\sigma(i+1)}) + \delta(v_{\sigma(n)}, v_{\sigma(1)}), \quad v_1, v_2, \dots, v_n \in V. \quad (1)$$

In DTSP [7], the problem described above is considered in time domain including non-deterministic environmental changes, particularly adding new vertices or deleting existing ones and modifying values of  $\delta(\cdot, \cdot)$ .

## 3 Compute Unified Device Architecture

In the Compute Unified Device Architecture (CUDA), the computing platform consists of two parts: a sequential platform with a standard processor and a parallel platform with a many-core graphics processor. The sequential platform executes the single-threaded mainstream of a program and invokes some multi-threaded subprograms on the parallel platform. The parallel platform consists of a number of multi-threaded streaming multiprocessors that execute threads of an invoked subprogram in parallel. Multi-threaded streaming multiprocessors operate in the Single-Instruction Multiple-Thread (SIMT) architecture that allows a further parallelization by running a number of threads concurrently on the same multiprocessor.

When invoking a multi-threaded subprogram, the computing platform splits all its threads into a number of blocks and assigns them to multiprocessors, so that each multiprocessor has one or more blocks of threads to execute. Multiprocessors divide threads from each block into so-called warps of 32 threads and process them consecutively. Multiprocessors start each thread from the same warp at the same time, but may further desynchronise them due to some conditional instructions, and finish the execution of a warp when all threads of the warp terminate. When all threads from the same warp execute the same instruction, they are executed concurrently – otherwise, some threads must wait. Therefore, the full efficiency of the computing platform may be obtained when all threads execute the same instruction in the same time, which means that conditional instructions significantly increase the computing time.

**Table 1.** Hardware platform specification (nVidia GeForce GTX 580)

number of multithreaded streaming multiprocessors	16
number of logical cores per multiprocessor	32
total number of logical cores	512
number of registers per multiprocessor	16384
maximum number of threads per block	512
number of threads per warp	32
shared memory per multiprocessor	64 kB
constant memory 64	kB
local memory per thread	16 kB
maximum number of active blocks per multiprocessor	8
maximum number of active warps per multiprocessor	32
maximum number of active threads per multiprocessor	1024

Experiments reported in this paper were performed on a computing platform with the Intel Core i7 950 processor and nVidia GeForce GTX 580 graphics card, containing one many-core graphics processor with 16 multiprocessors. Details of the hardware platform specification are presented in Table I.

In experiments, random numbers were generated in multiple thread independently using the cuRAND library.

## 4 Algorithm

The proposed algorithm is an extended CUDA-based parallel incarnation of CHC dedicated to DTSP.

Assume that for all time steps  $t \in \mathbb{N}$ ,  $P_t$  is the current population of  $N \in \mathbb{N}$  individuals represented by the sequences of  $L \in \mathbb{N}$  alleles. Each of these alleles is a positive integer encoding a city (or vertex) on the Traveling Salesman path.

Since DTSP is one of the dynamic optimization problems, the evolutionary approach to this issue requires an ability to maintain diversity within a population. CHC assures that by two mechanisms used together. First, it allows for mating only those individuals that are *sufficiently distant* according to given metrics. Second, it re-initializes a population repeatedly preserving only one individual (with highest fitness) in the unmodified form.

For a given individual  $p = (p^1, p^2, \dots, p^L) \in P_t$ , let  $\varphi_p(\cdot)$  be the function determining *next vertex* on the Traveling Salesman path as follows

$$\varphi_p(i) = \begin{cases} p^{i+1}, & i + 1 \leq L \\ p^1, & \text{otherwise} \end{cases}, \quad i = 1, \dots, L. \quad (2)$$

Thus, the mutual distance  $\Delta(\cdot, \cdot)$  within a pair of individuals  $(l, r) \in \mathcal{P}_t = P_t \times P_t$  can be computed using the following formula

$$\Delta(l, r) = \#\{1 \leq i \leq L; \quad \varphi_l(i) = \varphi_r(i)\}. \quad (3)$$

Let  $0 \leq d \leq L$  be the mating distance threshold (initially set to some fixed value  $d_{start}$ ). Original CHC uses binary representation of chromosomes and measures their

mutual Hamming distance. In the proposed algorithm,  $\Delta(\cdot, \cdot)$  function defined above is applied instead due to permutation representation. Regardless which metrics is currently used, only those individuals with the mutual distance exceeding threshold  $d$  are mated. Hence, population remains diversified by avoiding incest.

In addition to the selection mechanism described above, the re-initialization of a population is performed after each environmental change. During this process, best individual  $p_{best} \in P_t$  (i.e. the one with highest fitness) is selected. Then, the rest of a population is replaced with clones of  $p_{best}$ . However, each clone is perturbed independently by swapping  $\lfloor(r \cdot L)/2\rfloor$  alleles, where  $0 < r < 1$  is the predefined divergence rate.

A pseudo-code of the main loop is presented in Algorithm 1. Parallelism of the proposed algorithm is based on two GPU kernels, namely *ReinitializationKernel* and *EvolutionaryKernel*. The first of them is presented in Algorithm 2. It is executed in one block containing  $N$  threads, where  $N$  is the population size. At the beginning,  $p_{best}$  is picked in parallel using tournament. After that, each thread creates one local copy  $p$  of  $p_{best}$  then perturbs it relatively to divergence rate  $r$  as it was described above. Eventually,  $p$  is stored in global memory.

---

**Algorithm 1.** Modified CHC algorithm with parameters:  $r$  – divergence rate ( $0 < r < 1$ ),  $L$  – chromosome length ( $L \in \mathbb{N}$ ),  $d_{start}$  – initial difference threshold ( $0 \leq d_{start} \leq L$ )

---

```

 $d = d_{start}$ 
 $P_1 = \text{InitPopulation}()$ 
Evaluate( $P_1$ )
while not termination condition do
    if the function has changed then
         $P_t = \text{ReinitializeKernel}(P_t, r)$ 
    end if
     $P_{t+1} = \text{EvolutionaryKernel}(P_t, d)$ 
    if  $P_{t+1} = P_t$  then
         $d = \max(0, d - 1)$ 
    else
         $d = d_{start}$ 
    end if
end while

```

---

Algorithm 3 presents the second kernel, where a single iteration of evolutionary process takes place. It is executed in  $\lfloor N/2 \rfloor$  blocks, each of them running  $2 \times L$  threads. For all  $b_x \in \{0, 1, \dots, \lfloor N/2 \rfloor - 1\}$ ,  $t_x \in \{0, 1\}$  and  $t_y \in \{0, 1, \dots, L - 1\}$ , thread  $(t_x, t_y)$  in  $b_x$ -th block is responsible for computing  $t_y$ -th allele of  $t_x$ -th chromosome in  $b_x$ -th pair of individuals.

For each randomly selected pair of individuals  $(l, r) \in \mathcal{P}_t$  the mutual distance  $\Delta(l, r)$  is calculated in parallel. Later on, evolutionary operators are applied to all the pairs with  $\Delta(l, r) \geq d$ . First, OX crossover is performed. For this purpose, a parallel version of OX [5] is used. It is based on preparing a one-to-one mapping of each allele of

---

**Algorithm 2.** ReinitializeKernel( $P_t, r$ ) where:  $P_t$  – population at time step  $t > 1$ ,  $r$  – divergence rate ( $0 < r < 1$ ),  $L$  – chromosome length,  $N$  – population size ( $L, N \in \mathbb{N}$ )

---

```

 $p_{best} = \text{BestIndividual}(P_t)$  in parallel
for all  $p \in P_t \setminus \{p_{best}\}$  do in parallel
     $p = \text{LocalClone}(p_{best})$ 
    for  $i = 1 \rightarrow \lfloor(r \cdot L)/2\rfloor$  do
         $j = \text{RandomIndex}(\{1, \dots, L\} \setminus \{i\})$ 
         $p = \text{SwapAllels}(p, i \leftrightarrow j)$ 
    end for
    store  $p$  in global memory
end for

```

---

**Algorithm 3.** EvolutionaryKernel( $P_t, d$ ) where:  $P_t$  – population at time step  $t > 1$ ,  $d$  – difference threshold ( $d \geq 0$ ),  $N$  – population size,  $\delta$  – number of local search chunks ( $N, \delta \in \mathbb{N}$ )

---

```

 $\mathcal{P}_t = \text{RandomPairs}(P_t)$ 
for all  $(l, r) \in \mathcal{P}_t$  do in parallel
    if MutualDistance( $l, r$ )  $\geq d$  then
        for  $i = 1 \rightarrow N$  do in parallel
             $(\sigma_l(i), \sigma_r(i)) = \text{GetMapping(OX-crossover, } l, r, i)$ 
        end for
         $(l', r') = \text{MapAllels}(l, \sigma_l, r, \sigma_r)$ 
         $(l'', r'') = \text{RandomShiftOfAllels}(l', r')$ 
        for  $i = 1 \rightarrow \delta$  do in parallel
             $(l''_i, r''_i) = \text{GetChunk}(l'', r'', i)$ 
            LocalSearch2-opt( $l''_i, r''_i$ )
        end for
        Evaluate( $\{l'', r''\}$ ) in parallel
         $(l, r) = \text{Reduce}(\{l, l''\}, \{r, r''\})$ 
    end if
end for

```

---

parent chromosome onto its new location in offspring chromosome instead of explicit online modifications of chromosomes.

Unlike the original CHC, the proposed algorithm additionally performs a local search after crossing over. Application of the mechanism known as *2-opt* seems particularly appropriate for this purpose due to its effectiveness in solving TSP [5].

The idea behind 2-opt is as follows. Let  $p = (p^1, p^2, \dots, p^L) \in P_t$  be an individual encoding a TSP path. 2-opt struggles to improve  $p$  by finding such pair of indices  $1 \leq i < j \leq L$  that  $(p^1, p^2, \dots, p^{i-1}, p^i, p^j, p^{j-1}, \dots, p^{i+1}, p^{j+1}, p^{j+2}, \dots, p^L)$  outperforms the original  $p$  and replaces it.

In order to improve parallelism of 2-opt and, as a result, speed up computation, chromosomes are split into  $\delta \in \{1, 2, 3, 4\}$  sequences of consecutive alleles of the length  $\lfloor L/\delta \rfloor$ . For the remainder of this paper, the described variants of 2-opt will be referred to as 2-opt/ $\delta$  (note that 2-opt/1  $\equiv$  2-opt). This modification, however, results in decrease of effectiveness. To address this inconvenience, one very simple additional operator is

applied. Its role is to shift both alleles in every pair of chromosomes a random number  $\rho$  of loci, where  $0 < \rho < \lfloor L/\delta \rfloor$ . The parallel CHC algorithms using 2-opt/ $\delta$  with and without shifting will be referred to as PS-CHC + 2-opt/ $\delta$  and P-CHC + 2-opt/ $\delta$  respectively.

At the end of the evolutionary kernel, new individuals  $l''$  and  $r''$  are evaluated and compared with their parents  $l$  and  $r$ . Eventually, each child with higher fitness than its parent replaces it in the next generation.

## 5 Experiments

To our best knowledge, no completely reliable benchmark repository for DTSP was released so far. All the experiments presented in this paper were made against *ad hoc* modifications of static symmetric benchmark problems from TSPLIB [13]. The algorithm for preparing dynamic versions of TSP was inspired by the generator proposed in [17]. It is based on random re-locations of vertices at the end of iteration. In order to keep at least minimal control of these changes, all of them are stored in a memory and eventually restored one by one until the starting point is reached again. In other words, for any desired odd number of time steps  $T \gg 0$ , the first  $\lceil T/2 \rceil$  changes are generated randomly and then the remaining  $\lfloor T/2 \rfloor$  are their reversed copies like in a mirror reflection. Hence, for all  $0 < t \leq T/2$ , two states of the environment, i.e. those at time steps  $t$  and  $(T+1)-t$ , are equal.

The following testing procedure was applied. First, the three popular benchmark problems from TSPLIB were selected, namely *eil51*, *kroA100* and *kroB200*. All of them were transformed into dynamic problems containing  $T = 1001$  time steps satisfying the “mirror” rule described above. The obtained scenarios were stored and then replayed during each run of tested algorithms in the regular frequencies of 10, 20 and 30 generations per change (i.e. iterations of algorithm between each  $t$ -th and  $(t+1)$ -th time step).

A comparison of four algorithms will be presented, i.e. the original serial CHC and CHC with 2-opt local search added (CHC + 2-opt) as well as the proposed parallel versions, namely P-CHC + 2-opt/ $\delta$  and its variant enhanced with shift operator, PS-CHC + 2-opt/ $\delta$  (in both cases  $\delta \in \{1, 2, 3, 4\}$ ). Since the original CHC is rather rapid (in terms of time consumption) yet reacts more slowly to environmental changes (in terms of number of generations) when compared to the algorithms with local search, an additional variant of CHC with 10 times more generations per change (CHC x10) was also included in experiments.

For each algorithm, the following parameters were set: population size  $N = 100$ , divergence rate  $r = 0.5$  and initial distance threshold  $d_{start} = L/4$ , where  $L$  is the chromosome length (and so the number of vertices):  $L = 51$  in *eil51*,  $L = 100$  in *kroA100* and  $L = 200$  in *kroB200*.

In order to obtain the reference results, each of the tested DTSPs was split into  $T$  static sub-problems and solved independently with CHC + 2-opt using population of  $10N = 1000$  individuals during 100 generations. The obtained results are equal to the ones published in TSPLIB for instances with known answers (i.e. for  $t \in \{0, T-1\}$ ) and are assumed at least good approximations of correct results for  $0 < t < T-1$ .

**Table 2.** Averaged results of 10 runs against ad hoc dynamic version of benchmark problems: (a) eil51, (b) kroA100, (c) kroB200. Average and maximum deviations from the reference solution together with computation time are presented for problems with 10, 20 and 30 generations per environmental change.

(a) eil51

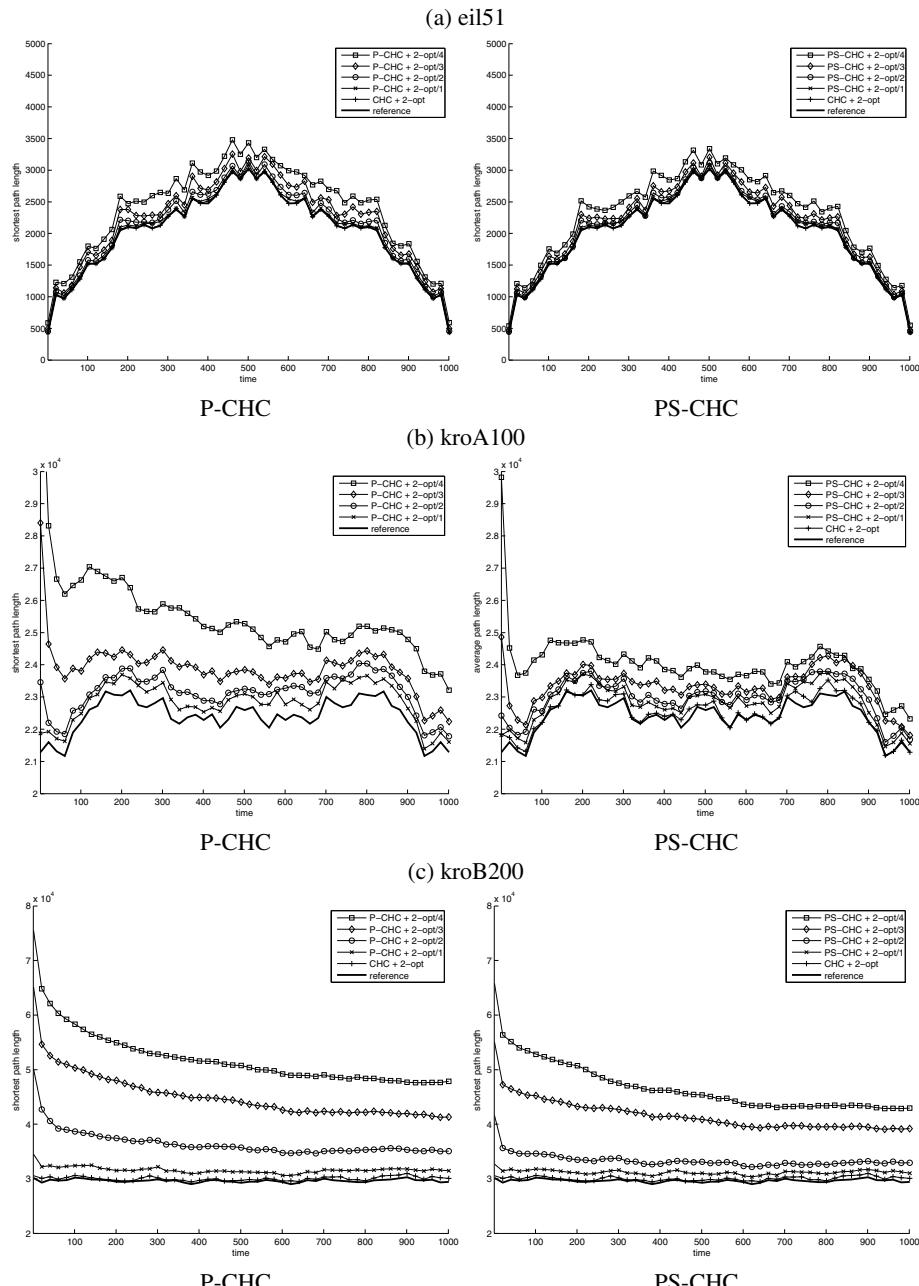
algorithm	10 gen./change			20 gen./change			30 gen./change					
	avg	dev	max dev	time	avg	dev	max dev	time	avg	dev	max dev	time
P-CHC + 2-opt/4	32.2%	56.8%	2.0s	19.4%	37.9%	2.8s	12.0%	30.9%	3.7s			
P-CHC + 2-opt/3	18.9%	28.9%	2.6s	10.1%	18.5%	3.6s	5.9%	11.9%	4.8s			
P-CHC + 2-opt/2	8.8%	14.9%	3.3s	4.4%	10.7%	5.0s	2.7%	9.6%	7.0s			
P-CHC + 2-opt/1	2.1%	8.6%	8.2s	1.3%	7.9%	14.8s	1.1%	7.4%	21.6s			
PS-CHC + 2-opt/4	25.2%	44.7%	2.0s	14.8%	28.3%	2.8s	9.2%	24.1%	3.7s			
PS-CHC + 2-opt/3	12.8%	20.2%	2.6s	6.8%	13.4%	3.6s	4.2%	11.2%	4.7s			
PS-CHC + 2-opt/2	5.4%	10.8%	3.3s	2.8%	9.5%	5.0s	1.9%	8.7%	6.9s			
PS-CHC + 2-opt/1	2.0%	8.4%	8.3s	1.3%	7.7%	14.8s	1.1%	7.3%	21.6s			
CHC + 2-opt	0.4%	2.6%	145.8s	0.2%	2.9%	291.2s	0.2%	1.9%	434.6s			
CHC×10	5.0%	85.6%	58.4s	3.4%	45.0%	111.9s	2.9%	17.2%	164.3s			
CHC	115.4%	192.2%	6.5s	71.9%	163.9%	12.8s	36.7%	150.6%	18.9s			

(b) kroA100

algorithm	10 gen./change			20 gen./change			30 gen./change					
	avg	dev	max dev	time	avg	dev	max dev	time	avg	dev	max dev	time
P-CHC + 2-opt/4	24.3%	82.5%	4.0s	13.6%	62.0%	6.7s	7.2%	51.8%	9.3s			
P-CHC + 2-opt/3	12.1%	49.3%	5.5s	6.5%	33.4%	9.5s	4.4%	24.0%	13.6s			
P-CHC + 2-opt/2	5.4%	17.5%	9.7s	3.3%	10.2%	18.0s	2.5%	6.0%	26.3s			
P-CHC + 2-opt/1	2.7%	4.7%	31.4s	2.0%	3.8%	61.5s	1.7%	3.2%	91.5s			
PS-CHC + 2-opt/4	11.9%	61.3%	4.0s	7.0%	40.1%	6.7s	3.8%	34.2%	9.2s			
PS-CHC + 2-opt/3	7.2%	30.7%	5.5s	4.1%	16.8%	9.5s	2.9%	12.9%	13.4s			
PS-CHC + 2-opt/2	3.8%	8.8%	9.4s	2.6%	5.3%	17.5s	2.0%	4.9%	25.5s			
PS-CHC + 2-opt/1	2.7%	4.3%	31.8s	1.9%	4.2%	62.7s	1.6%	3.2%	93.4s			
CHC + 2-opt	0.6%	3.3%	576.4s	0.33%	2.4%	1151.3s	0.2%	1.8%	1725.1s			
CHC×10	6.9%	322.3%	158.0s	5.3%	214.6%	310.9s	5.0%	162.0%	463.6s			
CHC	242.0%	530.7%	17.1s	122.4%	491.6%	33.7s	43.8%	451.3%	49.9s			

(c) kroB200

algorithm	10 gen./change			20 gen./change			30 gen./change					
	avg	dev	max dev	time	avg	dev	max dev	time	avg	dev	max dev	time
P-CHC + 2-opt/4	33.1%	88.2%	19.3s	21.5%	71.1%	36.3s	14.0%	58.0%	53.3s			
P-CHC + 2-opt/3	17.7%	57.6%	30.8s	12.3%	41.3%	59.3s	8.5%	34.7%	87.6s			
P-CHC + 2-opt/2	9.5%	26.4%	65.2s	6.8%	18.0%	128.0s	5.9%	14.4%	190.7s			
P-CHC + 2-opt/1	6.1%	10.3%	255.3s	4.8%	8.4%	530.9s	4.1%	7.9%	811.8s			
PS-CHC + 2-opt/4	19.8%	59.1%	19.1s	13.4%	46.5%	35.7s	10.0%	32.2%	52.3s			
PS-CHC + 2-opt/3	11.8%	32.3%	30.0s	7.4%	23.1%	57.5s	6.2%	19.6%	84.8s			
PS-CHC + 2-opt/2	6.8%	14.6%	63.8s	5.7%	11.1%	126.7s	4.7%	11.3%	191.2s			
PS-CHC + 2-opt/1	5.7%	10.5%	263.7s	4.5%	8.5%	557.6s	4.2%	7.8%	860.2s			
CHC + 2-opt	1.9%	7.1%	2308.5s	1.5%	3.9%	4605.5s	1.0%	4.5%	6901.3s			
CHC×10	22.1%	619.0%	501.5s	13.9%	493.2%	992.7s	10.4%	404.1%	1483.3s			
CHC	534.5%	862.5%	52.6s	348.7%	812.7%	104.0s	160.6%	770.0%	155.3s			



**Fig. 1.** Comparison of shortest path lengths found during  $T = 1000$  time periods, with 20 generations per change, by P-CHC (left column) and PS-CHC (right column) in: (a) eli51, (b) kroA100 and (c) kroB200. Reference solutions are printed in bold.

## 6 Results

Let  $\mathcal{F}_t^X(p)$  encode the evaluation of individual  $p \in P_t$  obtained using algorithm  $X$  at time step  $t$  of some DTSP and let  $\mathcal{F}_t^{ref}$  be the evaluation of best individual found by  $t$ -th reference algorithm. The two following metrics will be used for presenting the results:

(a) *average deviation* defined as

$$\frac{\sum_{i=1}^R \left[ \min_{p \in P_t} (\mathcal{F}_t^X(p) - \mathcal{F}_t^{ref}) \right]}{R \cdot \mathcal{F}_t^{ref}} \cdot 100\%, \quad (4)$$

(b) *maximum deviation* defined as

$$\frac{\max_{1 \leq i \leq R} \left[ \min_{p \in P_t} (\mathcal{F}_t^X(p) - \mathcal{F}_t^{ref}) \right]}{\mathcal{F}_t^{ref}} \cdot 100\%, \quad (5)$$

where  $R$  is the number of runs. It is worth noticing, that (as one might expect) the expression  $\mathcal{F}_t^X(p) - \mathcal{F}_t^{ref}$  was always positive in all test cases.

Average and maximum deviations for  $R = 10$  runs marked as *avg dev* and *max dev* respectively, together with average computation time are presented in Table 2.

It is clearly seen that CHC enhanced with 2-opt local search delivers evidently best results among the tested algorithms in all cases, however it works very slow. Unexpectedly, its parallel counterpart (i.e. P-CHC + 2-opt/1) performs slightly worse. Note that the source code of P-CHC + 2-opt/1 is in some sense practically the same as CHC + 2-opt, yet runs simultaneously. It is suspected that the discrepancy might be caused by the pseudo-random number generator cuRAND used in PCHC algorithm.

As one could expect, shifting operator scores no better in 2-opt/1, however it does improve 2-opt/ $\delta$  for  $\delta > 1$ . It is worth mentioning, that PS-CHC with 2-opt/2 local search delivered satisfying results requiring very short computation time in all test cases. It is visible that PS-CHC + 2-opt/2 resulted in 6x up to 60x speed-up compared to CHC + 2-opt. In fact, 2-opt/ $\delta$  with  $\delta \in \{2, 3\}$  caused even more speed-up, yet the obtained shortest paths were rather mediocre.

Comparison of shortest path lengths found by P-CHC and PS-CHC with frequency of 20 generations per change are presented in Figure 1.

## 7 Conclusions

Experiments confirms that the original CHC, despite its rapidness, performed rather poorly due to relatively slow (in terms of number of generations) reactivity to environmental changes. Its effectiveness reveals in average deviation scored by CHC x10, however large maximum deviation was definitely below expectations.

The proposed parallelization resulted in 6x up to 60x speed-up compared to CHC + 2-opt. However, the highest speed-up values resulted in significant deterioration of the obtained shortest paths.

## References

1. Branke, J.: Evolutionary Optimization in Dynamic Environments. Kluwer Academic Publishers, Dordrecht (2002)
2. Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of Genetic Algorithms* 1, 265–283 (1991)
3. Filipiak, P., Michalak, K., Lipiński, P.: Infeasibility Driven Evolutionary Algorithm with ARIMA-Based Prediction Mechanism. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) IDEAL 2011. LNCS, vol. 6936, pp. 345–352. Springer, Heidelberg (2011)
4. Filipiak, P., Michalak, K., Lipiński, P.: A Predictive Evolutionary Algorithm for Dynamic Constrained Inverse Kinematics Problems. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part III. LNCS, vol. 7208, pp. 610–621. Springer, Heidelberg (2012)
5. Fujimoto, N., Tsutsui, S.: A Highly-Parallel TSP Solver for a GPU Computing Platform. In: Dimov, I., Dimova, S., Kolkovska, N. (eds.) NMA 2010. LNCS, vol. 6046, pp. 264–271. Springer, Heidelberg (2011)
6. Kirk, D., Hwu, W.: Programming Massively Parallel Processors – A Hands-on Approach. Elsevier (2010)
7. Psaraftis, H.N.: Dynamic vehicle routing problems. In: Vehicle Routing: Methods and Studies, pp. 223–248. Elsevier, Amsterdam (1988)
8. Lipiński, P.: A Stock Market Decision Support System with a Hybrid Evolutionary Algorithm for Many-Core Graphics Processors. In: Guaraccino, M.R., Vivien, F., Träff, J.L., Cannatori, M., Danelutto, M., Hast, A., Perla, F., Knüpfer, A., Di Martino, B., Alexander, M. (eds.) Euro-Par 2010 Workshops. LNCS, vol. 6586, pp. 455–462. Springer, Heidelberg (2011)
9. Lipiński, P.: Parallel Evolutionary Algorithms for Stock Market Trading Rule Selection on Many-Core Graphics Processors. In: Brabazon, A., O'Neill, M., Maringer, D. (eds.) Natural Computing in Computational Finance. SCI, vol. 380, pp. 79–92. Springer, Heidelberg (2012)
10. Lipiński, P.: ECGA vs. BOA in Discovering Stock Market Trading Experts. In: Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2007, pp. 531–538. ACM (2007)
11. Lipiński, P., Korczak, J.J.: Performance Measures in an Evolutionary Stock Trading Expert System. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2004. LNCS, vol. 3039, pp. 835–842. Springer, Heidelberg (2004)
12. Lipiński, P.: Discovering Stock Market Trading Rules Using Multi-layer Perceptrons. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 1114–1121. Springer, Heidelberg (2007)
13. Reinelt, G.: Traveling Salesman Problems Library available on the Internet at:  
<http://comopt.iffi.uni-heidelberg.de/software/TSPLIB95/>
14. Simões, A., Costa, E.: CHC-Based Algorithms for the Dynamic Traveling Salesman Problem. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcázar, A.I., Merelo, J.J., Neri, F., Preuss, M., Richter, H., Togelius, J., Yannakakis, G.N. (eds.) EvoApplications 2011, Part I. LNCS, vol. 6624, pp. 354–363. Springer, Heidelberg (2011)
15. Verhoeven, M.G.A., Aarts, E.H.L., Swinkels, P.C.J.: A parallel 2-opt algorithm for the Traveling Salesman Problem. *Future Generation Computer Systems* 11, 175–182 (1995)
16. Whitley, D., Rana, S., Dzubera, J., Mathias, E.: Evaluating evolutionary algorithms. *Artificial Intelligence* 85, 245–276 (1996)
17. Younes, A., Basir, O., Calamai, P.: A benchmark generator for dynamic optimization. In: Proceedings of the 3rd International Conference on Soft Computing, Optimization, Simulation & Manufacturing Systems (2003)

# Modelling and Recognition of Signed Expressions Using Subunits Obtained by Data–Driven Approach

Mariusz Oszust and Marian Wysocki

Department of Computer and Control Engineering,

Rzeszow University of Technology,

2 Wincentego Pola, 35-959 Rzeszow, Poland

{moszust,mwysocki}@prz-rzeszow.pl

<http://www.prz-rzeszow.pl>

**Abstract.** The paper considers automatic vision based modelling and recognition of sign language expressions using smaller units than words. Modelling gestures with subunits is similar to modelling speech by means of phonemes. To define the subunits a data–driven procedure is proposed. The procedure consists in partitioning time series of feature vectors obtained from video material into subsequences which form homogeneous clusters. The cut points are determined by an optimisation procedure based on quality assessment of the resulting clusters. Then subunits are selected in two ways: as clusters’ representatives or as hidden Markov models of clusters. These two approaches result in differences in classifier design. Details of the solution and results of experiments on a database of 101 Polish words and 35 sentences used at the doctor’s and in the post office are given. Our subunit–based classifiers outperform their whole–word–based counterpart, which is particularly evident when new expressions are recognised on the basis of a small number of examples.

**Keywords:** Sign language recognition, Time series segmentation, Time series clustering, Evolutionary optimisation, Hidden Markov models, Computer vision, Data mining.

## 1 Introduction

Communication disorder such as hearing loss is a significant problem in social contacts. Therefore, there is a justification of research on supporting deaf people with vision–based, automatic sign language recognition system ensuring translation of hand gestures into spoken or written language. In most of such systems (see e.g. [5], [7]) one word model represents one sign. They can achieve good performance only with small sign vocabularies because the training corpus and the training complexity increase with the vocabulary size. Large–vocabulary systems require signed expressions modelling using smaller units than words. Such units are called subunits, and modelling is similar to modelling speech by means of phonemes. The main advantage of this approach is that an enlargement of the vocabulary can be achieved

by composing new signs through concatenation of subunit models and by tuning the composite model with only small sets of examples. In the literature different vision-based subunit segmentation algorithms have been reported presenting how to break down signs into subunits. For example Liddell and Johnson developed movement-hold model [15] modelling each word with series of movement and hold segments. In this approach a signer is assumed to make clear and not natural pauses during signing. Kraiss in [5] presents an iterative process of data-driven extraction of subunits using hidden Markov models (HMMs). In all following steps, two state HMMs for subunits determined in prior iteration step are concatenated to models of single signs. The boundaries of subunits for the next step result from the alignment of appropriate feature vector sequence to the states by the Viterbi algorithm. Theodorakis et al. in [11] describe their approach with pronunciation clustering step with respect to each sign. Pronunciation groups are found by HMMs, and then hierarchically clustered at the HMM level. After clustering model-based subunits are obtained. Kong et al. [4], in turn, use a naive Bayesian network to detect subunits' boundaries. They created subunits' transcription and trained HMMs to recognise the subunits in test expressions. Different approach can be found in [2]. The authors define subunits' boundaries using hand motion discontinuity, adapt temporal clustering by dynamic time warping (DTW) to merge similar time series segments, and finally select a representative subunit from each cluster.

In our work, we propose a new approach to find subunits' boundary points as the solution of an optimisation problem. The problem consists in finding subunits which can be grouped in clusters of good quality. Subunits' boundary points are determined by an immune-based, evolutionary algorithm [1], [13]. The approach refers to clustering of time series data [6], [16], and cluster-based time series segmentation [14]. The contribution of the paper lies in (1) formulation of the problem of determining subunits for sign language recognition as time series clusters' representatives or hidden Markov models of clusters, (2) formulation of the problem of modelling signed expressions with subunits, (3) proposition of solution methods, and verification of the approach by experiments on real data. It is worth noticing that we present the results of experiments when new expressions are recognised based on a small number of examples, justifying the need for the use of subunits models. We also present the recognition of sentences on a basis of subunit models obtained for words.

The rest of the paper is organized as follows. Section 2 gives preliminary information concerning Polish sign language (PSL) and the features derived from sign language videos. Section 3 contains description of the subunits extraction problem. Section 4 gives details on the recognition method. Recognition experiments and their results on 101 isolated PSL words and 35 sentences are given in Sect. 5. Section 6 concludes the paper.

## 2 Characteristics of the PSL and Feature Vectors

PSL signs are static or dynamic and mostly two-handed. Hands often touch each other or appear against the background of the face. Their motion can be single or repeated.

Every sign can be analysed by specifying at least three components: (i) the place of the body against which the sign is made, (ii) the shape of a hand or hands, (iii) the movement of a hand or hands [15]. Although in practical sign language communication some additional features (such as lip shape, face expression, etc.) are often used, we do not consider them in this paper.

For detection of the signer's hands and face we used a method based on a chrominance model of human skin. To detect skin-toned regions in a colour image, the image is transformed into a gray-tone form using the skin colour model in the form of a 2D Gaussian distribution in the normalised RGB space, and thresholded. The individual pixel intensity in a new image represents a probability that the pixel belongs to a skin-toned region. The areas of the objects toned in skin colour, their centres of gravity and ranges of motion are analysed to recognise the right hand, the left hand and the face. Comparison of neighbouring frames helps to notice whether the hands (the hand and the face) touch or partially cover each other. In order to ensure correct segmentation there were some restrictions for the background and the clothing of the signer.

We use seven features for each hand: the position of the hand with respect to the face (three spatial coordinates), the area, orientation, compactness and eccentricity of the hand. In our approach shape of the hand is described in a rough manner due to small hand size in respect to the observed size of the signer hindering more accurate modelling of the hand.

### 3 A Data–Driven Subunits Extraction Method

#### 3.1 The Input Data

Let  $S = \{X_1, X_2, \dots, X_n\}$  denote a data set, where a sequence  $X_i = \{x_i(1), x_i(2), \dots, x_i(T_i)\}$  of real valued feature vectors represents a signed expression. All vectors  $x_i(t)$ , where  $i \in I = \{1, 2, \dots, n\}$ , and  $t$  is a time sampling point,  $t \in T_i = \{1, 2, \dots, T_i\}$ , have identical structure and contain features (see Sect. 2). Two time sequences  $X_i$  and  $X_{j \neq i}$  may represent different expressions or different realisations of the same expression.

In signed expressions modelling we take into account that the features appear both sequentially and simultaneously. For example, the hand shape and hand position can change independently at the same time [5]. To model parallel processes we will distinguish  $N$  groups of features (channels). This is based on the assumption that the separate processes evolve independently from one another with independent output. Therefore, we will write  $x_i(t) = [x_i^1(t), x_i^2(t), \dots, x_i^N(t)]$  and, in accordance with it, we will use an upper index to indicate time series related to a group:  $X_i^l = \{x_i^l(1), x_i^l(2), \dots, x_i^l(T_i)\}$ ,  $S^l = \{X_1^l, X_2^l, \dots, X_n^l\}$ ,  $l \in \mathcal{N} = \{1, 2, \dots, N\}$ . During extraction of subunits all elements in a group will be considered jointly, whereas different groups will be considered separately. For instance, one can distinguish two independent channels ( $N=2$ ) related with both hands, or 14 independent, features related channels.

### 3.2 Time Series Partitioning

Let us consider a time decomposition  $D^l$ , which, for each  $i \in I$ , defines a number  $k_i^l = k_i^l(D^l) \geq 1$  and  $k_i^l - 1$  cut points  $t_{ij}^l = t_{ij}^l(D^l)$ , where  $1 < t_{i1}^l < t_{i2}^l < \dots < t_{i,k_i^l-1}^l < T_i$ . The decomposition means that  $X_i^l$  is partitioned into  $k_i^l$  subsequences. The first subsequence  $s_{i1}^l(D^l)$  starts at  $t = 1$  and ends at  $t = t_{i1}^l$ , the next subsequence  $s_{i2}^l(D^l)$  starts at  $t = t_{i1}^l$  and ends at  $t = t_{i2}^l$ , and so on until the last subsequence  $s_{i,k_i^l}^l(D^l)$  which starts at  $t = t_{i,k_i^l-1}^l$  and ends at  $T_i$ . The resulting data set  $S'^l(D^l) = \{s_1^l(D^l), s_2^l(D^l), \dots, s_n^l(D^l)\}$ , where  $s_i^l(D^l) = \{s_{i1}^l(D^l), \dots, s_{i,k_i^l}^l(D^l)\}$ ,  $i \in I$ , contains  $n^l = n^l(D^l) = \sum_{i=1}^n k_i^l(D^l)$  subsequences. The length of each subsequence is constrained by the minimal  $L_{min}$  and the maximal  $L_{max}$  number of points.

We propose determining a good decomposition into subsequences by solving a decision problem, based on the following main steps: (i) partition the set  $S'^l(D^l)$  into  $m^l$  (a given number) clusters, i.e.  $S'^l(D^l) = \{C_1^l(D^l), C_2^l(D^l), \dots, C_{m^l}^l(D^l)\}$ , (ii) evaluation of the decomposition  $D^l$  using a criterion (index)  $J(D^l)$  which characterises the quality of the resulting clusters. The criterion is the mean distance among the elements and their cluster centres.

### 3.3 Optimisation Method

Our approach to solve the time series partitioning problem (see Subsect. 3.2) adapts the immune-based, evolutionary algorithm CLONALG, which is often used in a wide variety of optimisation tasks [1], [13] especially for solving problems with many local optima and constraints. In the sequel we shortly describe the algorithm, the encoding method, and the mutation operator.

The main loop of the CLONALG algorithm [13] (repeated  $gen$  times, where  $gen$  is the number of generations) consists of four main steps: one initial step where all the elements of the population are evaluated computing  $J(D^l)$  and three transformation steps: clonal selection, mutation, apoptosis. Elements in the population are often called lymphocytes or antibodies. The antibody represents a decomposition  $D^l$  of the set  $S^l$  into a set  $S'^l$ . It has the form of the integer valued vector  $D^l = [t_{11}^l, t_{12}^l, \dots, t_{1,k_1^l-1}^l, t_{21}^l, t_{22}^l, \dots, t_{2,k_2^l-1}^l, \dots, t_{n1}^l, t_{n2}^l, \dots, t_{1,k_n^l-1}^l]$  composed of the cut points of the original sequences. In the clonal selection step algorithm chooses a reference set consisting of  $h$  elements at the top of the ranking. The mutation process consists of a given number  $mut$  of mutations conducted on a population element. The mutation means an operation changing solution maintained in antibody which satisfies the length constraints. In the step of the apoptosis  $b$  worst elements in population are replaced by randomly generated elements.

Before clustering we must define a distance between sequences. We used dynamic time warping (DTW) [9], which allows a nonlinear mapping of one sequence to another by minimizing a distance between them. The main motivation for using DTW is its ability to expand or compress the time comparing sequences that are similar but locally out of phase. For example, some related

parts of gestures representing the same expression may be performed with different velocities. Length of compared sequences can be different. In experiments we used K-means clustering algorithm [16], [12], which works with vector defined data due to calculation of clusters' centres at each step. Center of the cluster in K-means is a mean of all its elements, and it requires that elements in a cluster should have equal length. Therefore, we considered two approaches based on  $n^l(D^l)$  similarity vectors representing the set  $S^l(D^l)$  of sequences to be clustered. Each of the similarity vectors has  $n^l(D^l)$  elements where the  $j$ -th element of the  $i$ -th similarity vector is determined as the DTW distance between sequences  $s'_i$  and  $s'_j$  in the set  $S^l(D^l)$ . In second approach sequences were represented by short vectors containing statistical information (i.e. mean and standard deviation). DTW similarity vectors are large then generating shorter vectors by principal component analysis (PCA) [12] may be a solution. In our experiments we used short vectors.

The optimisation results in obtaining a good decomposition  $D_{opt}^l$ . We can use it to transform each sequence  $X_i^l$  to a string of labels  $X_i^{ls} = \{e_{i1}^l, e_{i2}^l, \dots, e_{i,k_i^l}^l\}$ , where  $e_{ik}^l \in E^l = \{\alpha_1^l, \alpha_2^l, \dots, \alpha_{m^l}^l\}$ ,  $\alpha_k^l$  denotes the label assigned to the cluster  $C_k^l(D_{opt}^l)$ , and  $e_{ik}^l$  is a label of the cluster the subsequence  $s_{ik}^l(D_{opt}^l)$  belongs to. Let us denote by  $X_i^s$  the string-based counterpart of  $X_i$ , i.e.  $X_i = \{X_i^{1s}, X_i^{2s}, \dots, X_i^{Ns}\}$  and, consequently, by  $S^s$  the counterpart of  $S$ .

## 4 Recognition

The subunits can be selected in two ways: as clusters' representatives or HMMs of clusters. Two types of subunits result in differences in a classifier design. Let us assume that an expression to be classified is represented by a sequence  $Y = \{y(1), y(2), \dots, y(T_y)\}$ . The feature vectors  $y(.)$  have the same structure as  $x(.)$  and therefore the sequences  $Y^l = \{y^l(1), y^l(2), \dots, y^l(T_y)\}$ , where  $l \in \mathcal{N}$ , will be considered separately. Two problems have to be solved. The first problem consists in finding an appropriate string representation of  $Y^l$ , i.e.  $Y^{ls} = \{e_{y1}^l, e_{y2}^l, \dots, e_{y,k_y^l}^l\}$ , where  $e_{yk}^l \in E^l$  and, consequently, the string representation  $Y^s$  of  $Y$  (according to the first, aforementioned representation of signed expressions). The second problem is to find  $NN(Y^s)$  – the nearest neighbour of  $Y^s$  in the set  $S^s$ . Then the unknown expression is assigned to the class which  $NN(Y^s)$  belongs to. The string representation can be found by solving an optimisation problem with respect to cut points of  $Y^l$  for each  $l \in \mathcal{N}$ . Let  $D_y^l = [t_{y1}^l, t_{y2}^l, \dots, t_{y,k_y^l-1}^l]$  characterises a decomposition. As opposed to the previous optimisation, now the criterion to be minimized is  $J(D_y^l) = \sum_{k=1}^{k_y^l} d_{DTW}(k)$ , where  $d_{DTW}(k)$  denotes the DTW distance between the  $k$ -th subsequence  $s_{y,k}^l(D_y^l)$  of  $Y^l$  and its nearest neighbour  $NN(s_{y,k}^l(D_y^l))$  in the set  $S^l(D_{opt}^l)$ .

The optimisation task is solved by CLONALG. Then  $e_{yk}^l$  is a label of the cluster the  $NN(s_{y,k}^l(D_{y,opt}^l))$  belongs to. The procedure is repeated for each  $l \in \mathcal{N}$ . The second problem is also an optimisation task. Here the so called edit distance [16] is used as a measure of the difference between two strings.

The method resembles DTW. It uses dynamic programming to find a minimum number of operations (insert, delete, replace) required to transform one string into the other. The sequence  $Y$  becomes assigned to the class  $X_j$  belongs to, where edit distance is minimal.

In the second representation of signed expressions sequences which belong to a cluster were used to train HMMs [8]. A HMM is a model used to characterize the statistical properties of a signal, consists of two stochastic processes: one is unobservable Markov chain with a finite number of states, an initial state probability distribution and a state transition probability matrix; the other is characterized by a set of probability density functions associated with observations generated by each state. We assumed that each cluster can be represented by one state HMM with Gaussian output. We used HTK software [17] to design the HMM-based models. Words (sentences) were recognized using a composite model built as a network of isolated subsequence models. The scheme used a statistical information about the transition probability between two successive subunits, calculated for any subunit in relation to each possible preceding subunit from the training corpus (bigram language model [5], [17]). The parsing were performed by a Viterbi algorithm based on token passing [17]. The modelling was proceeded in two steps. First, isolated subunit models were trained using the Viterbi algorithm and appropriate training data. Then parameters of these models were tuned on the basis of whole words/sentences. The HTK offers so-called embedded training that makes it possible. Embedded training uses the same procedure as for the isolated subunit case, but rather than training each model individually, all models are trained simultaneously. The location of subunit boundaries in the training data is not required for this procedure, but the symbolic transcription of each training sequence is needed [17]. This transcription is obtained during the subunit determining process described earlier. Networks of elementary HMMs representing whole words are automatically created.

## 5 Experiments

In order to examine usability of designed subunit-based classifiers in recognition task we have performed set of experiments on real sequences obtained for signed Polish words and sentences. The sequences represent 101 words and 35 sentences which can be used at the doctor's and in the post office. Each expression was performed 20 times by two signers (resulting in 4040 words' realisations and 1400 sentences' realisations). One signer is a PSL teacher, the other has learnt PSL for purposes of this research. The data have been registered with the rate 25 frames/s. The following Subsections present results of recognition. First we use cross-validation to estimate performance of the subunit-based classifiers of isolated words. Next we consider recognition of new words, i.e. not included in the vocabulary used to determine the subunits, on the basis of a small number of examples. Last experiment concerns recognition of PSL sentences on the basis of subunits determined from words.

## 5.1 Cross-Validation

All word's realisations were divided into ten disjoint subsets in order to perform cross-validation tests. Each subset consisted of data corresponding to four repetitions of each word (two repetitions performed by each signer). We performed ten experiments using nine subsets as the training set  $S$  and the remaining subset as the test set. Because of the stochastic nature of the optimisation method each experiment has been repeated ten times. Subunits for each feature were extracted independently ( $N = 14$ ). Parameters used by immune algorithm were as follows:  $B = h = 20$ ;  $c = 5$ ;  $b = 2$ ;  $mut = 2$ ;  $gen = 100$ ;  $L_{min} = 4$ ;  $L_{max} = 8$ . The optimisation task was solved for  $m^l = 10$  clusters.

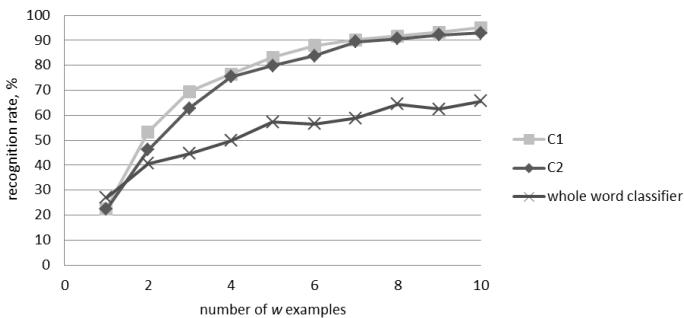
Table II shows the recognition performance during cross-validation for two classifiers (see Subsect. 4). Determining subunits during the learning stage took about 11 minutes, on the processing unit with 3.3 GHz, 4 cores, 16GB RAM, Windows 7 64 bit, and Java. The average time needed for recognition of one gesture from the test set was approximately 1 s. Representing the clusters of subunits obtained during the training by its medoids, and using the medoids instead of the whole clusters, speeds up the optimisation step described in Section 4, which makes the recognition about seven times faster (approx. 0.15 s) and with HMMs models of clusters approx. 0.26 s, but at the cost of recognition rate (approx. 2 percent). We also performed an experiment showing the impact of the number of clusters on recognition rate, good results (in the sense of ten-fold cross-validation) were obtained for more than five clusters.

**Table 1.** Sample results of the cross validation test for two classifiers based on different subunits' models. Because of the stochastic nature of the optimisation method each experiment has been repeated ten times. Mean recognition rate and standard deviation in %.

Testing subset	1	2	3	4	5	6	7	8	9	10
Subunits as subsequences in clusters										
Mean	96.88	99.11	99.38	99.11	99.58	99.01	98.32	99.41	99.08	98.94
StDev	0.50	0.42	0.46	0.35	0.17	0.44	0.33	0.42	0.35	0.57
Subunits as HMMs of clusters										
Mean	89.70	95.22	84.93	97.00	95.17	96.51	94.03	93.29	94.23	96.66
StDev	0.97	1.04	1.58	0.77	1.66	0.69	1.10	2.16	1.57	0.82

## 5.2 Recognition of New Words and Sentences

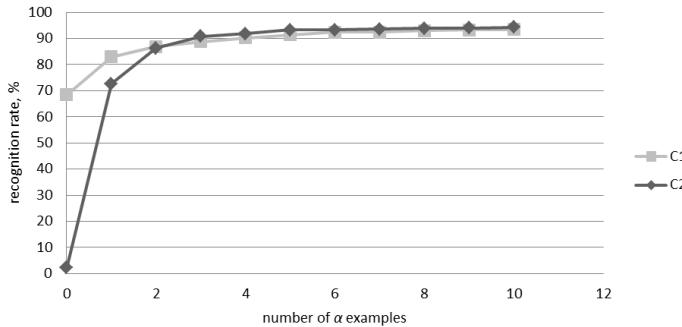
This Subsection considers a situation when some new words are recognised on the basis of small sets of examples presenting usability of our subunit-based approach in extending vocabulary of recognised expressions. Experiments presented below are motivation of the use of subunits in large-vocabulary systems. We used the same data-set as before. Instead of preparing new data we randomly chose ten words (since then called new) from the data set and omitted them from



**Fig. 1.** Recognition rate in % of new words vs. the number  $w$  of examples.  $C1$  denotes classifier using subunits taken from subsequences' clusters,  $C2$  denotes classifier using HMMs based subunits.

the process of determining subunits. So, models of subunits were determined on the basis of data related to remaining 91 words. A small number  $w$  of examples of each new word was used to tune subunit-based models of these words. Remaining  $40 - w$  examples of the new words were used for testing. This experiment has been repeated 20 times, each time with different group of new words. We repeated the experiment with whole word models and nearest neighbour classifier based on DTW distance. Figure 1 shows mean values of recognition rates in relation to the number  $w$  of examples. As we can see, a relatively small number of examples enables good recognition. Better recognition rate obtained by classifiers based on subunits can be explained as follows: the whole word models were represented by small sets of examples, whereas the subunit-based models additionally used information accumulated in subunits that have been modelled on the basis of large sets.

Each of 40 realisations of 35 sentences was optimally transcribed using the subsequences obtained for isolated words. The transcriptions were performed in the way described in Section 4, but here the sequences represented sentences instead of words. The resulting models will be called A-models. Another model of each sentence (B-model) was created directly from concatenated transcriptions of constitutive words. It represents idealized training examples, which do not take into account the coarticulation phenomenon [10]. We considered the recognition rate in dependence on the number of examples. So for each sentence we took its B-model and a small number  $\alpha$  of its drawn A-models, as the training set, and the remaining  $40 - \alpha$  A-models for testing. Experiment was repeated 20 times. The recognition rates are shown in Fig. 2. As we can see, a relatively small number of examples enables good recognition. Note that the quite acceptable result of 68.4% for  $\alpha = 0$  corresponds to the case when only B-models are present in the learning set. It is the common situation when a word in a sentence differs from its isolated realisation. Results obtained for classifier based on subunits modelled by HMMs are weaker for small number of sentences examples because some resulted HMMs of sentences were too long (sometimes more than 20 states).



**Fig. 2.** Recognition rate in % of sentences vs. the number  $\alpha$  of examples.  $C1$  denotes classifier using subunits taken from subsequences' clusters,  $C2$  denotes classifier using HMMs based subunits.

## 6 Conclusions

Large-vocabulary sign language recognition require the modelling of signed expressions using smaller units than words. However, an additional knowledge of how to break down signs into subunits is needed. In vision-based systems the subunits are related to visual information. As linguistic knowledge about the useful partition of signs in regard of sign recognition is not available, the construction of an accordant partition is based on a data-driven process when signs are divided into segments that have no semantic meaning - then similar segments can be grouped and labelled as a subunit or an HMM model using cluster of such segments can be trained. In this paper we propose a new approach to determining the subunits, which boundaries are considered as decision variables in an optimisation problem. Our approach has been successfully verified on a data base of 101 Polish sign language expressions. In future research we are considering more advanced experimentation including recognition words and sentences of PSL.

**Acknowledgment.** This research was partially supported by Polish Ministry of Science and Higher Education under the grant no. U-8608/DS/M.

## References

1. De Castro, L.N., Von Zuben, F.J.: Learning and optimization using the clonal selection principle. *IEEE Trans. on Evolutionary Computation* 6, 239–251 (2002)
2. Han, J., Awad, G., Sutherland, A.: Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recognition Letters* 30, 623–633 (2009)
3. Hendzel, J.K.: Dictionary of Polish Sign Language. OFFER, Olsztyn (1985) (in Polish)

4. Kong, W.W., Ranganath, S.: Automatic Hand Trajectory Segmentation and Phoneme Transcription for Sign Language. In: 8th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2008, pp. 1–6 (2008)
5. Kraiss, K.F.: Advanced man–machine interaction. Springer, Berlin (2006)
6. Liao, T.W.: Clustering of time series data – a survey. Pattern Recognition 38, 1857–1874 (2005)
7. Ong, S.C.W., Ranganath, S.: Automatic sign language analysis: a survey and the future beyond lexical meaning. IEEE Trans. Pattern Anal. Mach. Intell. 27(6), 873–891 (2005)
8. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. IEEE Proceedings 77, 257–286 (1989)
9. Ratanamahatana, C.A., Keogh, E.: Three myths about dynamic time warping data mining. In: SIAM Int. Conf. on Data Mining, pp. 506–510 (2005)
10. Segouat, J., Braffort, A.: Toward Modeling Sign Language Coarticulation. In: Kopp, S., Wachsmuth, I. (eds.) GW 2009. LNCS, vol. 5934, pp. 325–336. Springer, Heidelberg (2010)
11. Theodorakis, S., Pitsikalis, V., Maragos, P.: Model–level data-driven sub–units for signs in videos of continuous sign language. In: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 2262–2265 (2010)
12. Theodoridis, A., Kontroumbas, K.: Pattern Recognition. Acad. Press, London (1999)
13. Trojanowski, K., Wierzchon, S.: Immune-based algorithms for dynamic optimization. Information Sciences 179, 1495–1515 (2009)
14. Tseng, V.S., Chen, C.H., Huang, P.C., Hong, T.P.: Cluster-based genetic segmentation of time series with DWT. Pattern Recognition Letters 30(13), 1190–1197 (2009)
15. Vogler, C., Metaxas, D.A.: Framework for recognizing the simultaneous aspects of American sign language. Computer Vision and Image Understanding 81(3), 358–384 (2001)
16. Xu, R., Wunsch, D.C.: Clustering. J. Wiley and Sons, Inc., Hoboken (2009)
17. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book. Cambridge University (2006)

# Biometrics from Gait Using Feature Value Method

Tianxiang Zhang and Gentiane Venture

Department of Mechanical Systems Engineering,  
Tokyo Univ. of Agriculture and Technology  
venture@cc.tuat.ac.jp

**Abstract.** In order to develop truly intelligent systems, it is necessary to improve their ability to understand non-verbal communication. We propose a novel framework to recognize individuals and emotions from gait, in order to improve HRI. We collected the motion data of the torso from 4 professional actors' gait, using motion capture system, and 7 non-actors' using 2 IMU sensors. We developed Feature Value Method which is a PCA based classifier and finally we achieved high recognition rate through cross-validation.

**Keywords:** Gait, Recognition, PCA, Feature vector.

## 1 Introduction

When sharing the same space as humans, machines must know whom they are interacting with. Also, an intelligent system should understand humans' feelings for a further and more natural interaction. The actions to take with a stranger or a known user, a person who is angry or sad, are different. As well within the group of users, needs and information accessibility may differ. For example the system may act differently with a child or with an elderly. The authentication of the user is thus crucial. The most common authentication methods include voice recognition, face recognition, finger print recognition. So far little motion based biometrics is used.

However, these methods require direct interaction with the user, or cooperation. A face cannot be detected if not facing the system, a voice can hardly be recognized when not speaking or in a noisy environment. These methods must be completed by other methods that can compensate for these drawbacks. Motion based biometrics provides a unique contact-less and non-verbal way to recognize moving individuals and their emotions. It also can be considered as soft biometrics as it preserves privacy; and it is relatively difficult to counterfeit. Psychological studies have shown that one can recognize known individuals and emotional state in absence of anatomical cues by looking solely at the motion. The arm swing and back posture are key features one looks at [1], [2], [3], [4], [5], [6], [7]. Most of the gait studies make use of variables such as the gait cycle frequency, the length of the footsteps that can be easily measured [8], [9]. Some use directly optical marker position captured with a motion capture system [10].

Novel systems of motion capture provide important information and are relatively easy to use. With them, computational tools to access more kinematics variables are

also broadening. Making use of optical passive motion capture system to collect the gait data of several professional actors in several emotional states and use 2 IMU sensors to collect the gait data of some non-actors we show that using Feature Value Method allows for recognizing individuals and emotions.

The paper is structured as follows: first, we detail Feature Value Method for recognition. Then, we present the experimental results on emotion recognition, measured from motion capture systems. Finally we present the results on individual recognition, measured from IMU sensors.

## 2 Feature Value Method

We propose a simple classifier that can classify a data directly in the feature vector space visualized by PCA. First, we calculate the feature vectors of the model we use for recognition [11]. The feature vector of a data-set is computed as the auto-correlation matrix for each component of the data-set. In our case, we use the data measured by the IMU, so that the data-set is composed of the time-sequence of angular velocities or accelerations for each of the  $N_{dof}$  DOF of our model. Let us note  $\dot{q}_i[k] \in \mathbb{R}^{N_{dof}}$  the vector of measured data for the considered  $N_{dof}$  for a motion  $i$  and at time  $k$ . We compute the auto-correlation matrix  $M_i(l)$  as equation (1).  $l$  is a constant time difference, here we set  $l = 2$ . Then we arrange the elements of  $M_i(l)$  into a single column vector. The result is the feature vector  $m_i(l) \in \mathbb{R}^{N_{dof}^2}$ .

$$M_i(l) = \frac{1}{T_i} \sum_{k=1}^{T_i} \dot{q}_i[k] \dot{q}_i^T[k - l] \quad (1)$$

PCA of the obtained feature vector provides information of the clustering possibility of the training data-set. Consequently, it gives information on the possibility to discriminate a data-set from another data-set. Applied to individual recognition, it means that it gives information on the differences and resemblances of different motion data-set, thus different individuals. Depending on the resemblances, points create clusters of various shapes, in the space of principal components, which are dense or scattered. Often the 3-D space of the first three principal components is used. The 2-D space can also be used if the cluster structure is clear enough using only the first two components. The shape of a cluster highlights data-set with similarities, while scattered points represent data-set with little similarity to each other.

1. We collect data for each person or emotion we want to include in our database of known persons to create a minimal knowledge or training data.
2. We compute the feature vector for this data and generate the points  $T_i^p$   $(x_i^p, y_i^p, \dots)$  in the PCA space for each person or emotion  $p$  and each experiment  $i$  as detailed in Section 2.
3. We compute the barycenter of the class of each  $p$  for the  $n$  training data by:

$$A^p(x^p, y^p, \dots) = \frac{1}{n} \sum_{i=1}^n (x_i^p, y_i^p, \dots) \quad (2)$$

4. The distance  $d_j^p$  to any new point  $j$  to the barycenter of  $p$  is given by:

$$d_j^p = \sqrt{(x^p - x_j)^2 + (y^p - y_j)^2 + \dots} \quad (3)$$

5. The least square linear approximation for each class is then calculated. It gives for each person or emotion  $p$ , in the 2-D case:  $y = P_1^p x + P_2^p$ , where  $P_1^p$  and  $P_2^p$  are two real constants. And the orthogonal projection  $h_j^p$  of any point  $j(x_j, y_j)$  to the straight line is given by (4) in 2-D space. In a case of 3-D space, the linear approximation is  $P_1^p x + (Q_1^p - 1)y - z + (P_2^p + Q_2^p) = 0$  and the orthogonal projection is thus given by (5). An example of feature vector space visualized by PCA with barycenters and linear approximations is shown in Fig. 1(left).

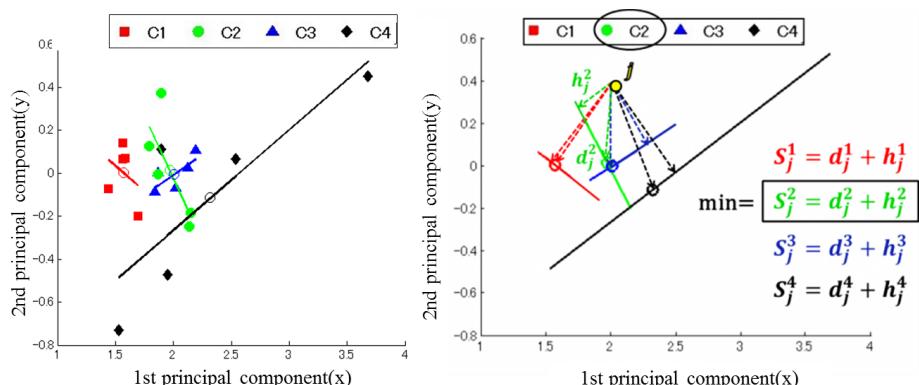
$$h_j^p = \frac{|P_1^p x_j - y_j + P_2^p|}{\sqrt{P_1^{p2} + 1}} \quad (4)$$

$$h_j^p = \left( \left( \frac{|P_1^p x_j - y_j + P_2^p|}{\sqrt{P_1^{p2} + 1}} \right)^2 + \left( \frac{|Q_1^p y_j - z_j + Q_2^p|}{\sqrt{Q_1^{p2} + 1}} \right)^2 \right)^{1/2} \quad (5)$$

6. We define the “feature value”  $S$  of point  $j$  with respect to  $p$  by:

$$S_j^p = d_j^p + h_j^p \quad (6)$$

When implementing recognition, we use PCA to analysis the feature vector, then in the 2-D or 3-D space configured by the first 2 or 3 principal components, if a minimum feature value  $S_j^p$  is found by utilizing a random data point  $j$ , we conclude that this data point fits the person or emotion  $p$  [12], as Fig.1(right) shows. To get more information of the feature vectors we can use the first three principle components and conduct 3-D feature vector space, while we can also just use 2-D



**Fig. 1.** Concept of the proposed algorithm

from the first two in the case of the cluster structure is clear enough so that the calculation can be reduced. If a data is too different from the data in the database, a threshold on  $S_j^p$  can be set to return a “non-recognized candidate” result.

### 3 Experiments

In order to test the feasibility and validate our method on emotion recognition and emotion recognition, we structured experiments by two parts. In experiment A, the first part, we collected 4 actors' gait data at several emotional states using motion capture. In experiment B, the second part, we collected 7 non-actors' gait data only at neutral attitude.

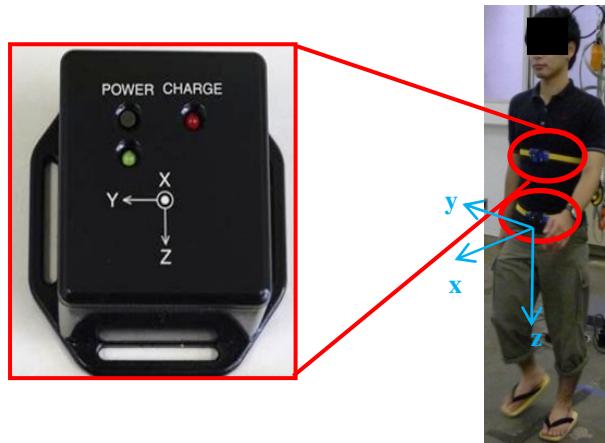
#### 3.1 Experiment A

The experiments were conducted at the Laboratoire de Physiologie de la Perception et de l'Action at Collège de France, Paris, France [13]. 4 candidates who are professional actors were asked to walk through a motion capture area in 6~8 steps' length while displaying the following emotion: neutral attitude, joy, anger, sadness and each was repeated consecutively 5 sequences. Finally 20 motions were obtained for each candidate, which makes a total of 80 motions. The motion capture is a Vicon system and a 41-marker-set is used.

To generate the geometric model, we use a systematic method based on the 3D position of the passive optical reflective marker, as described in [14]. For each candidate, a 34 degrees of freedom (DOF) model, made of 15 rigid links is directly calibrated from the markers information. Using the generated geometric model, the inverse kinematics is computed in order to obtain the 34 joint angles for each motion  $i$  ( $i = 1$  to 80), and the 6 generalized coordinates of the base-link. Psychological studies have shown that one has a tendency to look at the general attitude, the bounciness, the speed of the gait, rhythm to recognize someone [3]. Arm swing and stance length were also some noted criteria. At first we thus proposed to use the 6 DoF of the lower torso which give the general speed, bounciness, and the hip swing information. And then use the upper-torso movements and the head inclinations. A total motion data of 12 DoF are then extracted from the recorded data.

#### 3.2 Experiment B

The experiments are conducted in our lab. 7 candidates (C1~C7) are chosen among students and faculties of our campus (6 males 1 females). One IMU sensor is fixed on the lower torso (pelvis, centered) and the other one is fixed on the upper torso (sternum) of each candidate as Fig.2 shows.



**Fig. 2.** A picture of the experiments and IMU we used

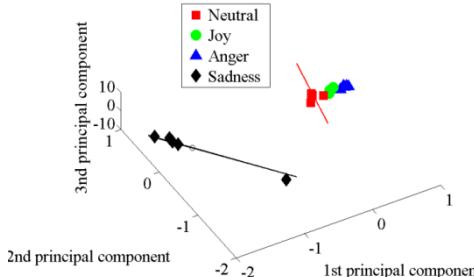
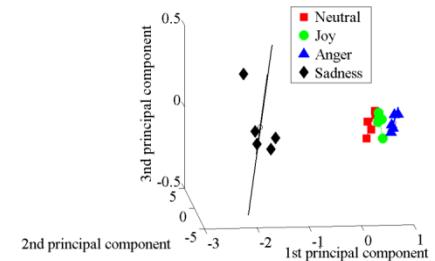
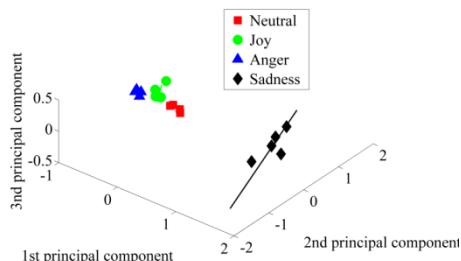
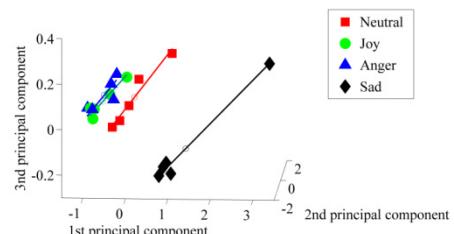
We measured 10 sequences per subject and it is enough to reflect their gait. We can obtain a 12DOF model which is made of the joint angular velocities (lower torso 3DOF, upper torso 3DOF) and the accelerations (lower torso 3DOF, upper torso 3DOF) from the IMU sensors fixed on each candidate. The candidates are asked to walk about 6~8 steps' distance and the experiment is repeated 10 times for each candidate. Thus, a total of 130 time-series motion data that has 12DOF is recorded synchronously by the two IMU sensors.

## 4 Experimental Results and Evaluation

Feature vectors for emotion recognition are computed from the joint angle data of the 12DoF mentioned in experiment A, and ones for individual recognition are computed from the joint angular velocity collected by the 2 IMU (3x2 DOF). Thus,  $m_i(l) \in R^{N_{12}^2}$  in the case of the former,  $m_i(l) \in R^{N_6^2}$  in the case of the later.

### 4.1 Results from Experiment A

The obtained PCA results with linear regressions and barycenters are shown in Fig.3 to Fig. 6 individually. Considering the data are not so plenty, we use all the data of each candidate as training data for the computations of the barycenters and regressions, instead of cross-validation. The recognition results are shown in Table 1.

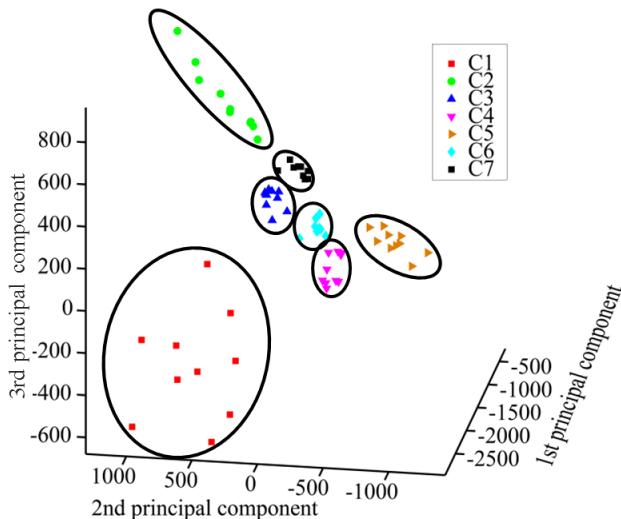
**Fig. 3.** Result of C1**Fig. 4.** Result of C2**Fig. 5.** Result of C3**Fig. 6.** Result of C4**Table 1.** Emotion recognition rate from Experiment A

Recognition rate	Neutral	Joy	Anger	Sadness	Average
Rate of C1	80%	100%	100%	100%	97%
Rate of C2	60%	100%	100%	100%	90%
Rate of C3	100%	100%	100%	100%	100%
Rate of C4	80%	80%	80%	100%	85%

Results shows that the motion in neutral attitude, joy and anger is quite similar but the motion in a sad state differs distinctly since its cluster is always far away from others. However globally the recognition rate is around 90% in average and is much higher than chance: 25%. To improve the accuracy, we can identify whether it is the sadness first and then perform PCA on the remaining data if not. Because the cluster of sadness is far away from others and its rate is always 100% at least in the experiments, such exclusion method can maximize the variance of the remaining three clusters and make the clustering more effective.

## 4.2 Results from Experiment B

First, the obtained results of PCA are shown in Fig.7 and it shows a clear cluster structures. We use 70-fold cross validation, which means a systematic rotation of 69 of the collected data to train our system and then plot the  $T_i^p$  ( $x_i^p, y_i^p, z_i^p$ ) in the feature vector space successively, and use the remaining one as the trial data. Therefore, the trial data is not included in the training data. The recognition rates are shown in Table. 2. The results show that using the proposed method we achieved an average recognition rate of 100% for each subject.



**Fig. 7.** Feature vector space visualized by PCA, obtained from Exp. B

**Table 2.** Recognition rate from Experiment B

Candidate	C1	C2	C3	C4	C5	C6	C7	Average
Rate(%)	100	100	100	100	100	100	100	100

## 5 Conclusions

In this paper we have used motion data of the head, upper torso and lower torso to recognize emotions and use ones of upper torso and lower torso only to recognize individuals. Our method is based on the use of PCA on feature vectors. The gait data is partly collected with motion capture system and partly obtained from 2 IMU sensors. Gait of 4 actors are recorded with a repetition of 5 sequences and gait of non-actors are recorded with a repetition of 10 sequences. Data of the former form 20 motions' database for each subject to run emotion recognition. In the case of the later, those data form a 130 motions' database with 129 training data, and the remaining

one is used for recognition (trial data). Our classifier is based on the computation of the barycenter and the linear regression of the cluster for each candidate in the database. The feature value is computed as a linear combination of the distance to the barycenter and the straight line obtained previously. The smallest feature value designates the recognized candidate. The results show that:

- It is possible to recognize several typical emotion states generated by professional actors from gait.
- It is possible to recognize the person from gait within a 7 persons' database, with a perfect rate (100%) significantly higher than the chance (8%) rate, from joint angular velocity data of the lower and upper torso.
- The results also confirmed the performance of the proposed method for classification. Also, when there are some clusters of training data's classes far away from the others and obviously easy to be classified in the feature vector space, we can identify whether the target is belong to them first, if not, we can run the PCA only to the remaining feature vectors, thus a new cluster structure made of these remaining data works more effectively for recognition.
- Compared with Supporting Vector Machine (SVM) [15], [16], [17], the proposed algorithm for recognition is simpler and takes less computation since its recognition results are acquired by simply summing two distances calculated by barycenter and linear regression between the targets and clusters made of training data, in feature vector space. Further, SVM is a classifier to classify two classes theoretically. Therefore, SVM is not able to be applied to identify multi-class at the same time in the form as it is, unless in the form of combination of multiple discriminant function. However, Feature Value Method is a classifier that can identify multi-class at one time.

The method is promising in developing biometric systems based on gait in order to enhance non-verbal communication abilities of intelligent systems for HRI even with rather simple sensors such as MEMS IMU sensors that allow collecting the data through wireless communication. In this paper we confirmed the ability to recognize individuals and emotions from the motion of torso during gait. In other words, no matter how these data are measured, such as IMU, Camera with computer vision, motion capture or any other devices, we can use the proposed method to implement recognition as long as they are accurately measured.

Further work includes the creation of a comprehensive database of gait data on a wide range of people. To enhance the response to classes in non-linear distribution we also plan to use Kernel PCA instead of PCA and then compare the Kernel based Feature Value Method with non-linear SVM in accuracy and calculation speed [18].

## References

1. Loula, F., Prasad, S., Harber, K., Shiffrar, M.: Recognizing people from their movement. *J. of Experimental Psychology: Human Perception and Performance* 31(1), 210–220 (2005)

2. Jokisch, D., Daum, I., Troje, N.: Self-recognition versus recognition of others by biological motion: Viewpoint-dependent effects. *Perception* 35, 911–920 (2006)
3. Cutting, J., Kozlowski, L.: Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin of Psychonomic Society* 9(5), 353–356 (1977)
4. Jain, A., Bolle, J., Pankanti, S.: *Biometrics-Personal Identification in Networked Society*. Springer US (2002)
5. Nixon, M.: Gait biometrics. *Biometric Technology Today* 16(7-8), 8–9 (2008)
6. de Gelder, B.: Towards the neurobiology of emotional body language. *Nature Reviews, Neuroscience* 7, 242–249 (2006)
7. Taylor, G.W., Hinton, G.E.: Factored conditional restricted Boltzmann Machines for modeling motion style. In: Proc. of the 26th International Conference on Machine Learning, pp. 1025–1032 (2009)
8. Murray, P., Drought, B., Kory, R.: Walking patterns of normal men. *Journal of Bone and Joint Surgery* 46(2), 335–360 (1964)
9. Boyd, J.E., Little, J.J.: Biometric Gait Recognition. In: Tistarelli, M., Bigun, J., Gross, E. (eds.) *Advanced Studies in Biometrics. LNCS*, vol. 3161, pp. 19–42. Springer, Heidelberg (2005)
10. Troje, N.: Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *J. of Vision* 2(5), 371–387 (2002)
11. Kadone, H., Nakamura, Y.: Symbolic Memory of Motion Patterns by an Associative Memory Dynamics with Self-organizing Nonmonotonicity. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) *ICONIP 2007, Part II. LNCS*, vol. 4985, pp. 203–213. Springer, Heidelberg (2008)
12. Zhang, T., Venture, G.: Emotion Recognition from Walk Pattern. In: 21st An. Conf. of the Japanese Neural Network Society, Okinawa, Japan, pp. 196–197 (December 2011)
13. Venture, G.: Human Characterization and Emotion Characterization from Gait. In: Proc. of the IEEE Int. Conf. Engineering in Medicine and Biology, Buenos Aires, Argentina, pp. 1292–1295 (2010)
14. Kadone, H., Hicheur, H., Grezes, J., Berthoz, A.: Nature of the kin-ematic cues underlying the perception of emotions during human gait. In: Proc. of the XIX Conf. of the Int. Society for Posture and Gait Research, vol. O-25 (2009)
15. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer (1995)
16. Burges, C.: A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2, 1–47 (1988)
17. Osuna, E., et al.: *Support Vector Machines: Training and Applications*, A.I. Memo 1602. MIT Press (1988)
18. Karg, M., et al.: A comparison of PCA, KPCA and LDA for feature extraction to recognize affect in gait kinematics. In: Proc. of the 3rd International Conference on Affective Computing and Intelligent Interaction (2009)

# Author Index

- Ali, Mostafa Z. 69  
Amaneddine, Nouhad 152  
Angelova, Galia 48  
Arendarenko, Ernest 89  
  
Balbo, Flavien 276  
Barták, Roman 184  
Baruch, Ieroham 212  
Beldjoudi, Samia 112  
Bessghaier, Nesrine 276  
Boytcheva, Svetla 48  
Broda, Bartosz 23, 32  
Burda, Karel 38  
  
Chauvin, Lionel 119  
Chovanec, Andrej 184  
Condotta, Jean-François 152  
Contet, Jean-Michel 246  
Courdier, Rémy 230  
  
Damova, Mariana 145  
Dařena, František 38  
Dascălu, Mihai 59  
Dessus, Philippe 59  
  
El Zaher, Madeleine 246  
  
Faron-Zucker, Catherine 112  
Filipiak, Patryk 202, 305  
Fomichov, Vladimir A. 296  
  
Gangat, Yassine 230  
Garcia, Laurent 119  
Gechter, Franck 246  
Genest, David 119  
Gomaa, Walid 256  
Gortcheva, Elena 212  
Gruca, Aleksandra 125  
Gunady, Mohamed K. 256  
  
HaCohen-Kerner, Yaakov 1  
Hastings, Janna 103  
Hernandez, Sergio M. 212  
  
Iocchi, Luca 135  
  
Kabziński, Jacek 172  
Kakkonen, Tuomo 89  
Khader, Ahamad Tajudin 178  
Kirillov, Anton V. 296  
Kiryakov, Atanas 145  
Kissoum, Yacine 286  
Kłyk, Łukasz 32  
Kong, Xiangbin 240  
Koukam, Abderrafiaa 246  
Kurc, Roman 23  
Kutz, Oliver 103  
  
Le Dorze, Aymeric 119  
Leonetti, Matteo 135  
Lindgren, Helena 266  
Lipiński, Piotr 202, 305  
Loiseau, Stéphane 119  
Lupan, Diana 59  
  
Maamri, Ramdane 286  
Michałak, Krzysztof 202  
Minda, Paweł 14  
Moreno-Cruz, Jacob 212  
Mossakowski, Till 103  
Myszkowski, Paweł B. 32  
  
Nadi, Farhad 178  
Nieves, Juan Carlos 266  
  
Osareh, Alireza 195  
Oszust, Mariusz 315  
  
Patrizi, Fabio 135  
Payet, Denis 230  
Piasecki, Maciej 14, 23, 32  
  
Qi, Jin 240  
  
Ramocki, Radosław 14, 23  
Rekabdár, Banafsheh 195  
Reynolds, Robert G. 69  
  
Sahnoun, Zaidi 286  
Seridi-Bouchelaghem, Hassina 112  
Shadgar, Bita 195  
Sikora, Marek 125, 162  
Simion, Adrian 223

- Simov, Kiril 145  
Skowron, Adam 162  
Surie, Dipak 266  
  
Takeuchi, Ikuo 256  
Tandukar, Udeep 79  
Tashev, Zdravko 145  
Tayeb, Aharon 1  
Trăusăan-Matu, Stefan 59, 223  
  
Urbansky, David 32
- Vassileva, Julita 79  
Venture, Gentiane 325  
  
Wróbel, Lukasz 162  
Wysocki, Marian 315  
  
Zargayouna, Mahdi 276  
Zhang, Tianxiang 325  
Zhu, Junwei 240  
Žižka, Jan 38