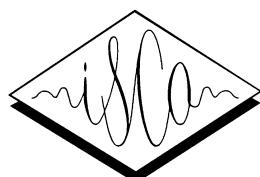


SIGDIAL 2020



**21st Annual Meeting of the
Special Interest Group on Discourse and
Dialogue**



Proceedings of the Conference

01-03 July 2020
1st virtual meeting

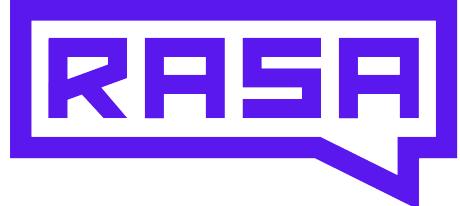
In cooperation with:

Association for Computational Linguistics (ACL)
International Speech Communication Association (ISCA)
Association for the Advancement of Artificial Intelligence (AAAI)

We thank our sponsors:

- Apple
- Rasa Technologies
- Honda Research Institute
- Toshiba Research Europe

Gold



Silver



In cooperation with



©2020 The Association for Computational Linguistics

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-952148-02-6

Introduction

We are excited to welcome you to SIGDIAL 2020, the 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue. This year the conference is being held virtually, on July 1-3, 2020, with the Satellite Event YRRSDS 2020 (Young Researchers' Roundtable on Spoken Dialog Systems) and just before ACL 2020 that will take place also virtually July 5-10, 2020.

The SIGDIAL conference is a premier publication venue for research in discourse and dialogue. This year, the program includes three keynote talks, nine presentation sessions, three demo sessions, and a special session entitled “Situated Dialogue with Virtual Agents and Robots (RoboDial 2.0)” organized by Jose David Lopes, Stephanie Lukin, Matthew Marge, Vikram Ramanarayanan, Matthias Scheutz, Casey Kennington, and Cynthia Matuszek.

We received 104 submissions this year, which comprised 62 long papers, 32 short papers and 10 demo descriptions. This year, for the first time, we had 8 Senior Program Committee (SPC) members who were responsible for a set of 10-15 papers each, guiding the discussion process and writing a meta-review. Every submission was assigned to one SPC and received at least three reviews. When making our selections for the program, we carefully considered the reviews, meta-reviews and the comments made during the discussions among reviewers. The members of the Senior Program Committee and Program Committee did an excellent job in reviewing the submitted papers, and we thank them for their essential role in selecting the accepted papers and helping produce a high quality program for the conference. In line with the SIGDIAL tradition, our aim has been to create a balanced program that accommodates as many favorably rated papers as possible. We accepted 41 papers: 23 long papers, 10 short papers, and 8 demo descriptions. These numbers give an overall acceptance rate of 39%. The acceptance rate for long papers (37%) and short papers (31%) remains in line with the acceptance rate from last year.

Each of the three conference days features one keynote and several oral sessions, with the remaining time given to demos, special session and sponsor sessions. In organizing the virtual conference, we decided to keep as much as possible the spirit of an in person conference. All keynotes, talks and demos are pre-recorded and made available at the beginning of the conference for participants to watch asynchronously. The long and short papers are organized in thematic sessions and take into consideration the speakers' different time zones. The sessions contain 3-4 pre-recorded talks followed by a Live QA part with the presenters. For demos, we organized Live Question Answering sessions with the demo presenters. Topic-wise, we have papers on evaluation and corpora, natural language generation, task oriented dialogues, knowledge use and acquisition, behaviour modeling, dialogue policy and dialogue state tracking, modeling convergence in dialogues, and the semantics and pragmatics of discourse and dialogue.

A conference of this scale requires advice, help and enthusiastic participation of many parties, and we have a big ‘thank you’ to say to all of them. Regarding the program, we thank our three keynote speakers, Asli Celikyilmaz (Microsoft Research), Diane Litman (University of Pittsburgh) and Gabriel Skantze (KTH Royal Institute of Technologies), for their inspiring talks on "Neural text Generation: Progress and Challenges", "Argument Mining, Discourse Analysis, and Educational Applications" and "Conversational Turn-taking in Human-robot Interaction". We also thank the organizers of the special session on Situated Dialogue with Virtual Agents and Robots (RoboDial 2.0). We are grateful for their smooth and efficient coordination with the main conference.

We extend special thanks to our Local Chair, Casey Kennington, for handling the situation of adapting to a virtual conference. SIGDIAL 2020 would not have been possible without his effort in arranging the virtual platform, handling registration, numerous preparations for the conference, and last but not least, Casey’s personal contributions, which exceeded those of a local organizer. We also thank the virtual presentation co-chairs, Koji Inoue and Erik Ekstedt, for helping the authors with their video

presentations, arranging for the video streaming during the conference and hosting the Zoom Live QAs sessions.

David Vandyke, our Sponsorship Chair, has conducted the massive task of recruiting and liaising with our conference sponsors, many of whom continue to contribute year after year. We thank David for his dedicated work and his assistance with conference planning. We gratefully acknowledge the support of our sponsors: (Gold level) Apple and Rasa Technologies and (Silver level) Toshiba Research Europe and Honda Research Institute.

In addition, we thank Nina Dethlefs, Mentoring Chair for SIGDIAL 2020, for her dedicated work on the mentoring process. The goal of mentoring is to assist authors of papers that contain important ideas but require significant stylistic modifications, and we thank our mentoring team for their excellent support of the authors; and Stefan Ultes, our publication chair, capped the long organizational process by putting together these high quality conference proceedings.

We thank the SIGdial board, both current and emeritus officers, Gabriel Skantze, Mikio Nakano, Vikram Ramanarayanan, Ethan Selfridge, Jason Williams and Amanda Stent, for their advice and support from beginning to end.

We once again thank our senior program committee members (Dilek Hakkani-Tur, Annie Louis, Mikio Nakano, Rebecca J. Passonneau, Gabriel Skantze, Manfred Stede, David Traum, Koichiro Yoshino) and program committee members for committing their time to help us select an excellent technical program. Finally, we thank all the authors who submitted to the conference and all conference participants for making SIGDIAL 2020 a success and for growing the research areas of discourse and dialogue with their fine work.

Olivier Pietquin, General Chair

Smaranda Muresan and Yun-Nung (Vivian) Chen, Program Co-Chairs

General Chair:

Olivier Pietquin, Google Research, France

Program Chairs:

Smaranda Muresan, Columbia University, USA
Vivian Chen, National Taiwan University, Taiwan

Local / Virtual Chair:

Casey Kennington, Boise State University, USA

Sponsorship Chair:

David Vandyke, Apple Inc., United Kingdom

Mentoring Chair:

Nina Dethlefs, University of Hull, United Kingdom

Publication Chair:

Stefan Ultes, Mercedes-Benz AG, Germany

Video Presentation Chair:

Koji Inoue, Kyoto University, Japan
Erik Ekstedt, KTH Royal Institute of Technology, Sweden

SIGdial Officers:

President: Gabriel Skantze, KTH Royal Institute of Technology, Sweden
Vice President: Mikio Nakano, Honda Research Institute, Japan
Secretary: Vikram Ramanarayanan, Educational Testing Service (ETS) Research, USA
Treasurer: Ethan Selfridge, Interactions, USA
President Emeritus: Jason Williams, Apple, USA

Senior Program Committee:

Dilek Hakkani-Tur, Amazon
Annie Louis, Google
Mikio Nakano, Honda Research Institute
Rebecca J. Passonneau, Penn State University
Gabriel Skantze, KTH
Manfred Stede, Universität Potsdam
David Traum, University of Southern California
Koichiro Yoshino, Nara Institute of Science and Technology

Program Committee:

Sean Andrist, Microsoft Research, United States
Masahiro Araki, Kyoto Institute of Technology, Japan
Ron Artstein, USC Institute for Creative Technologies, United States
Yoav Artzi, Cornell University, United States
Timo Baumann, Universität Hamburg, Germany
Frederic Bechet, Aix Marseille Universite - LIS/CNRS, France
Steve Beet, Aculab plc, United Kingdom
Jose Miguel Benedi, Universitat Politècnica de València, Spain
Luciana Benotti, Universidad Nacional de Cordoba, Argentina
Yonatan Bisk, Carnegie Mellon University, United States
Nate Blaylock, Cerence, United States
Dan Bohus, Microsoft Research, United States
Johan Boye, KTH, Sweden
Chloé Braud, IRIT - CNRS, France
Hendrik Buschmeier, Bielefeld University, Germany
Andrew Caines, University of Cambridge, United Kingdom
Christophe Cerisara, Université de Lorraine, CNRS, LORIA, France
Senthil Chandramohan, Microsoft, United States
Lin Chen, Head of AI, Cambia Health Solutions, United States
Paul Crook, Facebook, United States
Heriberto Cuayahuitl, University of Lincoln, United Kingdom
Nina Dethlefs, University of Hull, United Kingdom
David DeVault, University of Southern California, United States
Barbara Di Eugenio, University of Illinois at Chicago, United States
Jens Edlund, KTH Speech, Music and Hearing, Sweden
Maxine Eskenazi, Carnegie Mellon University, United States
Keelan Evanini, Educational Testing Service, United States
Mauro Falcone, Fondazione Ugo Bordoni, Italy
Michel Galley, Microsoft Research, United States
Milica Gasic, Heinrich Heine University Duesseldorf, Germany
Kallirroi Georgila, University of Southern California, ICT, United States
Alborz Geramifard, Facebook AI, United States
Debanjan Ghosh, Educational Testing Service, United States
Jonathan Ginzburg, Université Paris-Diderot (Paris 7), France
Joakim Gustafson, KTH, Sweden
Ivan Habernal, Technische Universität Darmstadt, Germany
Helen Hastie, Heriot-Watt University, United Kingdom
Michael Heck, Heinrich Heine University, Germany
Behnam Hedayatnia, Amazon, United States
Ryuichiro Higashinaka, NTT Media Intelligence Labs., Japan
Takuya Hiraoka, NEC Central Research Laboratories, Japan
Thomas Howard, University of Rochester, United States
David M. Howcroft, Heriot-Watt University, United Kingdom
Ruihong Huang, Texas A&M University, United States
Michimasa Inaba, The University of Electro-Communications, Japan
Koji Inoue, Kyoto University, Japan
Filip Jurcicek, Apple Inc., United Kingdom
Tatsuya Kawahara, Kyoto University, Japan

Chris Kedzie, Columbia University, United States
Simon Keizer, Toshiba Research Europe Ltd, United Kingdom
Chandra Khatri, Senior AI Research Scientist, Uber AI, United States
Alexander Koller, Saarland University, Germany
Kazunori Komatani, Osaka University, Japan
Ivana Kruijff-Korbayova, DFKI, Germany
Kornel Laskowski, Carnegie Mellon University, United States
Fabrice Lefevre, Avignon Univ., France
Oliver Lemon, Heriot-Watt University, United Kingdom
Junyi Jessy Li, University of Texas at Austin, United States
Pierre Lison, Norwegian Computing Centre, Norway
Bing Liu, Facebook, United States
Eduardo Lleida Solano, University of Zaragoza, Spain
Ramon Lopez-Cozar, University of Granada, Spain
Nurul Lubis, Heinrich Heine University, Germany
Ross Mead, Semio, United States
Teruhisa Misu, Honda Research Institute USA, United States
Seungwhan Moon, Facebook Conversational AI, United States
Raymond Mooney, University of Texas at Austin, United States
Elena Musi, University of Liverpool, United Kingdom
Satoshi Nakamura, Nara Institute of Science and Technology and RIKEN AIP Center, Japan
Vincent Ng, University of Texas at Dallas, United States
Douglas O'Shaughnessy, INRS-EMT (Univ. of Quebec), Canada
Alexandros Papangelis, Uber AI, United States
Cecile Paris, CSIRO, Australia
Nanyun Peng, University of Southern California, United States
Laura Perez-Beltrachini, School of Informatics, University of Edinburgh, United Kingdom
Paul Piwek, The Open University, United Kingdom
Heather Pon-Barry, Mount Holyoke College, United States
Andrei Popescu-Belis, HEIG-VD / HES-SO, Switzerland
Abhinav Rastogi, Google Research, United States
Ehud Reiter, University of Aberdeen, United Kingdom
Norbert Reithinger, DFKI GmbH, Germany
Antonio Roque, Tufts University, United States
Carolyn Rose, Carnegie Mellon University, United States
Clayton Rothwell, Infoscitex Corp., United States
Sakriani Sakti, Nara Institute of Science and Technology (NAIST) / RIKEN AIP, Japan
Ruhi Sarikaya, Amazon, United States
David Schlangen, University of Potsdam, Germany
Ethan Selfridge, Interactions LLC, United States
Georg Stemmer, Intel Corp., Germany
Matthew Stone, Rutgers University, United States
Svetlana Stoyanchev, Toshiba Europe, United Kingdom
Kristina Striegnitz, Union College, United States
Pei-Hao Su, PolyAI, United Kingdom
Hiroaki Sugiyama, NTT Communication Science Labs., Japan
António Teixeira, DETI/IEETA, University of Aveiro, Portugal
Takenobu Tokunaga, Tokyo Institute of Technology, Japan
Bo-Hsiang Tseng, University of Cambridge, United Kingdom
Gokhan Tur, Amazon Alexa AI, United States

Stefan Ultes, Mercedes-Benz AG, Germany
David Vandyke, Apple, United Kingdom
Hsin-Min Wang, Academia Sinica, Taiwan
Yi-Chia Wang, Uber AI, United States
Nigel Ward, University of Texas at El Paso, United States
Jason D Williams, Apple, United States
Tom Williams, Colorado School of Mines, United States
Yen-chen Wu, University of Cambridge, United Kingdom
Steve Young, Cambridge University, United Kingdom
Zhou Yu, University of California, Davis, United States
Kai Yu, Shanghai Jiao Tong University, China
Jian ZHANG, Dongguan University of Technology and Hong Kong University of Science and Technology, China

Secondary Reviewers: Zeyu Dai, Shrey Desai, Sanuj Sharma, Wenlin Yao

Invited Speakers:

Asli Celikyilmaz, Microsoft Research, USA
Diane Litman, University of Pittsburgh, USA
Gabriel Skantze, KTH Royal Institute of Technology, Sweden

Table of Contents

<i>Semantic Guidance of Dialogue Generation with Reinforcement Learning</i>	
Cheng-Hsun Hsueh and Wei-Yun Ma	1
<i>Counseling-Style Reflection Generation Using Generative Pretrained Transformers with Augmented Context</i>	
Siqi Shen, Charles Welch, Rada Mihalcea and Verónica Pérez-Rosas	10
<i>Learning from Mistakes: Combining Ontologies via Self-Training for Dialogue Generation</i>	
Lena Reed, Vrindavan Harrison, Shereen Oraby, Dilek Hakkani-Tur and Marilyn Walker	21
<i>TripPy: A Triple Copy Strategy for Value Independent Neural Dialog State Tracking</i>	
Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi and Milica Gasic	35
<i>Conversational Agents for Intelligent Buildings</i>	
Weronika Sieńska, Christian Dondrup, Nancie Gunson and Oliver Lemon	45
<i>Retico: An incremental framework for spoken dialogue systems</i>	
Thilo Michael	49
<i>MC-Saar-Instruct: a Platform for Minecraft Instruction Giving Agents</i>	
Arne Köhn, Julia Wichlacz, Christine Schäfer, Álvaro Torralba, Joerg Hoffmann and Alexander Koller	53
<i>ConvoKit: A Toolkit for the Analysis of Conversations</i>	
Jonathan P. Chang, Caleb Chiam, Liye Fu, Andrew Wang, Justine Zhang and Cristian Danescu-Niculescu-Mizil	57
<i>Commonsense Evidence Generation and Injection in Reading Comprehension</i>	
Ye Liu, Tao Yang, Zeyu You, Wei Fan and Philip S. Yu	61
<i>Identifying Collaborative Conversations using Latent Discourse Behaviors</i>	
Ayush Jain, Maria Pacheco, Steven Lancette, Mahak Goindani and Dan Goldwasser	74
<i>A Case Study of User Communication Styles with Customer Service Agents versus Intelligent Virtual Agents</i>	
Timothy Hewitt and Ian Beaver	79
<i>It's About Time: Turn-Entry Timing For Situated Human-Robot Dialogue</i>	
Felix Gervits, Ravenna Thielstrom, Antonio Roque and Matthias Scheutz	86
<i>Learning Word Groundings from Humans Facilitated by Robot Emotional Displays</i>	
David McNeill and Casey Kennington	97
<i>Learning and Reasoning for Robot Dialog and Navigation Tasks</i>	
Keting Lu, Shiqi Zhang, Peter Stone and Xiaoping Chen	107
<i>An Attentive Listening System with Android ERICA: Comparison of Autonomous and WOZ Interactions</i>	
Koji Inoue, Divesh Lala, Kenta Yamamoto, Shizuka Nakamura, Katsuya Takanashi and Tatsuya Kawahara	118

<i>A Spoken Dialogue System for Spatial Question Answering in a Physical Blocks World</i>	128
Georgiy Platonov, Lenhart Schubert, Benjamin Kane and Aaron Gindi	128
<i>rrSDS: Towards a Robot-ready Spoken Dialogue System</i>	132
Casey Kennington, Daniele Moro, Lucas Marchand, Jake Carns and David McNeill	132
<i>Discovering Knowledge Graph Schema from Short Natural Language Text via Dialog</i>	136
Subhasis Ghosh, Arpita Kundu, Aniket Pramanick and Indrajit Bhattacharya	136
<i>User Impressions of Questions to Acquire Lexical Knowledge</i>	147
Kazunori Komatani and Mikio Nakano	147
<i>Simulating Turn-Taking in Conversations with Delayed Transmission</i>	157
Thilo Michael and Sebastian Möller	157
<i>Is this Dialogue Coherent? Learning from Dialogue Acts and Entities</i>	162
Alessandra Cervone and Giuseppe Riccardi	162
<i>Analyzing Speaker Strategy in Referential Communication</i>	175
Brian McMahan and Matthew Stone	175
<i>Contextualized Emotion Recognition in Conversation as Sequence Tagging</i>	186
Yan Wang, Jiayu Zhang, Jun Ma, Shaojun Wang and Jing Xiao	186
<i>How Self-Attention Improves Rare Class Performance in a Question-Answering Dialogue Agent</i>	196
Adam Stiff, Qi Song and Eric Fosler-Lussier	196
<i>Filtering conversations through dialogue acts labels for improving corpus-based convergence studies</i>	203
Simone Fuscone, Benoit Favre and Laurent Prévot	203
<i>Nontrivial Lexical Convergence in a Geography-Themed Game</i>	209
Amanda Bergqvist, Ramesh Manuvinakurike, Deepthi Karkada and Maike Paetzl	209
<i>A unifying framework for modeling acoustic/prosodic entrainment: definition and evaluation on two large corpora</i>	215
Ramiro H. Gálvez, Lara Gauder, Jordi Luque and Agustín Gravano	215
<i>Unsupervised Evaluation of Interactive Dialog with DialoGPT</i>	225
Shikib Mehri and Maxine Eskenazi	225
<i>Towards Unified Dialogue System Evaluation: A Comprehensive Analysis of Current Evaluation Protocols</i>	236
Sarah E. Finch and Jinho D. Choi	236
<i>Human-Human Health Coaching via Text Messages: Corpus, Annotation, and Analysis</i>	246
Itika Gupta, Barbara Di Eugenio, Brian Ziebart, Aiswarya Baiju, Bing Liu, Ben Gerber, Lisa Sharp, Nadia Nabulsi and Mary Smart	246
<i>Agent-Based Dynamic Collaboration Support in a Smart Office Space</i>	257
Yansen Wang, R. Charles Murray, Haogang Bao and Carolyn Rose	257
<i>Emora STDM: A Versatile Framework for Innovative Dialogue System Development</i>	261
James D. Finch and Jinho D. Choi	261
<i>Boosting Naturalness of Language in Task-oriented Dialogues via Adversarial Training</i>	265
Chenguang Zhu	265

<i>A Sequence-to-sequence Approach for Numerical Slot-filling Dialog Systems</i>	272
Hongjie Shi	272
<i>Beyond Domain APIs: Task-oriented Conversational Modeling with Unstructured Knowledge Access</i>	278
Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu and Dilek Hakkani-Tur	278
<i>Multi-Action Dialog Policy Learning with Interactive Human Teaching</i>	290
Megha Jhunjhunwala, Caleb Bryant and Pararth Shah	290
<i>Is Your Goal-Oriented Dialog Model Performing Really Well? Empirical Analysis of System-wise Evaluation</i>	297
Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao and Minlie Huang	297
<i>Similarity Scoring for Dialogue Behaviour Comparison</i>	311
Stefan Ultes and Wolfgang Maier	311
<i>Collection and Analysis of Dialogues Provided by Two Speakers Acting as One</i>	323
Tsunehiro Arimoto, Ryuichiro Higashinaka, Kou Tanaka, Takahito Kawanishi, Hiroaki Sugiyama, Hiroshi Sawada and Hiroshi Ishiguro	323
<i>Adaptive Dialog Policy Learning with Hindsight and User Modeling</i>	329
Yan Cao, Keting Lu, Xiaoping Chen and Shiqi Zhang	329
<i>Dialogue Policies for Learning Board Games through Multimodal Communication</i>	339
Maryam Zare, Ali Ayub, Aishan Liu, Sweekar Sudhakara, Alan Wagner and Rebecca Passonneau	339

Conference Program

Note that all shown times are GMT-6. Please adjust the times for your local time zone.
All presentations are pre-recorded unless stated otherwise.

Wednesday, 1 July 2020

7:30–7:45 *Opening Remarks*

7:45–8:30 *Keynote 1: Conversational Turn-taking in Human-robot Interaction*
Gabriel Skantze

8:30–9:00 *Keynote 1: live QA*
Gabriel Skantze

9:00–9:30 *Break*

9:30–10:20 *Generation + Task-Oriented Dialogues (1)*

Semantic Guidance of Dialogue Generation with Reinforcement Learning
Cheng-Hsun Hsueh and Wei-Yun Ma

Counseling-Style Reflection Generation Using Generative Pretrained Transformers with Augmented Context
Siqi Shen, Charles Welch, Rada Mihalcea and Verónica Pérez-Rosas

Learning from Mistakes: Combining Ontologies via Self-Training for Dialogue Generation
Lena Reed, Vrindavan Harrison, Shereen Oraby, Dilek Hakkani-Tur and Marilyn Walker

TripPy: A Triple Copy Strategy for Value Independent Neural Dialog State Tracking
Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi and Milica Gasic

Wednesday, 1 July 2020 (continued)

10:20–10:45 Generation + Task-Oriented Dialogues (1) live QA

10:45–11:30 Demo (1) pre-recorded presentations + live QA

Conversational Agents for Intelligent Buildings

Weronika Sieńska, Christian Dondrup, Nancie Gunson and Oliver Lemon

Retico: An incremental framework for spoken dialogue systems

Thilo Michael

MC-Saar-Instruct: a Platform for Minecraft Instruction Giving Agents

Arne Köhn, Julia Wichlacz, Christine Schäfer, Álvaro Torralba, Joerg Hoffmann and Alexander Koller

ConvoKit: A Toolkit for the Analysis of Conversations

Jonathan P. Chang, Caleb Chiam, Liye Fu, Andrew Wang, Justine Zhang and Cristian Danescu-Niculescu-Mizil

11:30–11:50 Break

11:50–12:20 Knowledge Acquisition/Use and Behaviour Modeling (1)

Commonsense Evidence Generation and Injection in Reading Comprehension

Ye Liu, Tao Yang, Zeyu You, Wei Fan and Philip S. Yu

Identifying Collaborative Conversations using Latent Discourse Behaviors

Ayush Jain, Maria Pacheco, Steven Lancette, Mahak Goindani and Dan Goldwasser

A Case Study of User Communication Styles with Customer Service Agents versus Intelligent Virtual Agents

Timothy Hewitt and Ian Beaver

Wednesday, 1 July 2020 (continued)

12:20–12:40 Knowledge Acquisition/Use and Behaviour Modeling (1) live QA

12:40–15:30 Breakout Discussion Sessions

15:30–16:00 Break

16:00–17:00 Special Session: Greetings and Talks

It's About Time: Turn-Entry Timing For Situated Human-Robot Dialogue
Felix Gervits, Ravenna Thielstrom, Antonio Roque and Matthias Scheutz

Learning Word Groundings from Humans Facilitated by Robot Emotional Displays
David McNeill and Casey Kennington

Learning and Reasoning for Robot Dialog and Navigation Tasks
Keting Lu, Shiqi Zhang, Peter Stone and Xiaoping Chen

An Attentive Listening System with Android ERICA: Comparison of Autonomous and WOZ Interactions
Koji Inoue, Divesh Lala, Kenta Yamamoto, Shizuka Nakamura, Katsuya Takanashi and Tatsuya Kawahara

17:00–17:30 Demo (2) pre-recorded presentations + live QA

A Spoken Dialogue System for Spatial Question Answering in a Physical Blocks World
Georgiy Platonov, Lenhart Schubert, Benjamin Kane and Aaron Gindi

rrSDS: Towards a Robot-ready Spoken Dialogue System
Casey Kennington, Daniele Moro, Lucas Marchand, Jake Carns and David McNeill

Wednesday, 1 July 2020 (continued)

17:30–18:15 Special Session: live QA

18:15–19:15 Special Session: Late-breaking

19:15–19:30 Break

19:30–20:15 Breakouts Discussion Sessions

Thursday, 2 July 2020

5:00–5:35 Knowledge Acquisition/Use and Behaviour Modeling (2)

Discovering Knowledge Graph Schema from Short Natural Language Text via Dialog

Subhasis Ghosh, Arpita Kundu, Aniket Pramanick and Indrajit Bhattacharya

User Impressions of Questions to Acquire Lexical Knowledge

Kazunori Komatani and Mikio Nakano

Simulating Turn-Taking in Conversations with Delayed Transmission

Thilo Michael and Sebastian Möller

Thursday, 2 July 2020 (continued)

5:35–6:00 Knowledge Acquisition/Use and Behaviour Modeling (2) live QA

6:00–6:45 Semantic and Pragmatics Modeling

Is this Dialogue Coherent? Learning from Dialogue Acts and Entities
Alessandra Cervone and Giuseppe Riccardi

Analyzing Speaker Strategy in Referential Communication
Brian McMahan and Matthew Stone

Contextualized Emotion Recognition in Conversation as Sequence Tagging
Yan Wang, Jiayu Zhang, Jun Ma, Shaojun Wang and Jing Xiao

How Self-Attention Improves Rare Class Performance in a Question-Answering Dialogue Agent
Adam Stiff, Qi Song and Eric Fosler-Lussier

6:45–7:10 Semantic and Pragmatics Modeling live QA

7:10–7:30 Break

7:30–8:00 Modeling Convergence

Filtering conversations through dialogue acts labels for improving corpus-based convergence studies
Simone Fuscone, Benoit Favre and Laurent Prévot

Nontrivial Lexical Convergence in a Geography-Themed Game
Amanda Bergqvist, Ramesh Manuvinakurike, Deepthi Karkada and Maike Paetzel

A unifying framework for modeling acoustic/prosodic entrainment: definition and evaluation on two large corpora
Ramiro H. Gálvez, Lara Gauder, Jordi Luque and Agustín Gravano

Thursday, 2 July 2020 (continued)

8:00–8:20 Modeling Convergence live QA

8:20–8:45 Break

8:45–9:30 Sponsor Booth

9:30–10:15 Breakout Discussion Sessions

10:15–10:45 Break

10:45–11:25 Evaluation + Corpora (1)

Unsupervised Evaluation of Interactive Dialog with DialoGPT

Shikib Mehri and Maxine Eskenazi

Towards Unified Dialogue System Evaluation: A Comprehensive Analysis of Current Evaluation Protocols

Sarah E. Finch and Jinho D. Choi

Human-Human Health Coaching via Text Messages: Corpus, Annotation, and Analysis

Itika Gupta, Barbara Di Eugenio, Brian Ziebart, Aiswarya Baiju, Bing Liu, Ben Gerber, Lisa Sharp, Nadia Nabulsi and Mary Smart

Thursday, 2 July 2020 (continued)

11:25–11:50 Evaluation + Corpora (1) live QA

11:50–12:30 Break

12:30–13:00 Demo (3) pre-recorded presentations + live QA

Agent-Based Dynamic Collaboration Support in a Smart Office Space

Yansen Wang, R. Charles Murray, Haogang Bao and Carolyn Rose

Emora STDM: A Versatile Framework for Innovative Dialogue System Development

James D. Finch and Jinho D. Choi

13:00–16:30 Breakout Discussion Sessions

16:30–17:00 Break

17:00–17:45 *Keynote 2: Neural Text Generation: Progress and Challenges*

Asli Celikyilmaz

17:45–18:15 *Keynote 2: live QA*

Asli Celikyilmaz

18:15–18:35 Break

Thursday, 2 July 2020 (continued)

18:35–19:15 Generation + Task-Oriented Dialogues (2)

Boosting Naturalness of Language in Task-oriented Dialogues via Adversarial Training

Chenguang Zhu

A Sequence-to-sequence Approach for Numerical Slot-filling Dialog Systems

Hongjie Shi

Beyond Domain APIs: Task-oriented Conversational Modeling with Unstructured Knowledge Access

Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu and Dilek Hakkani-Tur

Multi-Action Dialog Policy Learning with Interactive Human Teaching

Megha Jhunjhunwala, Caleb Bryant and Pararth Shah

19:15–19:40 Generation + Task-Oriented Dialogues (2) live QA

Friday, 3 July 2020

5:30–6:05 Evaluation + Corpora (2)

Is Your Goal-Oriented Dialog Model Performing Really Well? Empirical Analysis of System-wise Evaluation

Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao and Minlie Huang

Similarity Scoring for Dialogue Behaviour Comparison

Stefan Ultes and Wolfgang Maier

Collection and Analysis of Dialogues Provided by Two Speakers Acting as One

Tsunehiro Arimoto, Ryuichiro Higashinaka, Kou Tanaka, Takahito Kawanishi, Hiroaki Sugiyama, Hiroshi Sawada and Hiroshi Ishiguro

Friday, 3 July 2020 (continued)

6:05–6:30 Evaluation + Corpora (2) live QA

6:30–6:55 Dialogue Policy

Adaptive Dialog Policy Learning with Hindsight and User Modeling

Yan Cao, Keting Lu, Xiaoping Chen and Shiqi Zhang

Dialogue Policies for Learning Board Games through Multimodal Communication

Maryam Zare, Ali Ayub, Aishan Liu, Sweekar Sudhakara, Alan Wagner and Rebecca Passonneau

6:55–7:10 Dialogue Policy live QA

7:10–7:30 Break

7:30–8:15 Keynote 3: Argument Mining, Discourse Analysis, and Educational Applications

Diane Litman

8:15–8:45 Keynote 3: live QA

Diane Litman

8:45–9:45 Business Meeting, Awards, Closing (live)

Keynote Abstracts

Keynote 1 - Conversational Turn-taking in Human-robot Interaction

Gabriel Skantze

KTH Royal Institute of Technologies

Abstract

The last decade has seen a breakthrough for speech interfaces, much thanks to the advancements in speech recognition. Apart from voice assistants in smart speakers and phones, an emerging application area are social robots, which are expected to serve as receptionists, teachers, companions, coworkers, etc. Just like we prefer physical meetings over phone calls and video conferencing, social robots can potentially offer a much richer interaction experience than non-embodied dialogue systems. One example of this is the Furhat robot head, which started as a research project at KTH, but is now used in commercial applications, such as serving as a concierge at airports and performing job interviews. However, even though this recent progress is very exciting, current dialogue systems are still limited in several ways, especially for human-robot interaction. In this talk, I will specifically address the modelling of conversational turn-taking. As current systems lack the sophisticated coordination mechanisms found in human-human interaction, they are often plagued by interruptions or sluggish responses. In a face-to-face conversation, we use various multi-modal signals for this coordination, including linguistic and prosodic cues, as well as gaze and gestures. I will present our work on the use of deep learning for modelling these cues, which can allow the system to predict, and even project, potential turn-shifts. I will also present user studies which show how the robot can regulate turn-taking in multi-party dialogue by employing various turn-taking signals. This can be used to both facilitate a smoother interaction, as well as shaping the turn-taking dynamics and participation equality in multi-party settings.

Biography

Gabriel Skantze is professor in speech technology with a specialization in dialogue systems at KTH Royal Institute of Technology. His research focuses on the development of computational models for situated dialogue and human-robot interaction. He is also co-founder and chief scientist at Furhat Robotics, a startup based in Stockholm developing a platform for social robotics. Since 2019, he is the president of SIGdial.

Keynote 2 - Neural Text Generation: Progress and Challenges

Asli Celikyilmaz

Microsoft Research

Abstract

Automatic text generation enables computers to summarize text, describe pictures to visually impaired, write stories or articles about an event, have conversations in customer-service, chit-chat with individuals, and other settings, etc. Neural text generation – using neural network models to generate coherent text – have seen a paradigm shift in the last years, caused by the advances in deep contextual language modeling (e.g., LSTMs, GPT) and transfer learning (e.g., ELMO, BERT). While these tools have dramatically improved the state of text generation, particularly for low resource tasks, state-of-the-art neural text generation models still face many challenges: a lack of diversity in generated text, commonsense violations in depicted situations, difficulties in making use of multi-modal input, and many more. I will discuss existing technology to generate text with better discourse structure, narrative flow, or one that can use world knowledge more intelligently. I will conclude the talk with a discussion of current challenges and shortcomings of neural text generation, pointing to avenues for future research.

Biography

Asli Celikyilmaz is a Principal Researcher at Microsoft Research (MSR) in Redmond, Washington. She is also an Affiliate Professor at the University of Washington. She has received Ph.D. Degree in Information Science from University of Toronto, Canada, and later continued her Postdoc study at Computer Science Department of the University of California, Berkeley. Her research interests are mainly in deep learning and natural language, specifically on language generation with long-term coherence, language understanding, language grounding with vision, and building intelligent agents for human-computer interaction. She is serving on the editorial boards of Transactions of the ACL (TACL) as area editor and Open Journal of Signal Processing (OJSP) as Associate Editor. She has received several “best of” awards including NAFIPS 2007, Semantic Computing 2009, and CVPR 2019.

Keynote 3 - Argument Mining, Discourse Analysis, and Educational Applications

Diane Litman

University of Pittsburgh

Abstract

The written and spoken arguments of students are educational data that can be automatically mined for purposes such as student assessment or teacher professional development. This talk will illustrate some of the opportunities and challenges in educationally-oriented argument mining. I will first describe how we are using discourse analysis to improve argument mining systems that are being embedded in educational technologies for essay grading and for analyzing classroom discussions. I will then present intrinsic and extrinsic evaluation results for two of our argument mining systems, using benchmark persuasive essay corpora as well as our recently released Discussion Tracker corpus of collaborative argumentation in high school classrooms.

Biography

Diane Litman is Professor of Computer Science, Senior Scientist with the Learning Research and Development Center, and Faculty Co-Director of the Graduate Program in Intelligent Systems, all at the University of Pittsburgh. Her current research focuses on enhancing the effectiveness of educational technology through the use of spoken and natural language processing techniques such as argument mining, summarization, multi-party dialogue systems, and revision analysis. She is a Fellow of the Association for Computational Linguistics, has twice been elected Chair of the North American Chapter of the Association for Computational Linguistics, has co-authored multiple papers winning best paper awards, and was the SIGdial Program Co-Chair in 2018.

Semantic Guidance of Dialogue Generation with Reinforcement Learning

Cheng-Hsun Hsueh
National Yang-Ming University
jimbokururu27@gmail.com

Wei-Yun Ma
Academia Sinica
ma@iis.sinica.edu.tw

Abstract

Neural encoder-decoder models have shown promising performance for human-computer dialogue systems over the past few years. However, due to the maximum-likelihood objective for the decoder, the generated responses are often universal and safe to the point that they lack meaningful information and are no longer relevant to the post. To address this, in this paper, we propose semantic guidance using reinforcement learning to ensure that the generated responses indeed include the given or predicted semantics and that these semantics do not appear repeatedly in the response. Synsets, which comprise sets of manually defined synonyms, are used as the form of assigned semantics. For a given/assigned/predicted synset, only one of its synonyms should appear in the generated response; this constitutes a simple but effective semantic-control mechanism. We conduct both quantitative and qualitative evaluations, which show that the generated responses are not only higher-quality but also reflect the assigned semantic controls.

1 Introduction

Dialogue generation systems with adequate artificial intelligence responses hold great potential for practical use. A decent human-computer dialogue system should generate coherent and informative responses based on human-provided posts (Li et al., 2017). Sequence-to-sequence models (Sutskever et al., 2014) with long-short term memory (Hochreiter and Schmidhuber, 1997) or gated recurrent networks (Cho et al., 2014) have demonstrated profound improvements in open-domain dialogue systems (Shang et al., 2015; Vinyals and Le, 2015; Luan et al., 2016; Xu et al., 2016; Yao et al., 2017). However, these models often generate overly generic responses (Sordoni et al., 2015; Li et al., 2016a) that are independent of the given posts

due to the maximum-likelihood-estimation-based objectives.

To improve the variety of the responses, recent studies usually use semantically conditioned LSTM, relying on additional semantic indicators such as keywords to guide the decoding process. However, keywords typically appear repeatedly in generated utterances with this strategy. To address this, Wen et al. (2015b) propose a special gate mechanism to reduce the influence of the keywords. However, since this design does not directly address the concern in the objectives, repeated keywords still often remain a problem in practice; we confirm this is in our experiments.

To address this issue, in this paper, we introduce the semantically controlled and recorded LSTM (SCR-LSTM) cell, which provides semantic guidance via reinforcement learning (RL) as well as a recording mechanism that records the existence of the desired semantics to ensure that the generated responses indeed include the given or predicted semantics; also, the desired semantics are not to appear repeatedly in the response. For the form of the assigned semantics we use synsets, which provide a more flexible semantic representation for practical use, and any lexical or knowledge taxonomy can be used to serve this role. For a given/assigned/predicted synset, only one of its covering synonyms should appear in the generated response.

In addition, when synsets are used to semantically control the generated responses, the responses may indeed show the assigned semantics, but the responses could be not diverse enough, or the relation to the given posts may be tenuous, because the major goal of the model is to meet the semantic constraints. Therefore, we add a conditional SeqGAN (Yu et al., 2017) to assure that the generated responses are similar to true human responses and are related to the given posts while specifying

semantics to avoid dull or repetitive responses.

As with conventional GAN (Goodfellow et al., 2014), our conditional SeqGAN comprises a generator and a discriminator; however, with the proposed discriminator we seek to not only distinguish machine-generated utterances from human-generated utterances but also distinguish post-independent from post-dependent utterances. The resulting additional SeqGAN architecture generates responses that are more related to the posts.

2 Background

2.1 Semantically conditioned LSTM

To incorporate given dialogue acts into utterance generation, Wen et al. (2015b) propose the semantic controlled LSTM (SC LSTM) cell, a special neural cell. The assigned dialogue acts are represented in one-hot form, and are fed into dialogue acts cells, which rely on a decreasing mechanism on dialogue acts information to avoid repetition. The formula for this semantically conditioned LSTM is as following:

$$i_t = \sigma(W_{wi}w_t + W_{hi}h_{t-1}) \quad (1)$$

$$f_t = \sigma(W_{wf}w_t + W_{hf}h_{t-1}) \quad (2)$$

$$o_t = \sigma(W_{wo}w_t + W_{ho}h_{t-1}) \quad (3)$$

$$\hat{c}_t = \tanh(W_{wc}w_t + W_{hc}h_{t-1}) \quad (4)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \hat{c}_t + \tanh(W_{dc}d_t) \quad (5)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (6)$$

With its additional third term, only formula (5) of cell value c_t differs from traditional LSTM. Term d_t serves as the dialogue act one-hot vector, and is derived from the following formula:

$$r_t = \sigma(W_{wr}w_t + \sum_l \alpha_l W_{hr}^l h_{t-1}^l) \quad (7)$$

$$d_t = r_t \otimes d_{t-1} \quad (8)$$

Wen et al. (2015b) term the mechanism based on (7) and (8) a *dialogue act cell* (DA cell). Vector r_t , known as the reading gate, is determined by the input of the current time step and the hidden state of the past generation history, and is multiplied element-wise with the dialogue act vector d_t to either retain or discard its information in future generation time steps.

The monotonically decreasing value of the dialogue act vector is intended to reduce repetition.

However, the design provides an insufficient guarantee on avoiding repetition, as the model provides no direct link between the dialogue act generation possibility and the value of d_t ; thus repeated keywords continue to remain a problem in practice.

2.2 Sequence GAN

The original generative adversarial network (GAN) is ill-suited to text generation given the discrete nature of text. In particular, the changing-signal guidance from the discriminator does not correspond to discrete dictionary tokens (Yu et al., 2017). Furthermore, the rewards can only be given to entire sequences when the whole generation is finished, making it impossible to estimate the value of a specific token in the generation step. Sequence GAN introduces a policy gradient (Sutton et al., 1999) as well as a rollout mechanism to help the discriminator pass its scores to the generator.

Given a current and incompletely generated response $Y_{1:t} = [y_1, y_2, y_3, \dots, y_t]$, where t is the current time step of generation and y_t is the token generated at the current step, a reward is to be given to the current token y_t . However, these rewards can be estimated only once the entire sequence has been generated. To account for this, the generator must “roll out” the complete responses at every current step. For example, if we roll out starting from time step t , the complete utterance can be generated using Monte Carlo search as

$$Y_{1:T}^n \in MC^G(Y_{1:t}; N) \quad (9)$$

where MC denotes Monte Carlo search, G denotes the generator, and N denotes the assigned repeating turn for searching. The incomplete responses are completed after the rollout and then judged by the discriminator, which assigns reward scores to the rollouted responses. Rollout is accomplished using N Monte Carlo searches, and the rewards are averaged to serve as the expected utility for the incomplete utterance generated at time step t :

$$V(Y_{1:t}) = \frac{1}{N} \sum_{n=1}^N D_\phi(Y_{1:T}^n) \quad (10)$$

where $D_\phi(Y_{1:T}^n)$ denotes the score assigned by the discriminator.

2.3 Conditional GAN

Unconditioned GAN loses control on generating the intended type of data. By giving conditions for

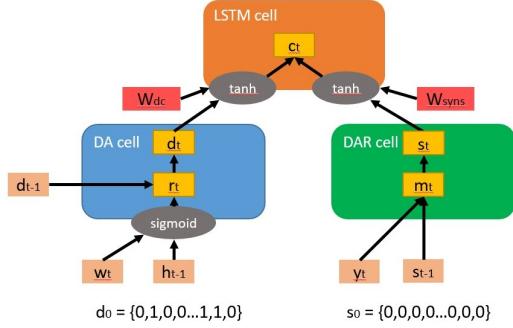


Figure 1: Proposed semantically controlled and recorded LSTM (SCR LSTM) cell.

GAN to depend on (Mirza and Osindero, 2014), it is possible to guide the generation process. Conditional GAN extends the original GAN by providing extra information y for both the generator and the discriminator. The generator conditions on y , whereas the discriminator judges whether the generated data is suitable based on the relatedness of the generated results and the extra information y . Thus, the formula for conditional GAN extends the original GAN with y to become

$$\begin{aligned} \min_G \max_D V(D, G) &= E_{x \sim P_{data}(x)}[\log D(x | \mathbf{y})] \\ &+ E_{z \sim P_z(z)}[\log(1 - D(G(z | \mathbf{y})))] \quad (11) \end{aligned}$$

where z denotes the generated data and x denotes the training data.

3 Methods

3.1 Semantically Controlled and Recorded LSTM Cell

Extended from (Wen et al. 2015b), we introduce the semantically controlled and recorded LSTM (SCR LSTM) cell, which provides semantic guidance and a recording mechanism, as shown in Figure 1. It integrates a DA cell with a synonym act and a special recording cell which we propose to provide a mechanism to record the existence of the desired semantics. We term this a *dialog act record cell* (DAR cell).

3.2 DA Cell with Synonym Act

The DA cell (Wen et al. 2015b) is integrated in our SCR LSTM cell, but here we slightly change the definition of *act*. We define an act as an element (synonym) of the assigned synset; we expect that just one of the acts (synonyms) will be used in the generation. A one-hot vector is used to encode this synonym act, denoted by d_t , where each element

corresponds to a word in the vocabulary, and it is assigned 1 if the corresponding word belongs to the assigned synset. For example, in Figure 2, given an assigned synset - synset_{*i*}, there are three vocabulary elements (synonyms)—‘tribe’, ‘group’, and ‘clan’—thus the vocabulary-size d_t is represented as [0..0, 1, 0..0, 1, 0..0, 1, 0..0], in which the three 1s refer to ‘tribe’, ‘group’, and ‘clan’, respectively. Value d_t is fed into the DA cell, which relies on a decreasing mechanism for dialogue act information to prevent repetition, as shown in (7) and (8).

Note that although the three synonyms are all indexed in d_t , this does not mean that all three synonyms (dialogue acts) are to appear in the response. Instead, we expect only one of these to appear, in fact to appear exactly once, in contrast to (Wen et al. 2015b). However, the DA cell merely decays the influence of the assigned dialogue acts, and does not directly address this concern in the objectives; thus repeated keywords still remain a problem in practice, as we verify experimentally. To address this shortcoming, we propose the dialog act record cell (DAR cell) in concert with the DA cell.

3.3 DAR Cell with Synset Act

With the DAR cell we seek to provide a mechanism to record the occurrence of the desired semantics¹ to ensure that they are indeed included in the generated responses. At every generation time step we use a one-hot vector s_t , the dimension of which is the total number of synsets, to record whether the assigned synsets appear. Each element of s_t indicates whether the synset appears or not. For generation, s_t is initialized as [0..0]. Once an element of the assigned synset appears during generation, the synset’s corresponding element in s_t is changed to 1. We develop a special gate called an *MGate* to realize this function, which is formally presented in Algorithm 1.

Figure 2 illustrates the overall structure. Given an assigned synset - synset_{*i*}, there are three vocabulary elements (synonyms): ‘tribe’, ‘group’, and ‘clan’. As generation proceeds, at the second time step, as ‘tribe’ is generated, s_t is updated from [0..0,0,0..0] to [0..0,1,0..0], in which 1 refers to synset_{*i*}’s current status. The updated s_t informs the model that synset_{*i*} has already appeared, instructing it to not generate any element of synset_{*i*} afterward.

¹In our model, the desired semantics can be multiple synsets or a single synset. All of our experiments are based on a single synset.

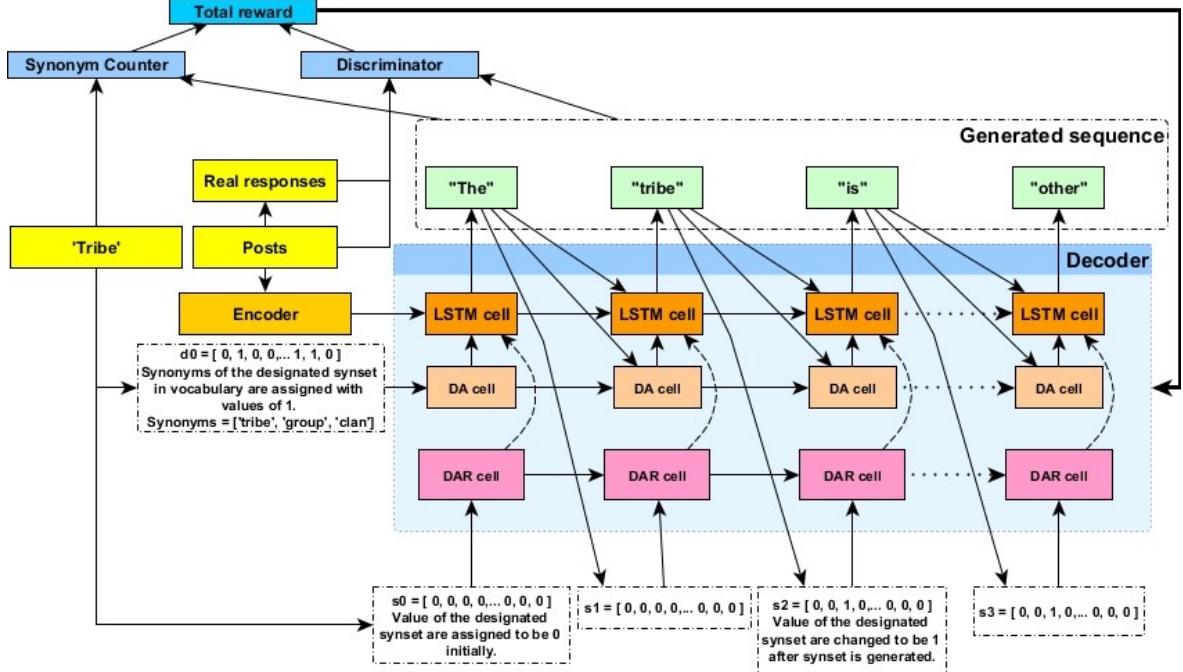


Figure 2: Overall view of the model. The decoder incorporates synset information through additional DAR cells, which retain or discard synset information in every generation step based on whether the assigned synset has appeared or not.

Algorithm 1 M Gate Algorithm.

Input: s_{t-1} and y_{t-1} (y refers to generated token)

Output: s_t

- 1: $s_t = s_{t-1}$
 - 2: **for** each synset $_i \in$ assigned synset **do**
 - 3: **if** $y_{t-1} \in$ synset $_i$ **then**
 - 4: $s_t(i) = 1$ // $s_t(i)$ refers to i -th element of s_t
 - 5: **end if**
 - 6: **end for**
-

The SCR LSTM cell value c_t integrating the DA cell and DAR cell is

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \hat{c}_t + \tanh(W_{dc}d_t) + \tanh(W_{syns}s_t) \quad (12)$$

where $W_{dc}d_t$ and $W_{syns}s_t$ are the outputs of the DA and DAR cell, respectively, and $W_{dc} \in R^{h \times d_1}$, $W_{syns} \in R^{h \times d_2}$, $d_t \in \{0, 1\}^{d_1}$, and $s_t \in \{0, 1\}^{d_2}$. Value d_1 denotes the vocabulary size, d_2 denotes the total number of synsets, and h denotes the dimension of hidden states in the decoder.

To both prevent repetition and ensure the desired semantics in the generated responses, we use reinforcement learning to penalize our model for

violations. The reward is

$$C_{syn} = 1 - |\text{semantic occurrence} - 1| \quad (13)$$

where semantic occurrence is an integer that records the current occurrence (frequency) of the elements of the assigned synset at every time step of the generation. Thus we expect that when the generation is finished, semantic occurrence will be exactly 1 instead of a number greater than 1, indicating repetition of the desired semantics, or 0, indicating the absence of the desired semantics.

Thus only a semantic occurrence of 1 results in the highest value of 1 for C^{syn} ; a semantic occurrence of 0 and a semantic occurrence greater than 1 cause C^{syn} to be less than or equal to 0.

Although this reward encourages appropriate appearances of the assigned synsets in the response, it could cause the model ignore other critical requirements for a response, including fluency, relevance to the posts, and information. To account for this, we add a conditional SeqGAN to provide another reward D_ϕ , which is the result of its discriminator, seeking to ensure that the generated responses approximate true human responses and are related to the given posts.

The discriminator not only distinguishes machine-generated utterances from human-generated utterances but also distinguishes

post-independent from post-dependent utterances. D_ϕ derives its score by projecting the concatenated final hidden states of two LSTM sequence-to-sequence networks to a 2-dimensional vector followed by softmax. The first LSTM network is given posts as encoder inputs and responses as decoder inputs, whereas the second network switches posts and responses. Therefore, the discriminator model can be formulated as

$$D_\phi(p, q) = \text{softmax} \left(W^D [h_1^{final}_{p|q}; h_2^{final}_{q|p}] \right) \quad (14)$$

where p denotes post, q denotes response, W^D denotes the projection matrix, and h_1 and h_2 denote two sequence-to-sequence networks respectively. During training, a third of the training batches are pairs composed of posts with their correlated human responses, another third is composed of pairs of posts with an uncorrelated human response, and the final third is pairs of posts with a generated response. Only the first third is labeled *true*; the other two-thirds are labeled *false*.

For every generation step, the expected utility V is given by both the semantic occurrence and the discriminator, calculated using Monte Carlo search as

$$V(p, Y_{1:t}) = \frac{1}{N} \sum_{n=1}^N D_\phi(p, Y_{1:T}^n) + C^{syn}(Y_{1:T}^n), \\ Y_{1:T}^n \in MC^G(Y_{1:t}, N) \quad (15)$$

where the notation p denotes the post, $Y_{1:t} = [y_1, y_2, y_3, \dots, y_t]$ denotes the generated sequence, and G denotes the generator. N is the number of turns in the Monte Carlo search, here set to 5. The utility is then applied in the REINFORCE algorithm (Williams, 1992) as

$$\nabla J(\theta) \approx \sum_t (V(p, Y_{1:t}) - b) \nabla \log p(y_t | x, Y_{1:t-1}) \quad (16)$$

where b denotes the baseline value to reduce the variance of the utility.

4 Evaluation

4.1 Dataset

Conversation data from Weibo was used for training and evaluation. The training data is composed of 570k post-response pairs with 3360 synonym

Algorithm 2 Training Algorithm.

Input: (post, response) pairs with assigned synsets

- 1: Initialize generator and discriminator
- 2: Pre-train generator G using maximum likelihood estimation
- 3: **repeat**
- 4: Generator G generates response $Y_{1:T}$ given post and assigned synset
- 5: **for** $t \in \text{range}(T)$ **do** // N is turns of MC search
- 6: $s_t \leftarrow \text{M-Gate}(y_t^n, s_{t-1})$
- 7: Roll out $Y_{1:t}^n$ to full sentence $Y_{1:T}^n$
- 8: **end for**
- 9: Calculated the expected utility of $Y_{1:t}$ by equation(15)
- 10: **end for**
- 11: Update generator G
- 12: Update discriminator D
- 13: **until** reinforcement learning converges

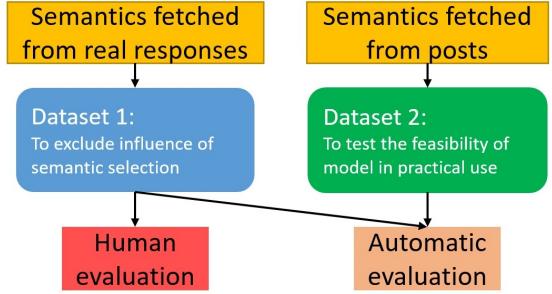


Figure 3: The two datasets in experiment.

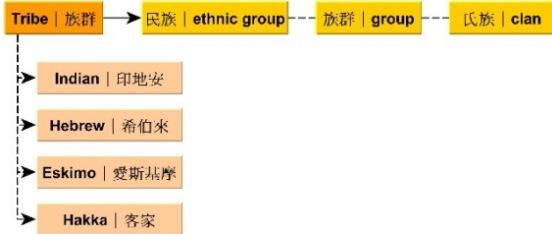


Figure 4: Structure of E-HowNet.

sets, and the testing data is composed of 2k post-response pairs with 1731 synonym sets.

Here we established two datasets, as shown in Figure 3. In the first dataset, we attempted to eliminate interference from semantic selection and focus mainly on the effect of the model. Therefore, we fetched the assigned semantics from human response by randomly selecting one synset from the human response. In the first dataset, we used both human evaluation and automatic evaluation to analyze the efficacy of our model. Thereafter

we switched to the second dataset, where the assigned semantics are simply fetched from posts by randomly selecting one synset from a post. We analyze the feasibility of our model in practical use. Automatic evaluations are also performed for the second dataset.

The synonyms of an assigned synset are retrieved from E-hownet (Ma and Shih, 2018; Chen et al. 2005), a structured Chinese knowledge net. The structure of E-hownet is shown in Figure 4. The synonyms of an assigned word are at the same level of the word, whereas meanings of a word are inferior to the word.

For the experiments here we fetch only the synonyms. Note that our model is not confined to E-hownet; other synonym datasets could be used for our proposed model as well.

4.2 Baselines

SEQ2SEQ

The Sequence-to-sequence model (Sutskever et al., 2014) with an attention mechanism (Bahdanau et al., 2014) is implemented without auxiliary keywords.

Hierarchical Gated Fusion Unit (HGFU)

HGFU (Yao et al., 2017) incorporates assigned keywords into sequence generation. We replace the keyword input with the synset to focus the comparison on the model design and ensure a fair comparison.

Semantically conditioned LSTM (SC-LSTM)

Wen et al. (2015b) use dialogue acts cells to generate utterances that contain the assigned dialogue acts. Here we replace the dialogue acts with synsets for comparison. In addition, for a full comparison, we implement SC-LSTM with over-generation, as suggested by Wen et al. 2015a, generating 20 sequences and selecting the top-scoring one.

4.3 Proposed method

SCR-LSTM + RL

This approach extends the former method using an RL mechanism and an additional DAR cell to record whether the synonym set has already been generated in previous generation steps (Section 3.1).

The proposed methods and baselines all leverage beam search with a beam size of 5 to generate appropriate responses. Only the top-scored sequences are selected for further evaluation.

Methods	Average score
SEQ2SEQ	0.99
HGFU	1.19
SC-LSTM (over generation)	1.06
SCR-LSTM+RL	1.23

Table 1: Scores of different models from human evaluation

Situation	Percentage
HGFU win	24.65%
Tie	42.71%
SCR-LSTM + RL win	32.64%

Table 2: Comparison between HGFU and SCR-LSTM + reinforcement learning

Methods	Repetition percentage	Non-appearing percentage
HGFU	22.99%	4.00%
SC-LSTM (over generation)	3.05%	5.55%
SCR-LSTM+RL	2.10%	0.25%

Table 3: Percentage of synonym repetition and non-appearing when semantics are fetched from responses

Methods	Repetition percentage	Non-appearing percentage
HGFU	31.03%	4.05%
SC-LSTM (over generation)	2.90%	5.55%
SC-LSTM (MLE)	13.10%	1.25%
SCR-LSTM+RL (w/o discriminator)	2.20%	0.50%
SCR-LSTM+RL (w/ discriminator)	4.15%	0.70%

Table 4: Percentage of synonym repetition and non-appearing when semantics are fetched from posts

Methods	Uni -gram	Bi -gram	Tri -gram
SEQ2SEQ	3.20%	10.81%	19.15%
HGFU	7.11%	23.99%	39.87%
SC-LSTM (over generation)	15.02%	61.47%	89.85%
SC-LSTM (MLE)	7.60%	22.33%	36.53%
SCR-LSTM+RL (w/o discriminator)	8.73%	25.10%	39.80%
SCR-LSTM+RL (w/ discriminator)	8.91%	26.17%	42.21%

Table 5: Unigram, bigram and trigram variations of different methods

Methods	Example 1	Example 2
	Post: 好...我抱着一丝期待等答案公布啊! Post: OK... I am anticipating the answer! Synset: 希望、盼、盼望、期待、期盼 (anticipate)	Post: 因为我不会画，所以就被丑化啦 Post: I am defamed because I can't draw Synset: 没法、不能、无法、不会、不可 (cannot)
SEQ2SEQ	好的，谢谢你的支持 Ok, thank you for your support	我画的是你画的 I drew what you drew
HGFU	你说的是什么? What are you saying?	不会吧 No way
SC-LSTM (over generation)	不用期待了吗? Not looking forward to it?	人生才不会造成吧? Life wouldn't cause that, would it?
SCR-LSTM (MLE)	好的，期待! Okay, looking forward to it!	不会吧，我也很喜欢 No, I like it too
SCR-LSTM + RL (w/o discriminator)	期待中! Looking forward to it!	不会吧 No way
SCR-LSTM + RL (w/ discriminator)	好的，敬请期待 OK, stay tuned	不会吧，我也觉得挺好看的 No, I think it's pretty good too

Table 6: Examples of different models

Methods	Example 1	Example 2
	Post: 是反话啊，很明显...哈哈 Post: It's ironic, obviously ... haha Synset: 向、是 (is)	Post: 哈哈，每道菜我都要品尝! Post: Haha, I will try every dish! Synset: 品尝、尝 (taste)
SCR-LSTM + RL (w/o discriminator)	是的，我也很喜欢他的 Yes, I like him too	尝了吧! Taste it!
SCR-LSTM + RL (w discriminator)	是啊，我也觉得很搞笑 Yeah, I also find it funny	欢迎您来品尝! You are welcome to have a taste!

Table 7: Examples SCR-LSTM with and without discriminator

Methods	SCR-LSTM (w/ discriminator)
Example 1	Post: 抚州娃发来贺电，南昌新年好 Post: Greetings from the baby in Fuzhou, happy new year in Nanchang Synset: 娃、小子、孩子、孩儿 (kids, baby) Response: 谢谢!孩子们! Response: Thanks, kids!
Example 2	Post: 讲什么的，育儿? Post: What is it about? Raising child? Synset: 说、说话、讲、讲话(say) Response: 你说的是什么 Response: What do you mean?

Table 8: Synsets help to extend semantics

4.4 Results and analysis

Human evaluation

Since automatic metrics such as the BLEU score or perplexity are not suitable in evaluating dialogue generation (Shang et al., 2015), we used human judgments instead. The criteria of human evaluation are referenced from Shang et al. (2015) with three levels: unsuitable, neutral, and suitable. To

be judged ‘suitable’, the response must be clearly correlated to the post and must be natural. For ‘neutral’, the response can be suitable in a specific scenario. The response is considered ‘unsuitable’ if it does not fit in any scenario provided by the post. Scores of 0, 1, and 2 were given for the three levels respectively. Four methods for comparison were evaluated, with 230 generated responses each. Every generated response was evaluated by three people using Amazon Turk. As mentioned above, the semantics for this part of data were fetched from real human responses.

Table 1 demonstrates that SCR-LSTM + RL receives the highest score and HGFU ranks second. To further compare the two methods, 96 posts and generated responses from the two methods were compared directly, with ties allowed. Table 2 shows that the proposed method still outperforms HGFU.

Also note that the proposed model outperforms SEQ2SEQ, which does not rely on extra semantic guidance, demonstrating that semantic guidance plays an important role in generating meaningful and related sequences given the post.

Automatic evaluation

To further evaluate the effect of the proposed model, we implemented automatic evaluations. We also calculate the percentage of semantic repetition and non-appearance. Table 3 shows that when semantics are fetched from human responses, SCR-LSTM + RL generates sequences with the least semantic repetition and absence. For dataset 1, both human evaluation and automatic evaluation prove that with semantic selection, the proposed model generates natural responses with the assigned semantics appearing only once.

To further evaluate the feasibility of our model in practical use, we shift to dataset 2, in which semantics are fetched from posts. We evaluate the effect of reinforcement learning and the discriminator, respectively, using three methods: SCR-LSTM trained with maximum-likelihood-estimation without RL (SCR-LSTM MLE), SCR-LSTM trained with synset occurrences during reinforcement learning but without the discriminator (SCR-LSTM + RL w/o discriminator), and SCR-LSTM trained with synset occurrences and the discriminator (SCR-LSTM + RL w/ discriminator), respectively.

We implement as automatic methods the percentage of semantic repetition and that of non-appearance. Table 4 shows that SCR-LSTM + RL both with and without discriminator methods generate less semantic repetition and absence than SCR-LSTM MLE. This shows that reinforcement learning with the target of single-appearance semantics has achieved its goal. SCR-LSTM+RL without the discriminator, which is trained using only synset occurrences as a reward, reduces semantic repetition and absence even more, resulting in the best performance in Table 4. In addition, SCR-LSTM MLE also results in significantly less semantic repetition and fewer absences than HGFU, showing that the proposed SCR-LSTM design alone is enough to induce the desired semantics to appear just once.

Another metric is the percentage of distinct unigrams, bigrams, and trigrams. Proposed by Li et al. 2016b, this quantifies the diversity of a generated sequence. This metric is calculated by counting the distinct unigrams, bigrams, and trigrams, and divided this by the total number of unigrams, bigrams, and trigrams respectively. Table 5 shows that SCR-LSTM + RL with the discriminator achieves higher distinct unigram, bigram, and trigram percentages than SCR-LSTM + RL without the discriminator. Thus the discriminator does help the reinforcement

learn to generate more diverse responses. Note that the over-generation of SC-LSTM yields the highest diversity because the model generates words randomly and thus has a higher possibility to pick up non-frequent words. Table 6 contains examples from different models.

Case study: Effect of the discriminator

The effect of the discriminator is seen in Table 7, which compares SCR-LSTM + RL with and without the discriminator. In the first example, SCR-LSTM + RL w/o discriminator generates a sequence that is not correlated with the given post. SCR-LSTM + RL w/ discriminator generates a better sequence that is relevant to the post. For the second example, both methods generate relevant sequences to the post, but SCR-LSTM + RL w/o discriminator generates a sequence that is too short and not very informative while the LSTM + RL w/ discriminator generates a sequence that is more meaningful and diverse.

Case study: Semantic coverage

With the synset implementation we seek to extend the semantic coverage of the desired keywords. In Table 8, keywords from posts are not directly used when generating responses. Instead, the synonyms of the keywords are used as extra information during the generation process. This shows that a particular synonym may be used as semantic guidance in generated responses.

5 Conclusion

In this work, to develop an effective semantic control mechanism, we propose the SCR-LSTM model with reinforcement learning to ensure that the desired semantics appear once and do not repeat. We also present a conditional SeqGAN to help generate more coherent and informative responses. Results from both human and automatic evaluations show that the proposed models outperform other baselines and achieve the lowest repetition and absence percentages of the assigned synsets in the generated responses, proving that the proposed approach indeed produces high-quality responses under the desired semantic control. Also, we prove that SeqGAN is an essential part of enabling the model to generate more diverse and coherent responses.

The proposed model leverages synsets to serve as the semantic guidance. To investigate the feasibility of our model in practical use, in this work, the assigned synsets are simply fetched from posts. However, the selection or prediction of the desired

semantics is an interesting task that we leave to future study.

References

- K. Cho, B.V. Merrienboer, Ç. Gülcöhre, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning phrase representations using rnn encoder- decoder for statistical machine translation. In *EMNLP*.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.
- S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. In *Neural Computation*, 9, pages 1735–1780.
- Chen Keh-Jiann, Huang Shu-Ling, Shih Yueh-Yin, , and Chen Yi-Jun. 2005. Extended-hownet - a representational framework for concepts. In *OntoLex 2005 - Ontologies and Lexical Resources IJCNLP-05 Workshop*.
- J. Li, M. Galley, C. Brockett, J. Gao, and W.B. Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *HLT- NAACL*.
- J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. In *EMNLP*.
- J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *EMNLP*.
- Y. Luan, Y. Ji, and M. Ostendorf. 2016. Lstm based conversation models. In *CoRR*.
- Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. In *CoRR*.
- L. Shang, Z. Lu, and H. Li. 2015. Neural responding machine for short-text conversation. In *ACL*.
- A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J. Nie, J. Gao, and W.B. Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *HLT- NAACL*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, page 3104–3112.
- R.S. Sutton, D.A. McAllester, S.P. Singh, and Y. Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.
- O. Vinyals and Q.V. Le. 2015. A neural conversational model. In *CoRR*.
- Ma Wei-Yun and Shih Yueh-Yin. 2018. Extended hownet 2.0 – an entity-relation common-sense representation model. In *LREC*.
- T. Wen, M. Gasic, D. Kim, N. Mrksic, P. Su, D. Vandyke, and S.J. Young. 2015a. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *EMNLP*.
- T. Wen, M. Gasic, D. Kim, N. Mrksic, P. Su, D. Vandyke, and S.J. Young. 2015b. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *SIGDIAL*.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning 8 (3-4)*, page 229–256.
- Z. Xu, B. Liu, B. Wang, C. Sun, and X. Wang. 2016. Incorporating loose-structured knowledge into lstm with recall gate for conversation modeling. In *CoRR*.
- L. Yao, Y. Zhang, Y. Feng, D. Zhao, , and R. Yan. 2017. Towards implicit content-introducing for generative short-text conversation systems. In *EMNLP*.
- L. Yu, W. Zhang, J. Wang, and Y. Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*.

Counseling-Style Reflection Generation Using Generative Pretrained Transformers with Augmented Context

Siqi Shen, Charles Welch, Rada Mihalcea, Verónica Pérez-Rosas

Department of Computer Science and Engineering,

University of Michigan

{shensq, cfwelch, mihalcea, vrncapr}@umich.edu

Abstract

In this paper, we introduce a counseling dialogue system that provides real-time assistance to counseling trainees. The system generates sample counselors' reflections – i.e., responses that reflect back on what the client has said given the dialogue history. We build our model upon the recent generative pretrained transformer architecture and leverage context augmentation techniques inspired by traditional strategies used during counselor training to further enhance its performance. We show that the system incorporating these strategies outperforms the baseline models on the reflection generation task on multiple metrics. To confirm our findings, we present a human evaluation study that shows that the output of the enhanced system obtains higher ratings and is on par with human responses in terms of stylistic and grammatical correctness, as well as context-awareness.

1 Introduction

A recent survey on mental and behavioral healthcare showed that while there is an increasing need for counseling services, the available mental health workforce is barely coping with this demand.¹ An important reason behind this unmet need is that the training of counselors requires a lot of time and effort. Typically, counselor training involves refining counseling skills through practice and feedback using role-play activities, simulated patients, or real patients, thus heavily relying on human supervision and interaction.

In clinician training, feedback and coaching can significantly improve the post-training counselor proficiency (Miller et al., 2004). However, the standard way of providing systematic feedback relies on human coding of the counseling sessions. This

process can take up to ten times as long as the duration of the session itself, and thus it does not scale up (Atkins et al., 2014).

Previous work has focused on developing automatic tools for counseling evaluation and training tasks, including automatic coding (i.e., recognizing a counselor behavior) and forecasting (i.e., predicting the most appropriate behavior for the next counselor's utterance) (Tanana et al., 2016; Park et al., 2019; Cao et al., 2019). These tools aim to facilitate the evaluation of a counseling encounter and, to some extent, provide generic guidance during the conversation. Although these systems help counselors by suggesting the timing of a certain counseling behavior, they do not offer any help on how to accomplish it.

Among the different skills to be learned by counselors, reflective listening has been shown to be an important skill related to positive therapeutic outcomes (Moyers et al., 2009). Reflective listening is a conversational strategy used by counselors to show that they understand their clients' perspectives, feelings, and values (Miller and Rollnick, 2013). During this process, the counselor listens to the client's statements and then makes a statement (reflection) that is a reasonable approximation of the meaning of what the client has said. Thus, the main role of reflections is to keep the conversation focused on the client and to move the conversation forward. For example, considering the following utterance by the client, a counselor could make reflections (a) or (b) to show an understanding of the client's feelings and concerns.

Client: I want to quit smoking because I don't want another heart attack; I want to see my kids grow up.

Counselor (a): You are scared that you might have another heart attack.

Counselor (b): It seems that you see a con-

¹<https://www.mhanational.org/issues/state-mental-health-america>

nection between your smoking and the possibility of having another heart attack.

Motivated by the importance of reflective listening skills and the significance of real-time feedback in the success of a counseling encounter, we envision our system as an automatic assistant that provides counselors with sample reflection language that is appropriate to the conversation context, thus helping counselors to acquire or improve reflective listening skills by emulating traditional psychotherapy training, but without the need of close human supervision.

We present a reflection generation system that leverages state-of-the-art language models, and further improve it with context augmentation techniques inspired by traditional counselor training. Specifically, we (1) identify previously used reflections from related sessions based on the current context, similar to how trainee counselors are exposed to several types of reflections on the same topic before they have to produce their own; and (2) we expand the content with synonyms for verbs and nouns, similar to how counselors are advised to use rephrasing strategies such as synonym rewording (Flasher and Fogle, 2012).

We perform a domain adaptation on an additional counseling corpus containing a variety of counseling styles, and fine-tune our system on a corpus of successful counseling interactions with labels available. Thus, it allows the system to benefit from successful counseling patterns derived from the cumulative experience of a large number of professionals. We conduct several comparative experiments, and perform evaluations using automatic metrics for language generation, including n-gram based, embedding-based and language diversity metrics. In addition, given the subjective nature of our task and the inability of automatic metrics to capture other relevant aspects of reflection generation, we conduct a human evaluation to assess the ability of our system to generate counseling reflections that are grammatically correct, fluent, and relevant to the conversation context.

2 Related Work

There have been significant efforts put in building automatic tools that provide support for mental and behavioral health. In particular, for dialogue-based counseling most of the existing work has focused on generating conversational agents that emulate

the counselor in chat-bot like settings. For instance, (Han et al., 2013) built a system that extracts 5w1h (who, what, when, where, why, and how) information and user emotions (happy, afraid, sad, and angry) to recognize what the user says, predict the conversation context and generate suitable responses based on utterance templates developed to encode three basic counseling techniques (paraphrasing, asking open questions, and reflecting feelings). A similar system is presented in (Han et al., 2015), where authors first detect the user emotion and intention (e.g., greeting, self-disclosure, informing, questioning) and then extract the entities present in the utterance as well as related information (from an external knowledge base) to generate an appropriate response using language templates.

While these studies have focused on the delivery of health interventions via conversational agents (i.e., virtual counselors), we seek to build an automatic dialogue generation system that can help training counselors to improve their everyday practice. This is in line with a recent study on the impact of technology in psychotherapy, which has identified the development of technologies for counselor’s training and feedback and technology-mediated treatment as important needs in this domain (Imel et al., 2017). Initial work in this direction is presented in (Tanana et al., 2019), where authors present a system that implements an artificial standardized client that interacts with the counselor and provides trainees with real-time feedback on their use of specific counseling skills by providing suggestions on the type of skills to use. Following the same line of work, our goal is to aid counselors while training specific skills, more specifically reflective listening skills. However, different from previous work, we focus on presenting the counselor with automatically generated samples for potential reflections that can be used immediately in the conversation.

Finally, potential applications of our proposed system include supporting counselor training in counseling platforms such as Talkspace², which currently has over a million users and five thousand therapists, and Crisis Text Line,³ with 20 thousand counselors, handling over three thousand conversations a day, allowing users to connect with licensed therapists and to seek help via text messaging. The ability to automatically generate reflections given

²<https://www.talkspace.com/>

³<https://www.crisistextline.org/>

a conversation context can assist these counselors in formulating what they are going to say, thus improving the efficiency and quality of their reflections, with the final goal of increasing the number of people they can help and the effectiveness of their interaction on patient outcomes.

3 Model Overview

To build an automatic reflection generation system, we rely on the Generative Pretrained Transformer 2 (GPT-2) architecture (Radford et al., 2019) as a base model. GPT-2 is a state of the art transformer-based general purpose language model that has been found useful for dialogue generation tasks (Zhang et al., 2019). Our choice is motivated by its ability to produce language that closely emulates text written by humans (Wolf et al., 2019b).

Our model learns how to generate a counselor reflection using a GPT-2 architecture by operating entirely in a sequence-to-sequence way. In order to condition the generation on the counseling dialogue context and to generate reflections that are stylistically correct, we fine-tune the model with conversations in the counseling domain.

Below, we describe important elements of the model architecture related to the reflection generation task.

Input representation. The input sequence for the model consists of a counselor’s utterance and a dialogue context including previous utterances from either the client or counselor. The window size of the dialogue context is set to five utterances, as a larger window size did not improve performance in preliminary experiments.

Embeddings. Besides learning word and positional embeddings, we also learn type embeddings to indicate whether the current token is part of the utterance from the client, counselor, or the reflection response. We use a trainable embedding matrix to map each location or type into a vector with the same size as the token embeddings. Separation tokens are also added to further delimit these elements in the dialogue.

Decoding details. The generator model consists of a transformer decoder with a similar structure to the decoder in (Vaswani et al., 2017) but only keeping the self-attention blocks. During the decoding stage, we assume we only have access to the augmented input and dialogue context and not the response. At each time-step, the model chooses

a token from the output distribution conditioned on the context and the previously decoded tokens. The chosen token will be added into the input in the next time-step. To generate more diverse and expressive reflections, we adopted the top-k random sampling method (Holtzman et al., 2019), where the model samples from the k options with the highest probabilities.

4 Counseling-style Reflection Generation

Our goal is not only to generate natural-looking text that is relevant to the prompt but also to resemble the language style that counselors use while generating reflections. Thus, we extend the base model to incorporate two strategies that are commonly used by counselors while generating reflective statements.

First, we consider a training scenario where trainees are first shown sample reflections made while discussing different behavioral change goals (e.g. smoking cessation or weight management). After they have been exposed to several types of reflections, trainees are usually asked to construct alternative reflections for a given scenario as a way to reinforce what they have learned. In this case, trainees might associate previous reflections with the same behavioral change target as potential examples to generate their own. We attempt to use the same strategy to improve our system’s responses. Thus, we devise a retrieval-based method to obtain a reflection to be used to expand the dialogue context.

Second, considering that counselors generate reflections using rephrasing strategies such as rewording with synonyms and verb tense changes, we design a content expansion method that augments the system input with verb and nouns synonyms. These methods are described in detail below.

4.1 Retrieval of the Most Similar Reflection

We seek to identify reflections that contain wording that could be useful for generating an appropriate reflection given the dialogue context. This is done in two main steps.

Selecting a relevant conversation. We start by identifying a set of relevant conversations i.e., conversations discussing the same behavior change. We then calculate the semantic similarity between the current dialogue context and this set of conversations. More specifically, we use TF-IDF (term frequency-inverse document frequency) encoding

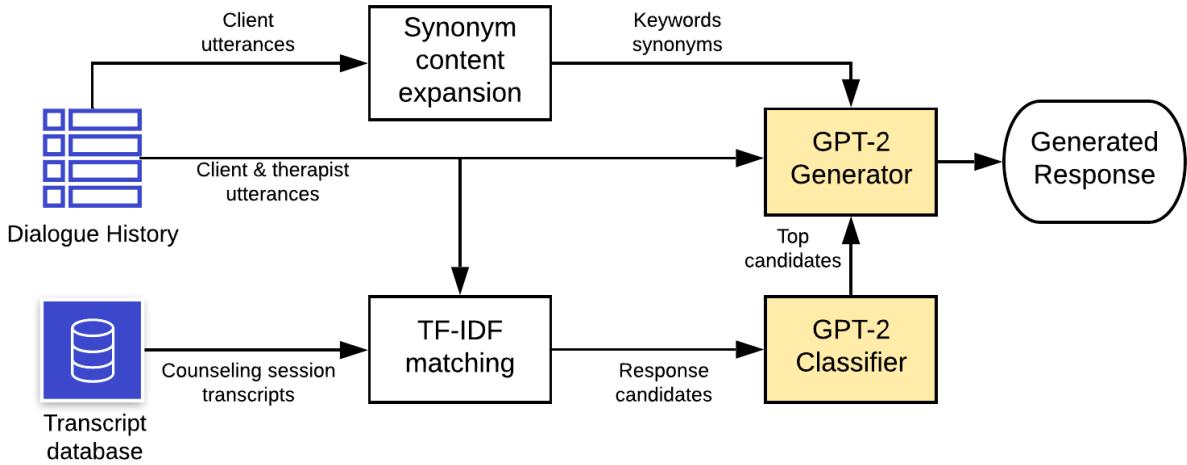


Figure 1: Model architecture. The fine-tuned model uses only client and therapist utterances, while the retrieval and content expansion models include additional input (TF-IDF matching and synonym content expansion) for the generation model.

Class	Precision	Recall	F1 score
In context	0.768	0.779	0.773
Not in context	0.765	0.754	0.759

Table 1: Performance metrics for the reflection-in-context classifier

for the dialogue context and candidate conversations and calculate their cosine similarity. We then select the conversation with the highest similarity as the most relevant conversation given the context. This stage may be further improved with methods such as BM25 or neural-based matching in future work.

Selecting a candidate reflection. Our next step focuses on identifying, among the reflections made in the most similar conversation, which of them is more likely to be a good match to the current context. The selected reflection is then added to the input of the generation system as a way to provide wording alternatives. For this task, we first build a set of candidate pairs by concatenating the current dialogue context and each of the reflections made in the most similar conversation. Then, we feed them to a binary classifier that aims to classify whether a sequence contains a valid reflection according to the given context. We score each sequence using the probabilities provided by the classifier and choose the one with the highest score as the best example reflection to be added to our current dialogue context.

To build the reflection-in-context classifier, we use a GPT-2 model and modify it by adding a clas-

sification layer to the output layer. The classifier is trained on a balanced set, with positive samples consisting of reflections from our main dataset, along with five previous utterances in the actual conversation, and negative samples consisting of reflections paired with random context windows taken from different conversations. We train the classifier using an 80%-20% split for training and testing sets respectively. The classifier achieves an accuracy of 76%, with detailed metrics per class shown in Table 1, thus showing reasonable performance on determining whether a reflection matches the current context.

4.2 Content Expansion

We augment the context content by applying synset expansion to synonyms and verbs. We first apply part-of-speech (POS) tagging on the context utterances using Stanford CoreNLP (Manning et al., 2014) to identify nouns and verbs and then obtain their corresponding synonyms for all their meanings using the English WordNet (Miller, 1998).

We then produce one rephrase for each utterance in the context by replacing the original nouns and/or verbs with a randomly selected synonym with the same POS tag. Our system uses the resulting utterances to augment the current context.

5 Experimental Setup

5.1 Counseling Datasets

We use the Motivational Interviewing (MI) counseling dataset from Pérez-Rosas et al. (2016) as the main corpus for training our retrieval and genera-

Total sessions	254
Vocabulary size	8,259
Total reflections	3,939
Average turns / session	97.2
Average tokens / reflection	20.9

Table 2: Statistics of the MI dataset

tion models, and perform language model domain adaptation using the Alexander Street dataset consisting of a variety of psychotherapy styles (e.g., cognitive behavioral, existential, solution focused). The datasets are described below.

MI Counseling Dataset: This dataset consists of 276 MI conversations annotated at utterance level with counselor verbal behaviors using the Motivational Interviewing Treatment Integrity 4.0 (MITI). In addition, the dataset also contains labels at the session-level, which evaluate the quality of the counseling interaction. The conversations portray MI encounters for three main behavior change goals: smoking cessation, medication adherence, and weight management. Among the different annotations available in the dataset, we focus on the annotations of counselor reflections, including simple reflections and complex reflections. Before we use the MI dataset, we remove transcripts corresponding to encounters that were deemed as low-quality counseling based on the global evaluation of the counseling interactions, i.e., sessions having low empathy scores or a low ratio of questions to reflections. We are thus left with a set of 254 counseling conversations. Dataset statistics are provided in Table 2. During our experiments using this dataset, we use 10% of the data as the test set and 5% as the validation set.

Alexander Street Dataset: This is a collection of psychotherapy videos that are published by Alexander Street Press.⁴ The videos and its corresponding transcripts, containing psychotherapy conversations between clients and therapists on several behavioral and mental issues, are available through a library subscription. From this library, we downloaded the transcripts available under the Counseling & Therapy in Video: Volume IV, which contains around 400 real therapy sessions. However, due to the format inconsistencies, we were able to collect only 312 transcripts.

⁴<http://alexanderstreet.com/>

5.2 Reflection Generation Neural Architecture

During our experiments, we use a medium-size pre-trained GPT-2 (Radford et al., 2019) model as the backbone network for the language generation models. Our models are implemented using the Transformers library (Wolf et al., 2019a). The base model uses a byte-pair encoding (BPE) (Gage, 1994) and has a vocabulary size of 50,257. We use dropout with probability 0.1 for the embedding and attention layers and also for the residual connection in the blocks.

In addition, we use a warmup scheme for the learning rate using 5% of the total steps as warmup steps (Popel and Bojar, 2018). We use the Adam optimizer with weight decay (Kingma and Ba, 2015) to optimize the network at a learning rate of 6e-5. All models are trained for 10 epochs with early stopping.

5.3 Reflection Generation Experiments

We conduct two main sets of experiments on automatic reflection generation as described below. During our experiments we use the datasets described in section 5.1.

Reflection generation using a fine-tuned GPT-2 model. In this experiment we use the base model described in section 5.2 to generate counselor reflections. We first perform domain adaption of the language model using the Alexander Street dataset. We then fine-tune the generator using the MI dataset.

Reflection generation with retrieval and content expansion strategies. We extend the fine-tuned model to include the retrieval of the most similar reflection and content expansion strategies described in section 4.1 and 4.2. We experiment with incremental models that incorporate one strategy at the time.

Finally, we compare our models with a seq2seq model, which is frequently used as a baseline for conditional text generation problems (Vinyals and Le, 2015). We use the seq2seq implementation available in OpenNMT (Klein et al., 2017). The encoder and decoder are 2-layers GRU (Gated Recurrent Units) (Cho et al., 2014) with 512 hidden units. We train the model for 10 epochs with an Adam optimizer at a learning rate of 0.001.

Models	ROUGE			Embedding			Diversity		Avg Len
	RG-1	RG-2	RG-L	Greedy	Average	Extrema	Div-1	Div-2	
Seq2Seq	0.078	0.004	0.060	0.363	0.613	0.309	0.156	0.447	11.189
Fine-tuned GPT-2	0.152	0.020	0.117	0.446	0.726	0.382	0.134	0.496	18.522
+ retrieval	0.156	0.025	0.117	0.456	0.735	0.390	0.127	0.486	18.677
+ content expansion	0.162	0.031	0.126	0.453	0.731	0.386	0.128	0.498	18.412

Table 3: Performance of our models and the seq2seq baseline on the automatic generation of counselor reflections using ROUGE and embedding based metrics and n-gram diversity. We also show the average length of generated utterances for each model.

5.3.1 Automatic Evaluation Metrics

For the quantitative analysis of our reflection generation model, we use well-known automatic metrics for language generation, including:

ROUGE metrics: We use the ROUGE metric, a word overlap metric frequently used in the evaluation of neural language generation systems (Lin, 2004), including ROUGE-N, and ROUGE-L.

We decided to use ROUGE over other n-gram-based metrics, such as BLEU, because our task of generating reflective responses shares some similarity with the task of text summarization, where ROUGE is the metric of choice. Additionally, evaluations that we ran with other n-gram-based metrics had results consistent with those obtained with ROUGE.

Embedding-based metrics: We also use three embedding-based metrics, namely greedy matching, embedding average, and vector extrema (Liu et al., 2016). The first matches each token in one sentence to its nearest neighbor in the reference sentence, this metric favours generated reflections containing keywords that are semantically similar to the ground truth reflection. The other two calculate similarity for a pair of sentences based on their vector representations instead of matching each word. The sentence vector representations are constructed by averaging the word embeddings or taking the number with the highest absolute value for each dimension.

Diversity: We also evaluate diversity by measuring the ratio of distinct n-grams in the generated reflection with respect to the reference reflection.

5.3.2 Human Evaluation for Reflection Generation

To assess our automatic reflection generation systems’ ability to produce relevant and coherent reflections, we also conducted a human evaluation

study. We recruited two annotators familiar with counseling reflections, and asked them to evaluate the generated outputs and the ground truth responses for 50 samples randomly chosen from our test set. Given the conversation context of the latest five utterances, the annotators are asked to evaluate three main properties of several response candidates: relevance, reflection-likeness, and quality. The candidates are composed of the ground truth response and generated responses from four systems, i.e. seq2seq, GPT fine-tuned, and two improved versions using retrieval and content expansion. The annotators evaluate one candidate at a time, without knowledge of its origin.

Quality is evaluated using a 5-point Likert scale (i.e., 5: very good, 4: good, 3: acceptable, 2: poor and 1: very poor). We chose a 3-point Likert scale (i.e., 1: not at all, 2: somewhat, 3: very much) to evaluate relevance and reflection-likeness, since a finer scale may exceed the annotators’ discriminating power (Jacoby and Matell, 1971). More specifically, we use the following prompts:

Relevance: Does the response seem appropriate to the conversation? Is the response on-topic?

Reflection-likeness: Does the response show understanding of the feelings of the client? Does the response paraphrase or summarize what the client has said?

Quality: How do you judge the overall quality of the utterance in terms of its grammatical correctness and fluency?

We measured inter-rater agreement using Krippendorff’s α (Krippendorff, 2018) and obtain agreement values of 0.18, 0.23, and 0.12 for relevance, reflection-likeness, and quality, respectively. The subjective nature of the question prompts may be the main reason for the low to fair levels of agreement on the different categories. The difference in

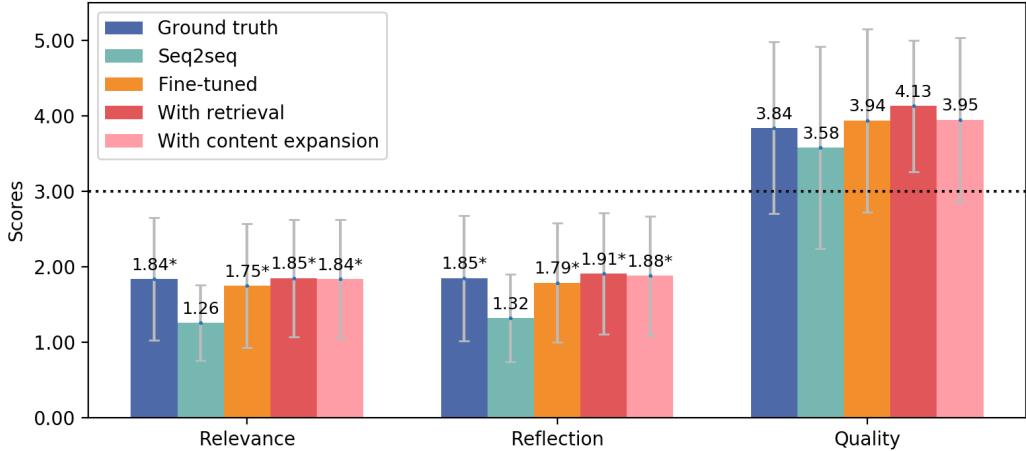


Figure 2: Human evaluation mean scores and standard deviations on the three criteria: relevance, reflection-likeness, and quality. (The former two criteria are in 3-point Likert scales. Quality uses a 5-point Likert scale; “*” indicate statistically significant improvement ($p < 0.01$) over the seq2seq baseline)

personal preference and the level of background knowledge can both be sources of disagreement (Amidei et al., 2018). We plan to use more sophisticated evaluation schemes in future work, such as magnitude estimation or RankME (Novikova et al., 2018), instead of a plain Likert scale.

6 Results

6.1 Automatic Metrics

Table 3 reports scores for our models and the seq2seq baseline. From this table, we observe that all our proposed models outperform the seq2seq baseline as measured by the different metrics. In addition, our models with context augmentation (i.e., including retrieval of the most similar reflection and content expansion) outperform the fine-tuned model, thus suggesting that the proposed retrieval and expansion strategies are useful to improve the generation of reflections. Interestingly, the generation model augmented with the most similar reflection scores higher when using the embedding metrics, thus indicating that the model benefits from augmenting the context with words that are semantically close to it. Similarly, when using context expansion, we observe improved scores for the ROUGE-based metrics as the model takes advantage of the additional wording alternatives.

6.2 Human Evaluations

The average scores for each system response on *relevance*, *reflection-likeness* and *quality* are shown

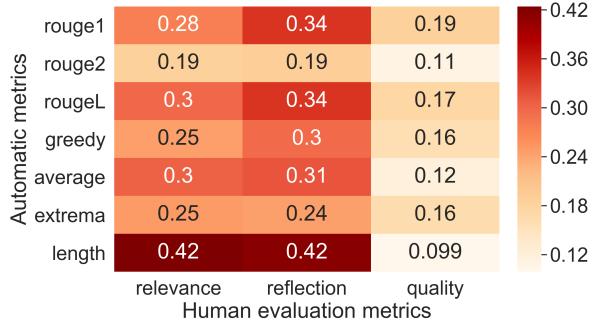


Figure 3: Spearman’s correlation between human evaluation metrics and automatic metrics

in Figure 2. From this figure, the general trend indicates that our systems perform on-par or above the reference reflections (ground truth), and outperform the baseline with statistical significance for both, relevance and reflection-likeness.

More specifically, in terms of *relevance*, we observe that our context-augmented models score the best by providing reflections that are more in line with the conversation context. Furthermore, they score very close to the ground truth reflection. In contrast, the seq2seq output obtains the lowest scores, thus suggesting that the generated reflections are more likely to be off-topic. The *reflection-likeness* follows a similar trend, confirming that the retrieval and expansion strategies do improve the model context.

For *quality* assessments, all systems (with the exception of the seq2seq model) produce responses considered to be of good quality according to the

Counseling Dialogue Context	Model	Generated Reflection
<p>C: Um, yes, um I think, um, more so in terms of my family history, um and then suddenly last week I got the diagnosis of pre - diabetes. Um, and um, that's kinda thrown a wrench into my otherwise healthy normal you know life that I think I lead.</p> <p>T: So you are worried that this diagnosis is going to impact your day to day activities?</p> <p>C: Um, I'm worried a whole lot because my dad has, um diabetes and he uh, is right now going under dialysis.... So, the idea that I might be facing some of these issues in my future is horrendous for me.</p>	Ground truth	Sounds like you really want to make some changes just to, as a prevention way to make sure you don't have to go through some of the things your farther went through.
	Seq2seq	Okay, so no missed meds.
	Fine-tuned	So you are really concerned about your weight.
	+ Content expansion	So you are really concerned about your health and where it might go in the future .
<p>C: The joy of smoking!</p> <p>T: It's just, there's something that is enjoyable</p> <p>C: Yes! It's habit! I mean it is a habit. And there are certain things that make me smoke, you know, reasons I smoke every day</p> <p>T: Uh huh,</p> <p>C: It's the nicotine, is a piece of it. But it's not the only piece.</p> <p>Retrieved: So it's really becoming realistic in terms of how this habit that you've gotten used to can play a part in your other behaviors and your activities that you enjoy too.</p>	Ground truth	Uh huh, ok, so you enjoy it.
	Seq2seq	In the trash can.
	Fine-tuned	There are other things that make you smoke, and that you like to have, so there's that, there's some withdrawal or other symptoms that make you smoke.
	+ Retrieval	There are other things that you enjoy doing, okay!

Table 4: **Top:** A case where content expansion helps. **Bottom:** A case where retrieval leads to an off-topic response. Examples of the generated reflections using the seq2seq baseline and our proposed systems. We show the dialogue context along with the ground truth reflection, where C stands for client, T stands for therapist. If the generated reflection consists of multiple utterances, we only show the first utterance.

Likert scale. Particularly, the system with retrieval scores 4.8% higher than the system with only fine-tuning. This can be attributed to the model conditioning the generation on sentences that have similar content but somewhat varying structure, thus allowing the model to take advantage of this information. The content expansion model performs similarly to the model with just fine-tuning. A potential explanation for this is that the content expansion sometimes produces subject-verb disagreement thus introducing noise during the generation process.

Finally, the *reflection-likeness* aspect of our evaluation obtains the highest scores from our retrieval model, followed by the content expansion model.

Human correlation analysis. To further validate our models, we conduct a correlation analysis between automatic metrics and human assessments as shown in Figure 3. In this analysis, we use Spearman’s correlation because we care more about the monotonic relationship of the metrics instead of a linear relationship. From the results, we observe that the automatic metrics show weak positive correlations with human evaluations of *relevance* and *reflection-likeness*. Moreover, the *quality* evaluation shows a weak correlation with automatic met-

rics, which is somehow expected as n-gram-based metrics and embedding-based metrics do not take grammar into consideration. Similarly, the average length of generated reflections has almost no impact on whether the response is fluent or contains grammatical errors. On the other hand, average length obtains the highest correlations with reflection-likeness and relevance, suggesting that a longer reflection is more likely to contain information the client has previously mentioned.

6.3 Qualitative Analysis

To gain further insights into how the augmented input helps with generation, we analyze a sample output for our different systems as shown in Table 4. From this table, we observe that all models based on the pre-trained GPT-2 are able to generate reflections that agree, to some extent, with the dialogue context.

For the counseling conversation shown in the upper side of the table, we observe that the seq2seq model generates an off-topic reflection while the reflections generated by the other systems seem to be more relevant to the context. Therefore, showing the effectiveness of transfer learning for counseling-style reflection generation. More interestingly, when using content expansion the sys-

tem is able to generate a reflection with the phrase “in the future” as a more specific response, which further confirms that our expansion strategy does strengthen the signal of important information that we want the model to capture.

We also observe cases where our methods introduce noise in the reflection generation system. For example, in the counseling conversation shown in the bottom section of Table 4, the model trained without augmented context produces the most appropriate response. The retrieved sentence successfully captures the idea of “habits,” while the conversation is about reasons other than habits that make the client to enjoy smoking, thus leading to the generation of a less relevant reflection.

7 Conclusion

We presented a system based on a state of the art language model that generates counseling reflections based on the counselor-client dialogue context. We first conducted domain adaptation and subsequently fine-tuned the system with motivational interviewing conversations. We then improved the system by augmenting the dialogue context using retrieval and content expansion methods that implement actual strategies used by counselors while generating reflections.

We conducted comparative experiments between systems implementing these strategies and demonstrated their effectiveness in generating improved reflections as measured by standard language generation metrics such as ROUGE as well as embedding-based and diversity metrics. To further validate our models, we conducted a human evaluation study on the generated responses. The evaluation showed that humans scored our proposed systems higher than the baseline model on quality, relevance, and reflection-likeness.

We believe that counselors could benefit from the proposed system by using the automatically generated reflections as reference while learning to formulate reflective statements.

Acknowledgements

We are grateful to Christy Li, Yinwei Dai, Jiajun Bao, and Allison Lahnlala for assisting us with the human evaluations. This material is based in part upon work supported by the Precision Health initiative at the University of Michigan, by the National Science Foundation (grant #1815291), and by the John Templeton Foundation (grant #61156). Any

opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Precision Health initiative, the National Science Foundation, or John Templeton Foundation.

References

- Jacopo Amidei, Paul Piwek, and Alistair Willis. 2018. [Rethinking the agreement in human evaluation tasks](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3318–3329, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- David C Atkins, Mark Steyvers, Zac E Imel, and Padhraic Smyth. 2014. Scaling up the evaluation of psychotherapy: evaluating motivational interviewing fidelity via statistical text classification. *Implementation Science*, 9(1):49.
- Jie Cao, Michael Tanana, Zac Imel, Eric Poitras, David Atkins, and Vivek Srikumar. 2019. [Observing dialogue in therapy: Categorizing and forecasting behavioral codes](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5599–5611, Florence, Italy. Association for Computational Linguistics.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Lydia V Flasher and Paul T Fogle. 2012. *Counseling skills for speech-language pathologists and audiologists*. Cengage Learning.
- Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.
- Sangdo Han, Jeesoo Bang, Seonghan Ryu, and Gary Geunbae Lee. 2015. [Exploiting knowledge base to generate responses for natural language dialog listening agents](#). In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 129–133, Prague, Czech Republic. Association for Computational Linguistics.
- Sangdo Han, Kyusong Lee, Donghyeon Lee, and Gary Geunbae Lee. 2013. [Counseling dialog system with 5W1H extraction](#). In *Proceedings of the SIGDIAL 2013 Conference*, pages 349–353, Metz, France. Association for Computational Linguistics.
- Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

- Zac E Imel, Derek D Caperton, Michael Tanana, and David C Atkins. 2017. Technology-enhanced human interaction in psychotherapy. *Journal of counseling psychology*, 64(4):385.
- Jacob Jacoby and Michael S Matell. 1971. Three-point likert scales are good enough.
- Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ICLR*.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.
- Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.
- William R Miller and Stephen Rollnick. 2013. *Motivational interviewing: Helping people change, Third edition*. The Guilford Press.
- William R Miller, Carolina E Yahne, Theresa B Moyers, James Martinez, and Matthew Pirritano. 2004. A randomized trial of methods to help clinicians learn motivational interviewing. *Journal of consulting and Clinical Psychology*, 72(6):1050.
- Theresa B Moyers, Tim Martin, Jon M Houck, Paulette J Christopher, and J Scott Tonigan. 2009. From in-session behaviors to drinking outcomes: a causal chain for motivational interviewing. *Journal of consulting and clinical psychology*, 77(6):1113.
- Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2018. RankME: Reliable human ratings for natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 72–78, New Orleans, Louisiana. Association for Computational Linguistics.
- Sungjoon Park, Donghyun Kim, and Alice Oh. 2019. Conversation model fine-tuning for classifying client utterances in counseling dialogues. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1448–1459, Minneapolis, Minnesota. Association for Computational Linguistics.
- Verónica Pérez-Rosas, Rada Mihalcea, Kenneth Resnick, Satinder Singh, and Lawrence An. 2016. Building a motivational interviewing dataset. In *Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology*, pages 42–51, San Diego, CA, USA. Association for Computational Linguistics.
- Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).
- Michael Tanana, Kevin A Hallgren, Zac E Imel, David C Atkins, and Vivek Srikumar. 2016. A comparison of natural language processing methods for automated coding of motivational interviewing. *Journal of substance abuse treatment*, 65:43–50.
- Michael J Tanana, Christina S Soma, Vivek Srikumar, David C Atkins, and Zac E Imel. 2019. Development and evaluation of clientbot: Patient-like conversational agent to train basic counseling skills. *J Med Internet Res*, 21(7):e12529.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.
- Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrette Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019a. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.
- Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019b. Transfertransfo: A transfer learning approach for neural network based conversational agents. *CoRR*, abs/1901.08149.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen,
Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing
Liu, and Bill Dolan. 2019. Dialogpt: Large-scale
generative pre-training for conversational response
generation.

Learning from Mistakes: Combining Ontologies via Self-Training for Dialogue Generation

Lena Reed¹, Vrindavan Harrison¹, Shereen Oraby^{2*},
Dilek Hakkani-Tür^{2*}, and Marilyn Walker¹

¹Natural Language and Dialogue Systems Lab, University of California, Santa Cruz

²Amazon Alexa AI

{lireed, vharriso, mawalker}@ucsc.edu
{orabys, hakkaniit}@amazon.com

Abstract

Natural language generators (NLGs) for task-oriented dialogue typically take a meaning representation (MR) as input, and are trained end-to-end with a corpus of MR/utterance pairs, where the MRs cover a specific set of dialogue acts and domain attributes. Creation of such datasets is labor intensive and time consuming. Therefore, dialogue systems for new domain ontologies would benefit from using data for pre-existing ontologies. Here we explore, for the first time, whether it is possible to train an NLG for a new **larger** ontology using existing training sets for the restaurant domain, where each set is based on a **different** ontology. We create a new, larger **combined** ontology, and then train an NLG to produce utterances covering it. For example, if one dataset has attributes for *family friendly* and *rating* information, and the other has attributes for *decor* and *service*, our aim is an NLG for the combined ontology that can produce utterances that realize values for *family friendly*, *rating*, *decor* and *service*. Initial experiments with a baseline neural sequence-to-sequence model show that this task is surprisingly challenging. We then develop a novel **self-training** method that identifies (errorful) model outputs, automatically constructs a corrected MR input to form a new (MR, utterance) training pair, and then repeatedly adds these new instances back into the training data. We then test the resulting model on a new test set. The result is a self-trained model whose performance is an absolute 75.4% improvement over the baseline model. We also report a human qualitative evaluation of the final model showing that it achieves high naturalness, semantic coherence and grammaticality.

1 Introduction

Natural language generators (NLGs) for task-oriented dialogue take meaning representations

*Work done prior to joining Amazon.

(MRs) as inputs, i.e. a set of dialogue acts with attributes and their values, and output natural language utterances realizing the MR. Current NLGs are trained end-to-end with a corpus of MR/utterance pairs where the MRs cover a specific set of dialogue acts and domain attributes. Creation of such datasets is labor intensive and time consuming. However, when building an NLG for a new domain ontology, it should be possible to re-use data built on existing domain ontologies. If this were possible, it would speed up development of new dialogue systems significantly.

Here we experiment with one version of this task by building a new domain ontology based on **combining** two existing ontologies, and utilizing their training data. Each dataset is based on a different domain ontology in the restaurant domain, with novel attributes and dialogue acts not seen in the other dataset, e.g. only one has attributes representing *family friendly* and *rating* information, and only one has attributes for *decor* and *service*. Our aim is an NLG engine that can realize utterances for the extended **combined** ontology not seen in the training data, e.g. for MRs that specify values for *family friendly*, *rating*, *decor* and *service*. Figure 1 illustrates this task. Example E1 is from a training set referred to as NYC, from previous work on controllable sentence planning in NLG (Reed et al., 2018), while E2 is from the E2E NLG shared task (Novikova et al., 2017a). As we describe in detail in Section 2, E1 and E2 are based on two distinct ontologies. Example E3 illustrates the task addressed in this paper: we create a test set of novel MRs for the combined ontology, and train a model to generate high quality outputs where individual sentences realize attributes from both ontologies.

To our knowledge, this is a completely novel task. While it is common practice in NLG to construct test sets of MRs that realize attribute combinations not seen in training, initial experiments

ID	Ontology	MEANING REPRESENTATION	EXAMPLE
E1	NYC (TRAIN- ING)	RECOMMEND[YES], INFORM(NAME[RESTAURANT], SERVICE[EXCELLENT], DÉCOR[EXCELLENT], PRICE[EXPENSIVE])	I suggest you go to [RESTAURANT]. The <u>food</u> , <u>service</u> and <u>atmosphere</u> <u>are all excellent</u> , even if it is <u>expensive</u> . Its in [AREA].
E2	E2E (TRAIN- ING)	INFORM(NAME[RESTAURANT], EATTYPE[RESTAURANT-TYPE], CUSTOMER-RATING[HIGH], AREA[AREA], NEAR[POINT-OF-INTEREST])	[RESTAURANT] is a [RESTAURANT-TYPE] in [AREA] <u>near [POINT-OF-INTEREST]</u> . It has a <u>high customer rating</u> .
E3	COMBINED (TEST)	RECOMMEND = YES, INFORM(NAME[RESTAURANT], EATTYPE[RESTAURANT-TYPE], FOOD = EXCELLENT, LOCATION[AREA], NEAR[POINT-OF-INTEREST], CUSTOMER-RATING[HIGH], DÉCOR = EXCELLENT, SERVICE=EXCELLENT, PRICE=EXPENSIVE)	[RESTAURANT] is the best because it has excellent service and atmosphere. It is a [RESTAURANT-TYPE] offering excellent food in [AREA] <u>near [POINT-OF-INTEREST]</u> with a <u>high customer rating</u> , but it is <u>expensive</u> .

Figure 1: E1 and E2 illustrate training instances from the two source datasets E2E and NYC. E2E attributes are represented in blue and NYC is in red. Some attributes are shared between both sources: here the unique dialogue acts and attributes for each source are underlined in E1 and E2. E3 illustrates an MR from the target test set that we dub COM. All the MRs in COM combine dialogue acts and attributes from E2E and NYC. There is no training data corresponding to E3. The MRs illustrate how some attribute values, e.g. RESTAURANT NAME, POINT-OF-INTEREST, are delexicalized to improve generalization.

showed that this task is surprisingly adversarial. However, methods for supporting this type of generalization and extension to new cases would be of great benefit to task-oriented dialogue systems, where it is common to start with a restricted set of attributes and then enlarge the domain ontology over time. New attributes are constantly being added to databases of restaurants, hotels and other entities to support better recommendations and better search. Our experiments test whether existing data that only covers a subset of attributes can be used to produce an NLG for the enlarged ontology.

We describe below how we create a test set — that we call COM — of combined MRs to test different methods for creating such an NLG. A baseline sequence-to-sequence NLG model has a slot error rate (SER) of .45 and only produces semantically perfect outputs 3.5% of the time. To improve performance, we experiment with three different ways of conditioning the model by incorporating *side constraints* that encode the source of the attributes in the MR (Sennrich et al., 2016; Harrison et al., 2019). However, this only increases the proportion of semantically perfect model outputs from 3.5% to 5.5% (Section 4.1).

We then propose and motivate a novel self-training method that greatly improves performance by learning from the model mistakes. An error analysis shows that the models **do** produce many **combined** outputs, but with errorful semantics. We develop a rule-based text-to-meaning semantic extractor that automatically creates novel correct MR/text

training instances from errorful model outputs, and use these in self-training experiments, thus learning from our mistakes (Section 4.2). We validate the text-to-meaning extractor with a human evaluation. We find that a model trained with this process produces SERs of only .03, and semantically perfect outputs 81% of the time (a 75.4 percent improvement). A human evaluation shows that these outputs are also natural, coherent and grammatical. Our contributions are:

- Definition of a novel generalization task for neural NLG engines, that of generating from unseen MRs that combine attributes from two datasets with different ontologies;
- Systematic experiments on methods for conditioning NLG models, with results showing the effects on model performance for both semantic errors and combining attributes;
- A novel self-training method that learns from the model’s mistakes to produce semantically correct outputs 81% of the time, an absolute 75.4% improvement.

We start in Section 2 by defining the task in more detail, describe our models and metrics in Section 3, and results in Section 4. We discuss related work throughout the paper where it is most relevant and in the conclusion in Section 5.

2 Ontology Merging and Data Curation

We start with two existing datasets, NYC and E2E, representing different ontologies for the restaurant

domain. The NYC dataset consists of 38K utterances (Reed et al., 2018; Oraby et al., 2018), based on a restaurant ontology used by Zagat (Stent et al., 2002, 2004).¹ The E2E dataset consists of 47K utterances distributed for the E2E Generation Challenge (Novikova et al., 2017a).² Each dataset consists of pairs of reference utterances and meaning representations (MRs). Figure 1 shows sample MRs for each source and corresponding training instances as E1 and E2.

Ontology Merging. We first make a new combined ontology ONTO-COM by merging NYC and E2E. Attributes, dialogue acts, and sample values for E2E and NYC are illustrated on the left-hand side of Figure 2, and the result of merging them to create the new ontology is on the right-hand side of Figure 2. Since there are only 8 attributes in each source dataset, we developed a script by hand that maps the MRs from each source into the ONTO-COM ontology.

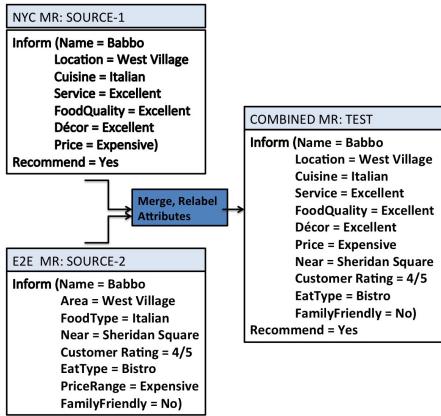


Figure 2: An example illustrating how dialogue acts and attributes for both source databases are merged and relabelled to make a new combined ontology used in train and test.

As Figure 2 shows, both datasets have the **INFORM** dialogue act, and include the attributes *name*, *cuisine*, *location*, and *price* after mapping. The unique attributes for the NYC ontology are scalar ratings for *service*, *food quality* and *decor*. The NYC dataset also has the **RECOMMEND** dialogue act, seen in E1 in Figure 1. The unique attributes of the E2E ontology are *customer rating*, *eat type* (“coffee shop”), *near* and *family friendly*.

Training Data. Given the combined ontology ONTO-COM, we then map the training data for both E2E and NYC into ONTO-COM by relabelling the

MRs to have consistent names for shared attributes as illustrated in Figure 2. We create a balanced training set of $\sim 77K$ from the two original datasets by combining all NYC references with a random same-size sample of E2E references.

Test Set. We then manually create a test set, COM, consisting of 3040 MRs based on the new combined ontology ONTO-COM. Each test MR must have at least one attribute from E2E and one attribute from NYC so that it combines attributes from both sources: these MRs provide combinations never seen in training.³ Example E3 in Figure 1 provides an example test MR. The procedure for creating the test set ensures that the length and complexity of the test set are systematically varied, with lengths normally distributed and ranging from 3 to 10 attributes. Recommendations only occur in the NYC training data, and they increase both **semantic** and **syntactic** complexity, with longer utterances that use the discourse relation of **JUSTIFICATION** (Stent et al., 2002), e.g. *Babbo is the best because it has excellent food*. We hypothesize that recommendations may be more challenging to combine across domains, so we vary MR complexity by including the **RECOMMEND** dialogue act in half the test references. We show in Section 4 that the length and complexity of the MRs is an important factor in the performance of the trained models.

3 Experimental Overview and Methods

Given the training and test sets for the combined ontology in Section 2, we test 4 different neural model architectures and present results in Section 4.1. We then propose a novel self-training method, and present results in Section 4.2. These experiments rely on the model architectures presented here in Section 3.1, and the Text-to-Meaning semantic extractor and performance metrics in Section 3.2.

3.1 Model Architectures

In the recent E2E NLG Challenge shared task, models were tasked with generating surface forms from structured meaning representations (MRs) (Dušek et al., 2020). The top performing models were all RNN encoder-decoder systems. Here we also use a standard RNN Encoder–Decoder model (Sutskever et al., 2014) that maps a source sequence (the input MR) to a target sequence (the utterance text). We

¹<http://nlds.soe.ucsc.edu/sentence-planning-NLG>

²<http://www.macs.hw.ac.uk/InteractionLab/E2E/>

³The train and test data are available at <http://nlds.soe.ucsc.edu/source-blending-NLG>

first implement a baseline model and then add three variations of model supervision that aim to improve semantic accuracy. All of the models are built with OpenNMT-py, a sequence-to-sequence modeling framework (Klein et al., 2017).

Encoder. The MR is represented as a sequence of (attribute, value) pairs with separate vocabularies for attributes and values. Each attribute and each value are represented using 1-hot vectors. An (attribute, value) pair is represented by concatenating the two 1-hot vectors.

The input sequence is processed using two single layer bidirectional-LSTM (Hochreiter and Schmidhuber, 1997) encoders. The first encoder operates at the pair level, producing a hidden state for each attribute-value pair of the input sequence. The second LSTM encoder is intended to produce utterance level context information in the form of a full MR encoding produced by taking the final hidden state after processing the full input sequence. The outputs of both encoders are combined via concatenation. That is, the final state of the second encoder is concatenated onto each hidden state output by the first encoder. The size of the pair level encoder is 46 units and the size of the MR encoder is 20 units. Model parameters are initialized using Glorot initialization (Glorot and Bengio, 2010) and optimized using Stochastic Gradient Descent with mini-batches of size 128.

Decoder. The decoder is a uni-directional LSTM that uses global attention with input-feeding. Attention weights are calculated via the *general* scoring method (Luong et al., 2015). The decoder takes two inputs at each time step: the word embedding of the previous time step, and the attention weighted average of the encoder hidden states. The ground-truth previous word is used when training, and the predicted previous word when evaluating. Beam search with five beams is used during inference.

Supervision. Figure 3 shows the baseline system architecture as well as three types of supervision, based on conditioning on source (E2E, NYC) information. The additional supervision is intended to help the model attend to the source domain information. We call the three types of supervision GUIDE, ATTR and BOOL, and the baseline architecture NOSUP, representing that it has no additional supervision.

The supervision methods are shown in Figure 4. The source feature has a vocabulary of three items: *nyc*, *e2e* and *both*. Since *both* is never seen

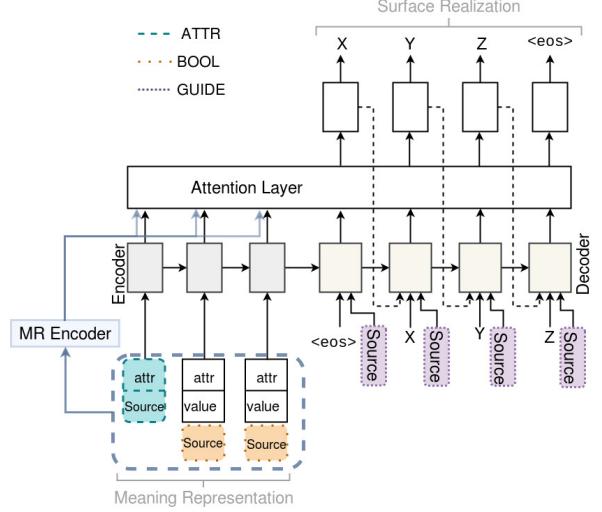


Figure 3: Attentional Encoder-Decoder architecture with each supervision method shown.

ATTR	Name Restaurant	Near Point-of-interest	Service Good
	nyc=true e2e=true	nyc=true e2e=false	nyc=false e2e=true
BOOL	Name Restaurant	Near Point-of-interest	Service Good
			Source nyc & e2e

Figure 4: An illustration of ATTR and BOOL supervision methods, with the source supervision (NYC or E2E) shown in red.

in train, the source information is represented using two booleans: *True||False* denotes a reference from E2E while *False||True* denotes a reference from NYC. This encoding is intended to encourage generalization at inference time. During inference, blending of information from both sources is specified by using *True||True*. The ATTR supervision method represents the source information by concatenating the boolean source token onto each attribute as seen in Figure 4. This redundantly represents the source information locally to each attribute, which has been effective for tasks such as question generation and stylistic control (Harrison and Walker, 2018; Harrison et al., 2019). The BOOL supervision method adds the boolean source token to the end of the sequence of attribute-value pairs as its own attribute, as in work on machine translation and controllable stylistic generation (Sennrich et al., 2016; Yamagishi et al., 2016; Ficler and Goldberg, 2017). The GUIDE model inputs the source information directly to the decoder LSTM. In previous work, putting information into the decoder in this way has yielded improvements in paraphrase

generation and controllable generation (Iyyer et al., 2018; Harrison et al., 2019)

3.2 Text-to-Meaning Semantic Extractor

Much previous work in NLG relies on a test set that provides gold reference outputs, and then applies automatic metrics such as BLEU that compare the gold reference to the model output (Papineni et al., 2002; Dušek et al., 2020), even though the limitations of BLEU for NLG are widely acknowledged (Belz and Reiter, 2006; Stent et al., 2005; Novikova et al., 2017b; Liu et al., 2016). To address these limitations, recent work has started to develop “referenceless” NLG evaluation metrics (Dusek et al., 2017; Kann et al., 2018; Tian et al., 2018; Mehri and Eskenazi, 2020).

Since there are no reference outputs for the COM test set, we need a referenceless evaluation metric. We develop a rule-based text-to-MR semantic extractor (TTM) that allows us to compare the input MR to an MR automatically constructed from an NLG model textual output by the TTM, in order to calculate **SER**, the slot error rate. The TTM system is based on information extraction methods. We conduct a human evaluation of its accuracy below. A similar approach is used to calculate semantic accuracy in other work in NLG, including comparative system evaluation in the E2E Generation Challenge (Juraska et al., 2018; Dušek et al., 2020; Wiseman et al., 2017; Shen et al., 2019).

The TTM relies on a rule-based automatic aligner that tags each output utterance with the attributes and values that it realizes. The aligner takes advantage of the fact that the RECOMMEND dialogue act, and the attributes and their values are typically realized from a domain-specific finite vocabulary. The output of the aligner is then used by the TTM extractor to construct an MR that matches the (potentially errorful) utterance that was generated by the NLG. We refer to this MR as the “retrofit MR”. The retrofit MR is then compared to the input MR in order to automatically calculate the slot error rate **SER**:

$$SER = \frac{D + R + S + H}{N}$$

where D is the number of deletions, R is the number of repetitions, S is the number of substitutions, H is the number of hallucinations and N is the number of slots in the input MR (Nayak et al., 2017; Reed et al., 2018; Wen et al., 2015). Section A.1 in the supplementary materials provides more detail

and examples for each type of semantic error. SER is first calculated on individual utterances and then averaged over the whole test set. For additional insight, we also report the percentage of **semantically perfect outputs** (perfect%), outputs where the SER is 0 and there are no semantic errors. This measure is analogous to the Sentence Error Rate used in speech recognition.

Human TTM Accuracy Evaluation. We evaluated the TTM and the automatic SER calculation with a separate experiment where two NLG experts hand-labelled a random sample of 200 model outputs. Over the 200 samples, the automatic SER was .45 and the human was .46. The overall correlation of the automatic SER with the human SER over all types of errors (D,R,S,H) is .80 and the correlation with deletions, the most frequent error type, is .97.

Retrofit MRs for Self-Training. The TTM is critical for our novel self-training method described in Section 4.2. The retrofit MRs match the (errorful) NLG output: when these MR/NLG output pairs combine attributes from both sources, they provide novel corrected examples to add back into training.

4 Results

We run two sets of experiments. We first run all of the NLG models described in Section 3.1 on the COM test set, and automatically calculate SER and perfect% as described in Section 3.2. We report these results in Section 4.1. Section 4.2 motivates and describes the self-training method and presents the results, resulting in final models that generate semantically perfect outputs 83% of the time.

4.1 Initial Model Results

Model	Training	Test	SER	PERFECT N %	
NOSUP	E2E + NYC	COM	.45	106	3.5%
GUIDE	E2E + NYC	COM	.66	15	0.5%
ATTR	E2E + NYC	COM	.46	167	5.5%
BOOL	E2E + NYC	COM	.45	86	2.8%

Table 1: SER and perfect% on test for each model type on the test of 3040 MRs (COM) that combine attributes from both sources.

Semantic Accuracy. Table 1 summarizes the results across the four models NOSUP, GUIDE, ATTR and BOOL. Overall, the results show that the task, and the COM test set, are surprisingly adversarial. All of the models have extremely high SER, and the SER for NOSUP, ATTR, and BOOL are very similar. Row 2 shows that the GUIDE model has much worse performance than the other models,

in contrast to other tasks (Iyyer et al., 2018). We do not examine the GUIDE model further. Row 3 shows that the ATTR supervision results in the largest percentage of perfect outputs (5.5%).

Model	Training	Test	SER	PERF %
NOSUP	E2E	E2E	.16	19%
NOSUP	E2E + NYC	E2E	.18	15%
NOSUP	NYC	NYC	.06	69%
NOSUP	E2E + NYC	NYC	.06	71%

Table 2: Baseline results for each source on its own test using the NOSUP model. E2E test $N = 630$. NYC test $N = 314$.

The results in Table 1 should be compared with the baselines for testing NOSUP on **only** E2E or NYC in Table 2. Both the E2E and NYC test sets consist of unseen inputs, where E2E is the standard E2E generation challenge test (Dušek et al., 2020), and NYC consists of novel MRs with baseline attribute frequencies matching the training data.⁴ Rows 1 and 3 test models trained on only E2E or only NYC, while Rows 2 and 4 test the same trained NOSUP model used in Row 1 of Table 1 on E2E or NYC test sets respectively. Comparing Rows 1 and 2 shows that training on the same combined data used in Table 1 slightly degrades performance on E2E, however, this SER is still considerably lower than the .45 SER for the NOSUP model tested on the COM test set, shown in the first row of Table 1. Row 4 shows that the NOSUP model trained on the combined data appears to improve performance on the NYC test because the perfect% goes up from 69% in Row 3 to 71%. The SER of .06 shown in Row 4 should also be compared to the .45 SER reported for the NOSUP model in the first row of Table 1. These results taken together establish that the combined MRs in the COM test provide a very different challenge than the E2E and NYC unseen test inputs.

However, despite the poor performance of the initial models, we hypothesized that there may be enough good outputs to experiment with self-training. Since the original training data had no combined outputs, decoding may benefit from even small numbers of training items added back in self-training.

Human Evaluation.

The automatic SER results

⁴Previous work on the E2E dataset has also used seq2seq models, with SOA results for SER of 1% (Dušek et al., 2020), but here we do not use the full training set. Our partition of the NYC dataset has not been used before, but experiments on comparable NYC datasets have SERs of .06 and .02 (Reed et al., 2018; Harrison et al., 2019).

Model	NAT.	COHER.	GRAMMAT.
NOSUP	4.04	4.13	4.12
ATTR	4.11	4.25	4.14
BOOL	3.97	4.18	4.25
AGREEMENT	.63	.62	.65

Table 3: Human Evaluation for NOSUP ($N = 100$) ATTR ($N = 100$) and BOOL ($N = 86$) for Naturalness, Semantic Coherence, and Grammaticality

provide insight into the semantic accuracy of the models, but no assessment of other aspects of performance. We thus conduct a human evaluation on Mechanical Turk to qualitatively assess fluency, coherency and grammaticality. We use the automatic SER to select 100 semantically perfect references from the NOSUP and the ATTR models’ test outputs, and the 86 perfect references from BOOL. We ask 5 Turkers to judge on a scale of 1 (worst) to 5 (best) whether the utterance is: (1) fluent and natural; (2) semantically coherent; and (3) grammatically well-formed. Table 3 reports the average score for these qualitative metrics as well as the Turker agreement, using the average Pearson correlation across the Turkers. The results show that the agreement among Turkers is high, and that all the models perform well, but that the ATTR model outputs are the most natural and coherent, while the BOOL model outputs are the most grammatical.

4.2 Self-Training

In order to conduct self-training experiments, we need perfect outputs that combine attributes from both sources to add back into training. These outputs must also be natural, coherent and grammatical, but Table 3 shows that this is true of all the models. A key idea for our novel self-training method is that the TTM (Section 3.2) automatically produces “retrofit” corrected MRs that match the output texts of the NLG models. Thus we expect that we can construct more perfect outputs for self-training by using retrofitting than those in Table 1. Here, we first analyse the outputs of the initial models to show that self-training is feasible, and then explain our method and present results.

Error Analysis. An initial examination of the outputs suggests that the models simply have trouble combining attributes from both sources. We provide examples in Table 10 in Section A.2 in the supplementary materials. To quantify this observation, we define a metric, Source Blending Rate (SB), that counts the percentage of outputs that combine attributes from both sources, whether or

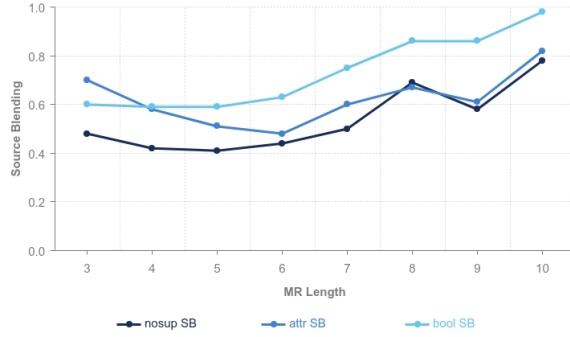


Figure 5: Source Blending Rate (SB) as a function of MR length for NOSUP, ATTR and BOOL.

not the attribute values are accurate:

$$SB = \frac{R_{sb}}{N}$$

where R_{sb} is the count of references r that contain an attribute $a_i \subseteq source_1$ and another attribute $a_j \subseteq source_2$, and N is the total number of references. Only attributes that appear uniquely in each source are included in the a_i, a_j : the unique attributes are illustrated in Figure 2.

Figure 5 graphs SB as a function of MR length showing that indeed the models **do** in many cases produce combined outputs and that the type of model supervision greatly influences SB. The NOSUP model is the worst: a fact that is masked by the NOSUP model’s SER in Table 1, which appears to be on a par with both ATTR and BOOL. Interestingly, all models are more likely to produce an SB output as the MRs get longer, but Figure 5 shows clearly that the BOOL model especially excels.

For self-training, we also need a model that generates utterances with the RECOMMEND dialogue act. As mentioned in Section 2, recommendations increase both semantic and syntactic complexity. Half the test items contain a recommendation, so we need a model that can produce them. Table 4 presents results for SER and SB depending on whether a RECOMMEND was in the MR, showing that the three models vary a great deal. However, the BOOL row for the SB column shows that when the MR includes a recommendation, the BOOL model produces a combined output far more frequently than NOSUP or ATTR ($SB = .73$).

Thus Figure 5 and Table 4 show that the BOOL model produces the most combined outputs. After TTM extraction, the BOOL model provides the most instances (1405) of retrofit MR/output pairs to add to self-training, and we therefore use BOOL in the self-training experiments below.

Retrofitting MRs for Self-Training. Table 5 illus-

Model	SER		SB	
	REC	NO-REC	REC	NO-REC
NOSUP	.43	.46	.44	.56
ATTR	.51	.41	.36	.77
BOOL	.47	.43	.73	.67

Table 4: Effect of the RECOMMEND dialogue act on Slot Error Rate (SER) and Source Blending (SB) for the three types of model supervision: NOSUP, ATTR and BOOL.

trates how the TTM works, and shows that it can effectively create a new MR that may not have been previously seen in training, allowing the model to **learn from its mistakes**. The caption for Table 5 explains in detail the retrofitting process and how it leads to new examples to use in self-training.

It is important to note that the retrofit MRs for some NLG outputs **cannot** be used for self-training. NLG model outputs whose semantic errors include repetitions can **never** be used in self-training, because valid MRs do not include repeated attributes and values, and the method doesn’t edit the NLG output string. However, deletion errors cause no issues: the retrofit MR simply doesn’t have that attribute. Substitutions and hallucinations can be used because the retrofit MR substitutes a value or adds a value to the MR, as long as the realized attribute value is valid, e.g. “friendly food” is not a valid value for *food quality*.^{5,6}

Experiments. To begin the self-training experiments, we apply the source-blending metric (SB) defined above to identify candidates that combine attributes from both sources, and then apply the TTM to construct MRs that match the NLG model outputs, as illustrated in Table 5, eliminating references that contain a repetition. We start with the same combined 76,832 training examples and the 1405 retrofit MR/NLG outputs from the BOOL model. We explore two bootstrapping regimes, depending on whether a model output is a repetition of one that we have already seen in training. One model keeps repetitions and adds them back into training, which we dub S-Repeat, and the other model only adds unique outputs back into training, which we dub S-Unique.

Quantitative Results. Figure 6 shows how the SER and perfect% continuously improve on the

⁵We applied the human evaluation in Section 3.2 to instances included in self-training: the correlation between human judgements and the automatic SER is .95, indicating that the retrofit MRs are highly accurate.

⁶Table 10 in Section A.2 provides additional examples of errorful outputs that **can** or **cannot** be used in self-training.

Original MR	Text-to-MR	OUTPUT
name[RESTAURANT], cuisine[fastfood], decor[good], qual[fantastic], location[riverside], price[cheap], eatType[pub], familyFriendly[no]	name[RESTAURANT], cuisine[fastfood], qual[good], location[riverside], familyFriendly[no]	[RESTAURANT] is a fast food restaurant located in the riverside area. it has good food and it is not family friendly.
name[RESTAURANT], recommend[yes], cuisine[fastfood], qual[good], location[riverside], familyFriendly[no]	name[RESTAURANT], cuisine[fastfood], qual[good], location[riverside], familyFriendly[no]	[RESTAURANT] is a fast food restaurant in the riverside area. it is not family friendly and has good food.

Table 5: Examples to show retrofitting. The examples start from different original MRs (col 1), but yield the same MR after text-to-MR extraction (col 2). In Row 1, the model output in column 3 deleted the attributes *price*, *decor* and *eat type* (pub), and substituted the value “good” for “fantastic” for the quality attribute. In Row 2 the model deleted the RECOMMEND dialogue act, but otherwise realized the original MR correctly. At test time, the original MRs produced different outputs (col 3). Thus the retrofitting yields two unique novel instances for self-training.

COM test set for S-Repeat over 10 rounds of self-training, and that S-Repeat has better performance, indicating that adding multiple instances of the same item to training is useful. The performance on the COM test set of the S-Unique model flattens after 8 rounds. After 10 rounds, the S-Repeat model has an SER of .03 and produces perfect outputs 82.9% of the time, a 77.4 percent absolute improvement over the best results in Table 1.



Figure 6: SER and perfect% on the COM test set for S-Repeat vs. S-Unique during self-training

COM-2 Test Set. Since the the self-training procedure used the COM test set during self-training, we construct a new test with 3040 novel MRs using the procedure described in Section 2, which we call COM-2. First we test the initial models on COM-2, resulting in a best SER of 0.45 for the BOOL model, identical with the result for COM. For perfect% the best result was 5.3% on the ATTR model, which is again comparable to the original COM test set. We then tested the final self-trained model on COM-2, with the result that the SER for S-Repeat (0.03) and S-Unique (0.11) are again identical to the result for COM. The perfect% is comparable to that reported in Figure 6; it decreases by 2.2% for S-Repeat to 80.7% and increases by .2% for S-Unique to 50.7%. Overall, the performance on COM-2 improved by

an absolute 75.4%.

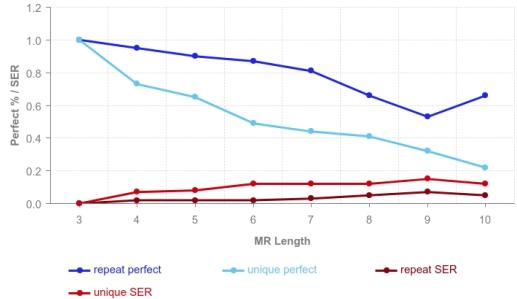


Figure 7: SER and perfect% on COM-2 as a function of MR length for BOOL supervision before self-training and for the S-Repeat model after self-training.

Figure 7 shows that the results improve, not only overall, but also by MR length. It plots the SER and perfect% results, by MR length, for the BOOL model before and after self-training. While the perfect% decreases as the number of attributes increase, there is a large improvement over the initial model results. Also, after self-training the worst perfect% is still above 0.5, which is higher than perfect% for any MR length before self-training. The SER also improves over all MR lengths after self-training, not exceeding .06, significantly better than even the shortest MR before self-training.⁷

Human Evaluation. We also performed a human

Model	NAT.	COHER.	GRAMMAT.
S-REPEAT	3.99	4.08	4.02
S-UNIQUE	4.06	4.13	4.14
AGREEMENT	.57	.61	.57

Table 6: Human Evaluation on Mechanical Turk for S-Repeat (N = 100) and S-Unique (N = 100) for Naturalness, Semantic Coherence, and Grammaticality

⁷Performance results for the self-trained model on the original E2E and NYC test sets in supplement A.3 shows that performance also improves on the E2E and NYC test sets.

evaluation on Mechanical Turk to assess the qualitative properties of the model outputs after self-training. We selected 100 perfect references for S-Repeat and 100 for S-Unique and used the same HIT as described in Section 4.1. Table 6 reports the average score for these qualitative metrics as well as the Turker agreement, using the average Pearson correlation across the Turkers. The results show that naturalness, coherence and grammaticality are still high after self-training for both models, but that the S-Unique model produce better outputs from a qualitative perspective. We believe we could improve the self-training method used here with additional referenceless evaluation metrics that aim to measure naturalness and grammaticality (Mehri and Eskenazi, 2020). We leave this to future work.

#	Realization
1	[RESTAURANT] is the best place because it is a family friendly pub with good decor and good food .
2	[RESTAURANT] is a family friendly restaurant with bland food and is in the low price range. It is the best restaurant .
3	[RESTAURANT] is a family friendly coffee shop with decent service and a low customer rating . It is in the £20-25 price range.
4	[RESTAURANT] is the best restaurant because it is in the east village, it is near [POINT-OF-INTEREST] with great service and it is affordable.

Table 7: Example outputs with source blending. NYC attributes are represented using red and E2E attributes are represented using blue

Qualitative and Linguistic Analysis. Table 7 provides outputs from the models that display different ways of combining attributes from the original sources. In Row 1 we can see that the RECOMMEND dialogue act from NYC can be combined in the same sentence as the attributes *family friendly* and *eat type* from E2E and aggregate these E2E attributes with NYC attributes *decor* and *food quality* using a “with” operator. Row 2 shows another example where the NYC and E2E attributes are joined using a “with” operator. In Row 3 there is a single sentence with four attributes where the NYC attribute is preceded and followed by E2E attributes. Row 4 concatenates the two sources in a single sentence using sentence coordination. The “east village” location from the NYC dataset, is concatenated with the attributes *near* from E2E and *service* from NYC. These examples show that the NLG models can combine attributes from both sources in many different ways. Table 11 in Section A.4 provides additional detail by providing

examples along with their corresponding MRs.

5 Conclusion

This paper presents the first experiments on training an NLG for an extended domain ontology by re-using existing within-domain training data. We show that we can combine two training datasets for the restaurant domain, that have different ontologies, and generate output that combines attributes from both sources, by applying a combination of neural supervision and a novel self-training method. While it is common practice to construct test sets with unseen attribute combinations, we know of no prior work based on constructing a new combined ontology. Our experiments show that the task is surprisingly adversarial, consistent with recent work suggesting that neural models often fail to generalize (Wallace et al., 2019; Feng et al., 2018; Ribeiro et al.; Goodfellow et al., 2014). Work on domain transfer shares similar goals to the experiments presented here (Wen et al., 2016; Golovanov et al., 2019), but these methods do not produce NLG outputs that integrate attributes from two different sources into the same sentence. Our final results show that the ability of our self-training method to automatically construct new training instances results in high quality natural, coherent and grammatical outputs with high semantic accuracy.

In future, we hope to generalize our novel self-training method to build an NLG that can combine two distinct domains, e.g. hotels or movies combined with restaurants in multi-domain dialogue (Budzianowski et al., 2018; Gašić et al., 2015; Hakkani-Tür et al., 2016; Cervone et al., 2019; Ultes et al., 2017). Ideally systems that cover multiple domains should be able to produce utterances that seamlessly integrate both domains, if data exists for each domain independently. However, there may be additional challenges in such combinations. Our results require the initial neural models to generate **some** combined outputs. It is not clear whether there are some aspects of our experimental setup that facilitate this, e.g. it may require some attributes to be shared across the two initial ontologies, or some shared vocabulary. Thus it is possible that initial models for two more distinct domains may not produce any combined outputs, and it may be necessary to seed the self-training experiments with a small number of combined training instances. We leave these issues to future work.

References

- Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of nlg systems. In *EACL*.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Ifigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.
- Alessandra Cervone, Chandra Khatri, Rahul Goel, Behnam Hedayatnia, Anu Venkatesh, Dilek Hakkani-Tur, and Raefer Gabriel. 2019. Natural language generation at scale: A case study for open domain question answering. In *arXiv preprint arXiv:1903.08097*.
- Yun-Nung Chen, Asli Celikyilmaz, and Dilek Hakkani-Tür. 2017. Deep learning for dialogue systems. *Proceedings of ACL 2017, Tutorial Abstracts*, pages 8–14.
- Ondrej Dusek, Jekaterina Novikova, and Verena Rieser. 2017. Referenceless quality estimation for natural language generation. *CoRR*, abs/1708.01759.
- Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.
- Shi Feng, Eric Wallace, Pedro Grissom II, Alvin Rodriguez, Mohit Iyyer, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretation difficult. In *Empirical Methods in Natural Language Processing*.
- Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, page 94–104. Association for Computational Linguistics.
- M Gašić, N Mrkšić, Pei-hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. Policy committee for adaptation in multi-domain spoken dialogue systems. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 806–812. IEEE.
- Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.
- Sergey Golovanov, Rauf Kurbanov, Sergey Nikolenko, Kyryl Truskovskyi, Alexander Tselousov, and Thomas Wolf. 2019. Large-scale transfer learning for natural language generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6058.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*.
- Vrindavan Harrison, Lena Reed, Shereen Oraby, and Marilyn Walker. 2019. Maximizing stylistic control and semantic accuracy in nlg: Personality variation and discourse contrast. In *Proceedings of the 1st Workshop on Discourse Structure in Neural NLG*, pages 1–12.
- Vrindavan Harrison and Marilyn Walker. 2018. Neural generation of diverse questions using answer focus, contextual and linguistic features. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 296–306.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885.
- Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn Walker. 2018. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 152–162.
- Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323.
- Chris Kedzie and Kathleen McKeown. 2019. A good sample is hard to find: Noise injection sampling and self-training for neural language generation models. In *Proceedings of the 12th International Conference on Natural Language Generation*.

- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.
- Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.
- Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.
- Shikib Mehri and Maxine Eskenazi. 2020. Unsupervised evaluation of interactive dialog with dialogpt. In *Proc. of the SIGDIAL 2020*.
- Neha Nayak, Dilek Hakkani-Tur, Marilyn Walker, and Larry Heck. 2017. To plan or not to plan? discourse planning in slot-value informed sequence to sequence models for language generation. In *Proc. of Interspeech 2017*.
- J. Novikova, O. Dušek, and V. Rieser. 2017a. The e2e dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Conference*.
- Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017b. Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252.
- Shereen Oraby, Lena Reed, Shubhangi Tandon, TS Sharath, Stephanie Lukin, and Marilyn Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. In *SIGDIAL*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.
- Lena Reed, Shereen Oraby, and Marilyn Walker. 2018. Can neural generators for dialogue learn sentence planning and discourse structuring? In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 284–295.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page 35–40. Association for Computational Linguistics.
- Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, volume 3, pages 41–51.
- Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. 2019. Pragmatically informative text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4060–4067.
- Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *Computational Linguistics and Intelligent Text Processing: 6th International Conference, CICLing 2005, Mexico City, Mexico, February 13-19, 2005, Proceedings*, volume 3406, page 341. Springer.
- Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentation in spoken dialogue systems. In *Meeting of the Association for Computational Linguistics*.
- Amanda Stent, Marilyn Walker, Steve Whittaker, and Preetam Maloor. 2002. User-tailored generation for spoken dialogue: An experiment. In *ICSLP*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Ye Tian, Ioannis Douratsos, and Isabel Groves. 2018. Treat the system like a human student: Automatic naturalness evaluation of generated text without reference texts. *INLG 2018*, page 109.
- Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, et al. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78.
- Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 Conference on Empirical Methods*

in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2153–2162.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–129.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. *CoRR*, abs/1707.08052.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the voice of a sentence in japanese-to-english neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210.

A Supplementary Materials: Learning from Mistakes: Combining Ontologies via Self-Training for Dialogue Generation

A.1 Types of Semantic Errors

The TTM is tuned to identify 4 common neural generation errors: *deletions* (failing to realize a value), *repetitions* (repeating an attribute), *substitutions* (mentioning an attribute with an incorrect value), and *hallucinations* (introducing an attribute that was not in the original MR at all).

Table 9 illustrates each of these types of semantic errors. Row 1 shows deletions of *cuisine*, *price* and *near* which are in the MR but not in the realization. Row 2 demonstrates a repetition, where *location* and *decor* are both repeated. *Decor* is realized with two different lexical values, “good ambiance” and “good decor”. There is a substitution in Row 3 where the MR states that the *food quality* is “bad”, but *food quality* is realized as “good”. Finally, Row 4 has a hallucination, *service* is not in the MR but it is in the second sentence of the realization.

A.2 Example Errorful NLG Model Outputs

Table 10 provides examples of NLG model output utterances with high SERs. It illustrates how the NLG models struggle to combine attributes from the two ontologies which is required by all the input MRs (Column SB). It also illustrates cases where it is not possible to produce a valid retrofit MR that can be added back into training during self-training (Column Valid). In most cases these are due to many repetitions. Row 1 is an example where there is no source blending and since it has a repetition (*price*) it cannot be used for self-training (valid = no). Row 1 also illustrates an ungrammatical realization of *price* which we have no way to automatically detect at present *it is in the high price*. Row 2 has three deletions as well as two repetitions. The output repeats *It is in midtown* three times in a row. Row 3 has five errors, it does not realize the dialogue act RECOMMEND and has deleted three other attributes and it hallucinates *food quality*. While this is a significant number of errors, this realization can still be used in self-training, since none of its errors are repetitions. Row 4 has all four types of errors. It deletes *cuisine*, *decor* and *service*, it realizes a value for *family friendly* twice with different values, a substitution and finally it hallucinates *food quality*. Row 5 actually has more errors than slots. It deletes all but two of its at-

tributes: *name* and *rating*. It also hallucinates *food quality* and repeats *rating*.

Model	Training	Test	SER	PERF %
BOOL	S-REPEAT	E2E	.14	25%
BOOL	S-REPEAT	NYC	.05	77%

Table 8: Performance of the self-trained S-Repeat model on the original E2E and NYC test sets. E2E test N = 630. NYC test N = 314.

A.3 Performance on E2E and NYC test sets

Table 2 provided a baseline for NOSUP’s performance before self-training on the original test sets for E2E and NYC. We also verify that the self-trained model performs well after self-training. Table 8 shows that self-training improves the results for the original E2E and NYC test sets.

A.4 Example Final Model Outputs

Table 11 provides outputs from the final iteration of self-training that display different ways of combining different attributes from the ontologies. Row 1 shows that the model can combine attributes from the two sources in the same sentence, with attributes from each source, *decor* and *rating*, appearing in a single sentence with *and*. Row 2 shows a different way of combining attributes from the two sources, with *family friendly* and *food quality*, in a single sentence, this time using *with*. In Row 3 we can see that the model can also generate complex sentences for recommendations using the marker *because*. Also, the attribute used in the *because* clause is from E2E i.e. *family friendly* but such sentences never appear in the original E2E training data. The last row shows a complex sentence where *decor* is combined with *eat type* and *customer rating*, again a novel combination.

Error Type	MR	Realization
Delete CUISINE, PRICE, NEAR	name[RESTAURANT], cuisine[mexican], location[midtown], price[expensive], eatType[coffee shop], familyFriendly[no], near[POINT-OF-INTEREST]	[RESTAURANT] is a coffee shop that is not family friendly. It is located in Midtown.
Repeat LOCATION, DECOR	name[RESTAURANT], decor[good], location[midtown west], eatType[coffee shop], rating[1 out of 5]	[RESTAURANT] is a coffee shop in Midtown West with good ambiance. It is in Midtown West with good decor.
Substitution QUALITY BAD to GOOD	name[RESTAURANT], decor[good], qual[bad], location[tribeca/soho], eatType[pub]	[RESTAURANT] is in Tribeca/Soho with good food and good decor. It is a pub.
Hallucination SERVICE	name[RESTAURANT], decor[good], qual[good], location[riverside], near[POINT-OF-INTEREST]	[RESTAURANT] is near [POINT-OF-INTEREST] in the riverside area. It has good food, good decor and good service.

Table 9: Model outputs illustrating types of semantic errors.

#	SER	Valid	SB	MR	Realization
1	.60	no	no	name[RESTAURANT], decor[fantastic], qual[acceptable], price[high], familyFriendly[yes]	[RESTAURANT] is a child friendly restaurant with french food and it is in the high price range. it is in the high price.
2	.63	no	no	name[RESTAURANT], recommend[yes], cuisine[japanese], decor[bad], location[midtown], service[fantastic], rating[low], near[POINT-OF-INTEREST]	[RESTAURANT] is the best restaurant since it is a japanese restaurant with bad ambiance and it is in midtown. it is in midtown. it is in midtown.
3	.71	yes	yes	name[RESTAURANT], recommend[yes], decor[fantastic], service[acceptable], eatType[pub], rating[high], near[xnear]	[RESTAURANT] is a pub with good food and it is located near the xnear.
4	.75	no	yes	name[RESTAURANT], cuisine[southern], decor[fantastic], location[city centre], price[cheap], service[good], eatType[coffee shop], familyFriendly[no]	[RESTAURANT] is a cheap, family friendly coffee shop with good food. it is in the city centre. it is not family friendly.
5	1.17	no	yes	name[RESTAURANT], cuisine[indian], decor[fantastic], service[good], rating[high], near[POINT-OF-INTEREST]	[RESTAURANT] provides good food and has great customer rating and has great customer rating .

Table 10: Example pathological outputs with high SERs from the NLG models before self-training. Valid realizations (col 3) are defined as those usable for self-training. In the MRs, the NYC attributes are represented using typewriter font and E2E attributes are represented using *italics*.

#	MR	Realization
1	name[RESTAURANT],cuisine[Indian], decor[fantastic],qual[good], service[good], rating[high], near[POINT-OF-INTEREST]	[RESTAURANT] is a Indian restaurant with great service and excellent ambiance and a high customer rating. It is located near the [POINT-OF-INTEREST].
2	name[RESTAURANT], recommend[yes], decor[good], qual[good], price[cheap], service[bad], familyFriendly[yes],	[RESTAURANT] is a family friendly restaurant with good food, good ambiance and bad service. It is in the low price range.
3	name[RESTAURANT], recommend[yes], decor[bad], qual[good], location[flatiron/union square], price[cheap], service[acceptable], eatType[coffee shop], rating[3 out of 5],	[RESTAURANT] is the best restaurant because it is a family friendly coffee shop with good food, friendly service and bad ambiance. It is in Flatiron/Union Square. It has a customer rating of 3 out of 5. It is cheap.
4	name[RESTAURANT], recommend[yes], cuisine[mediterranean], decor[fantastic], price[very expensive], eatType[pub], rating[5 out of 5] .	[RESTAURANT] is a Mediterranean pub with excellent ambiance and a customer rating of 5 out of 5. It is in the upper price range. It is the best restaurant.

Table 11: Example outputs of source blending from final self-training iterations. In the MRs, the NYC attributes are represented using typewriter font and E2E attributes are represented using *italics*.

TripPy: A Triple Copy Strategy for Value Independent Neural Dialog State Tracking

Michael Heck, Carel van Niekerk, Nurul Lubis,
Christian Geishauser, Hsien-Chin Lin, Marco Moresi, Milica Gašić

Heinrich Heine University Düsseldorf, Germany

{heckmi, niekerk, lubis, geishaus, linh, moresi, gasic}@hhu.de

Abstract

Task-oriented dialog systems rely on dialog state tracking (DST) to monitor the user’s goal during the course of an interaction. Multi-domain and open-vocabulary settings complicate the task considerably and demand scalable solutions. In this paper we present a new approach to DST which makes use of various copy mechanisms to fill slots with values. Our model has no need to maintain a list of candidate values. Instead, all values are extracted from the dialog context on-the-fly. A slot is filled by one of three copy mechanisms: (1) Span prediction may extract values directly from the user input; (2) a value may be copied from a system inform memory that keeps track of the system’s inform operations; (3) a value may be copied over from a different slot that is already contained in the dialog state to resolve coreferences within and across domains. Our approach combines the advantages of span-based slot filling methods with memory methods to avoid the use of value picklists altogether. We argue that our strategy simplifies the DST task while at the same time achieving state of the art performance on various popular evaluation sets including Multiwoz 2.1, where we achieve a joint goal accuracy beyond 55%.

1 Introduction

The increasing popularity of natural language human-computer interaction urges the development of robust and scalable task-oriented dialog systems. In order to fulfill a user goal, a dialogue system must be capable of extracting meaning and intent from the user input, and be able to keep and update this information over the continuation of the dialog (Young et al., 2010). This task is called dialog state tracking (DST). Because the next dialog system action depends on the current state of the conversation, accurate dialog state tracking (DST) is absolutely vital.

U:	i'm looking for an expensive restaurant in the center of town.
S:	there are 33 restaurants [...]. would you like to narrow your search by type of food?
U:	i do not care about food. surprise me.
S:	fitbillies restaurant serves british food. [...]. would you like to hear about any others? [...].
U:	that sounds fine! can you book it please and get me the reference number?
S:	sure , what day and time would you like and how many people ?
U:	i would like a table for 5 at 11:30 on tuesday [...].
S:	okay, the booking was successful. [...]. is there anything else i can help you with?
U:	i'm also looking for a place to stay. it needs [...] free wifi and [be] in the same area as the restaurant .

Turn	Domain-slot pair	Value	Type	Coreference
0	restaurant-pricerange	expensive	span	
0	restaurant-area	center	span	
1	restaurant-food	< don'tcare >	(don'tcare)	
2	restaurant-name	fitbillies	informed	
3	restaurant-people	5	span	
3	restaurant-book_time	11:30	span	
3	restaurant-book_day	tuesday	span	
4	hotel-internet	< true >	(bool)	
5	hotel-area	center	coreference (multiturn)	restaurant-area

Figure 1: Example dialog in MultiWOZ.

DST is tasked to extract from the user input information on different concepts that are necessary to complete the task at hand. For example, in order to recommend a restaurant to a user, the system needs to know their preferences in terms of price, location, etc. These concepts are encapsulated in an ontology, where dialogue domain (e.g., restaurant or hotel), slot (e.g., price range or location), and value (e.g. cheap or expensive) are defined. Solving this information extraction task is prerequisite for forming a belief over the dialog state.

Traditional approaches to DST operate on a fixed ontology and perform prediction over a pre-defined set of slot-value pairs (Mrkšić et al., 2016; Liu and Lane, 2017; Zhong et al., 2018). Such approaches perform very well on datasets which are defined over fairly small ontologies. Apply these methods to more complex datasets however reveals various limitations (Ren et al., 2018; Nouri and Hosseini-Asl, 2018). First, it is often difficult to obtain a complete ontology for a task. Second, slot-value pairs that were outside the ontology or the training data are impossible to capture during test time.

Third, such methods at best scale linearly with the size of the ontology. Most importantly, the idea of fixed ontologies is not sustainable, as in real world applications they are subject to constant change.

Human-computer interactions often need to be defined over multiple domains at the same time, ideally with unrestricted vocabulary. Recent approaches to multi-domain and open-vocabulary DST extract values from the dialog context directly by predicting value spans in the input (Gao et al., 2019; Chao and Lane, 2019; Kim et al., 2019; Zhang et al., 2019). Span prediction is a demonstrably potent method to detect relevant information in utterances, but its major drawback is that it only suits *extractive* values that are explicitly expressed as a sequence of tokens. This is the reason why span-based methods benefit from the support of a picklist, i.e., a list of value candidates from which a system can choose. Still, these methods fall short when handling nuanced and subtle phenomena that often occur in natural conversations, such as coreference and value sharing (“I’d like a hotel in the same area as the restaurant.”), and implicit choice (“Any of those is ok.”).

In this work, we propose a new approach to value independent multi-domain DST:

1. In addition to extracting values directly from the user utterance via span prediction and copy, our model creates and maintains two memories on-the-fly, one for system inform slots, and one for the previously seen slots.
2. The *system inform memory* solves the implicit choice issue by allowing copy mechanism from concepts mentioned by the system, e.g., values that are offered and recommended.
3. The *DS memory* allows the use of values already existing in the dialogue state to infer new values, which solves the coreference and value sharing problems.

We call this approach **TripPy**, **Triple copy** strategy DST.¹ Our experiments results show that our model is able to handle out-of-vocabulary and rare values very well during test time, demonstrating good generalization. In a detailed analysis we take a closer look at each of the model’s components to study their particular roles.

¹Our code will be released upon publication of this work.

2 Related Work

Dialog state tracking has been of broad interest to the dialog research community, which is reflected by the existence of a series of DST challenges (Henderson et al., 2014; Rastogi et al., 2019). These challenges consistently pushed the boundaries of DST performance. Current state-of-the-art has to prove to work on long, diverse conversations in multiple domains with a high slot count and principally unrestricted vocabulary (Eric et al., 2019). Dialogs of such complex nature are tough for traditional approaches that rely on the availability of a candidate list due to scalability and generalization issues (Mrkšić et al., 2016; Liu and Lane, 2017; Ramadan et al., 2018; Rastogi et al., 2017).

Span-based approaches recently alleviated both problems to some extent. Here, slot values are extracted from the input directly by predicting start and end positions in the course of the dialog. For instance, Xu and Hu (2018) utilizes an attention-based recurrent network with a pointer mechanism to extract values from the context. This extractive approach has its limitations, since many expressible values are not found verbatim in the input, but rather mentioned implicitly, or expressed by a variety of rephrasings.

With the assistance of contextual models such as BERT (Devlin et al., 2018), issues arising from expressional variations can be mitigated. Recent work has demonstrated that encoding the dialog context with contextual representations supports span prediction to generalize over rephrasings. SUMBT (Lee et al., 2019) utilizes BERT to encode slot IDs and candidate values and learns slot-value relationships appearing in dialogs via an attention mechanism. Dialog context is encoded with recurrence. BERT-DST (Chao and Lane, 2019) employs contextual representations to encode each dialog turn and feeds them into classification heads for value prediction. The dialog history, however, is not considered for slot filling. In Gao et al. (2019), DST is rendered as a reading comprehension task that is approached with a BERT-based dialog context encoder. A slot carryover prediction model determines whether previously detected values should be kept in the DS for the current turn.

An alternative to span prediction is value generation. TRADE (Wu et al., 2019) and MA-DST (Kumar et al., 2020) generate a DS from the input using a copy mechanism to combine the distributions over a pre-defined vocabulary and the vocabulary

of current context. SOM-DST (Kim et al., 2019) applies a similar mechanism for value generation, but takes the previous dialog turn as well as the previous DS as input to BERT to predict the current DS. A state operation predictor determines whether a slot actually needs to be updated or not. The downside of generative models is that they tend to produce invalid values, for instance by word repetitions or omissions.

Recently, a hybrid approach called DS-DST has been proposed that makes use of both span-based and picklist-based prediction for slot-filling (Zhang et al., 2019). In contrast to generative approaches, picklist-based and span-based methods use existing word sequences to fill slots. DS-DST somewhat alleviates the limitations of span prediction by filling a subset of slots with a picklist method instead.

Recent works seemed to reveal a trade-off between the level value independence in a model and the DST performance. Chao and Lane (2019) and Gao et al. (2019) solely rely on span-prediction, but their performance lacks behind methods that at least partially rely on a pre-defined list of candidate values. This has impressonably been demonstrated by Zhang et al. (2019). Their model could not compete when relying on span-prediction entirely. In contrast, when relying solely on their picklist slot-filling method, they achieved the to-date best performance on MultiWOZ 2.1. The proposed dual-strategy approach lies favorably between these two extremes.

To the best of our knowledge, none of the recent approaches to complex DST tasks such as MultiWOZ (Budzianowski et al., 2018; Eric et al., 2019) are value independent in the strict sense. What’s more, they tremendously benefit from the use of a value candidate list. Our work tackles this limitation by introducing a triple copy strategy that relies on span-prediction as well as memory mechanisms. In contrast to other hybrid approaches such as Zhang et al. (2019), our memory mechanisms create candidate lists of values on-the-fly with the dialog context as only source of information, thus avoiding the use of pre-defined picklists. We let the model decide which strategy to choose for each slot at each turn. Our approach differs from Chao and Lane (2019) and Kim et al. (2019) in that we consider the dialog history as context in addition to the current turn. We also differ from approaches like Lee et al. (2019) since we do not employ recurrence. Like Kim et al. (2019), we use auxiliary

inputs at each turn, but we do so as a late feature fusion strategy. With our slot-value copy mechanism to resolve coreferring value phrases, we employ a method which is reminiscent of Gao et al. (2019)’s slot carryover, but with the sharp distinction that we copy values between different slots, facilitating value sharing within and across domains.

3 TripPy: Triple Copy Strategy for DST

Our model expects the following input format to perform dialog state tracking. Let $X = \{(U_1, M_1), \dots, (U_T, M_T)\}$ be the sequence of turns that comprise a dialog of length T . U_t is the user utterance at turn t , M_t is the system utterance that precedes the user utterance. The task of the model is (1) to determine for every turn whether any of the N domain-slot pairs in $S = \{S_1, \dots, S_N\}$ is present, (2) to predict the values for each S_n and (3) to track the dialog state DS_t over the course of the dialog, i.e., for $t \in [1, T]$.

We employ a triple-copy strategy to fill the slots. The intuition is that values are either explicitly expressed by the user, that they are expressed by the system and referred to by the user via confirmation or rejection, or that they have been expressed earlier in the dialog as assignment to another domain-slot pair (coreference). Each of these cases is handled by one of three copy mechanisms. It becomes apparent that slots can not be filled by exclusively resorting to one particular copy method. Therefore, we employ slot gates that determine at each turn which method to use to fill the respective slot.

Figure 2 depicts our model. We encode the dialog context with a BERT front-end and feed-forward the resulting contextual representations to various classification heads to solve the sub-tasks for DST. The aggregate sequence representation is the input to the slot gates. The sequence of token representations is the input to the span predictors.

3.1 Context Encoder

We use BERT (Devlin et al., 2018) as front-end to encode at each turn t the dialog context as

$$R_t = \text{BERT}([\text{CLS}] \oplus U_t \oplus [\text{SEP}] \oplus M_t \oplus [\text{SEP}] \oplus H_t \oplus [\text{SEP}]), \quad (1)$$

where $H_t = (U_{t-1}, M_{t-1}), \dots, (U_1, M_1)$ is the history of the dialog up to and excluding turn t . The special token [CLS] precedes every input sequence to BERT, and [SEP] separates portions of the input sequence. It is then $R_t = [r_t^{\text{CLS}}, r_t^1, \dots, r_t^{\text{seqmax}}]$,

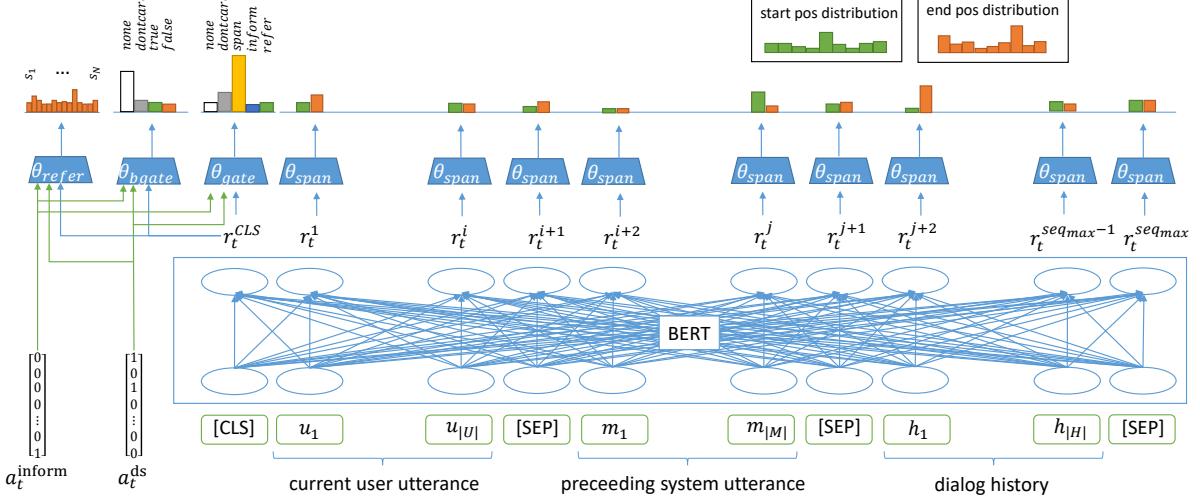


Figure 2: Architecture of our proposed model. TripPy takes the turn and dialog history as input and outputs a DS.

where r_t^{CLS} is a representation of the entire turn including the dialog context H_t . The vectors r_t^1 to $r_t^{seq_{max}}$ are contextual representations for the sequence of input tokens (including special tokens). Both types of representations are used for the following classification tasks.

3.2 Slot Gates

Our model is equipped with a slot gate for each domain-slot pair. This ensures greatest flexibility for multi-domain DST, as there is no restriction as to how many domains might be present in a single turn. At each turn t , slot gates assign each slot S_n to one of the classes in $C = \{none, dontcare, span, inform, refer\}$. The first two labels express special cases. *none* denotes that the slot does not take a value in this turn and *dontcare* states that any value is acceptable for this slot. The remaining three labels each denote one of the model’s copy mechanisms. *span* indicates that a value is present in U_t that can be extracted via span prediction. *inform* indicates that the user refers to a value that has been uttered by the system in M_t . Lastly, *refer* indicates that the user refers to a value that is already present in DS_t .

The input to the slot gates is r_t^{CLS} , and the probability distribution over classes C for domain-slot pair S_n at turn t is $p_{t,s}^{\text{gate}}(r_t^{\text{CLS}}) =$

$$\text{softmax}(W_s^{\text{gate}} \cdot r_t^{\text{CLS}} + b_s^{\text{gate}}) \in \mathbb{R}^5, \quad (2)$$

i.e., each slot gate is realized by a trainable linear layer classification head for BERT.

Boolean slots, i.e., slots that only take binary values, are treated separately. Here, the list of possible

classes is $C_{\text{bool}} = \{none, dontcare, true, false\}$ and the slot gate probability is $p_{t,s}^{\text{bgate}}(r_t^{\text{CLS}}) =$

$$\text{softmax}(W_s^{\text{bgate}} \cdot r_t^{\text{CLS}} + b_s^{\text{bgate}}) \in \mathbb{R}^4. \quad (3)$$

3.3 Span-based Value Prediction

For each slot s that is to be filled via span prediction, a domain-slot specific span prediction layer takes the token representations $[r_t^1, \dots, r_t^{seq_{max}}]$ of the entire dialog context for turn t as input and projects them as follows:

$$[\alpha_{t,i}^s, \beta_{t,i}^s] = W_s^{\text{span}} \cdot r_t^i + b_s^{\text{span}} \in \mathbb{R}^2 \quad (4a)$$

$$p_{t,s}^{\text{start}} = \text{softmax}(\alpha_{t,i}^s) \quad (4b)$$

$$p_{t,s}^{\text{end}} = \text{softmax}(\beta_{t,i}^s) \quad (4c)$$

$$\text{start}_t^s = \text{argmax}(p_{t,s}^{\text{start}}) \quad (4d)$$

$$\text{end}_t^s = \text{argmax}(p_{t,s}^{\text{end}}). \quad (4e)$$

Each span predictor is realized by a trainable linear layer classification head for BERT, followed by two parallel softmax layers to predict start and end position. Note that there is no special handling for erroneously predicting $\text{end}_t^s < \text{start}_t^s$. In practice, the resulting span will simply be empty.

3.4 System Inform Memory for Value Prediction

The system inform memory $I_t = \{I_t^1, \dots, I_t^N\}$ keeps track of all slot values that were informed by the system in dialog turn t . A slot in DS_t needs to be filled by an informed value, if the user positively refers to it, but does not express the value such that span prediction can be used. E.g., in Figure 1 the

slot gate for domain-slot `<restaurant, name>` should predict *inform*. The slot is filled by copying the informed value into the dialog state, i.e., $DS_t^s = I_t^s$, where i is the index of the respective domain-slot.

3.5 DS Memory for Coreference Resolution

The more complex a dialog can be, the more likely it is that coreferences need to be resolved. For instance, the name of a restaurant might very well be the destination of a taxi ride, but the restaurant might not be referred to explicitly upon ordering a taxi within the same conversation. Coreference resolution is challenging due to the rich variety of how to form referrals, as well as due to the fact that coreferences often span multiple turns. An example of a coreference that can be handled by our model is found in the example in Figure 1.

The third copy mechanism utilizes the DS as a memory to resolve coreferences. If a slot gate predicts that the user refers to a value that has already been assigned to a different slot during the conversation, then the probability distribution over all possible slots that can be referenced is

$$p_{t,s}^{\text{refer}}(r_t^{\text{CLS}}) = \text{softmax}(W_{\text{refer}}^s \cdot r_t^{\text{CLS}} + b_{\text{refer}}^s) \in \mathbb{R}^{N+1}, \quad (5)$$

i.e., for each slot, a linear layer classification head either predicts the slot which contains the referenced value, or *none* for no reference.

3.6 Auxiliary Features

Some recent approaches to neural DST utilize auxiliary input to preserve contextual information. For instance, SOM-DST adds the dialog state to its single-turn input as a means to preserve context across turns.

We already include contextual information in the input to BERT by appending the dialog history H_t . In addition to that, we also create auxiliary features based on the system inform memory and the DS memory. We generate two binary vectors $a_t^{\text{inform}} \in \{0, 1\}^N$ and $a_t^{\text{ds}} \in \{0, 1\}^N$ that indicate whether (1) a slot has recently been informed (based on the system inform memory), or (2) a slot has already been filled during the course of the dialog (based on the DS memory). These vectors are added to the output of BERT in a late fusion approach, and the slot gate probabilities in Equations 2, 3 and 5 become $p_{t,s}^{\text{gate}}(\hat{r}_t^{\text{CLS}})$, $p_{t,s}^{\text{bgate}}(\hat{r}_t^{\text{CLS}})$ and $p_{t,s}^{\text{refer}}(\hat{r}_t^{\text{CLS}})$, with $\hat{r}_t^{\text{CLS}} = r_t^{\text{CLS}} \oplus a_t^{\text{inform}} \oplus a_t^{\text{ds}}$.

3.7 Partial Masking

We partially mask the dialog history H_t by replacing values with BERT’s generic [UNK] token. The masking is partial in the sense that it is applied only to the past system utterances. For the system utterances, the contained values are known and their masking is straightforward. The idea behind partially masking the history is that the model is compelled to focus on the historical context information rather than the sighting of specific values. This should result in more robust representations r_t^{CLS} and therefore better overall slot gate performance.

3.8 Dialog State Update

We employ the same rule-based update mechanism as Chao and Lane (2019) to track the dialog state across turns. At every turn, we update a slot, if a value has been detected which is not *none*. If a slot-value is predicted as *none*, then the slot will not be updated.

4 Experimental Setup

4.1 Datasets

We train and test our model on four datasets, MultiWOZ 2.1 (Eric et al., 2019), WOZ 2.0 (Wen et al., 2016), sim-M and sim-R (Shah et al., 2018). Among these, MultiWOZ 2.1 is by far the most challenging dataset. It is comprised of over 10000 multi-domain dialogs defined over a fairly large ontology. There are 5 domains (train, restaurant, hotel, taxi, attraction) with 30 domain-slot pairs that appear in all portions of the data.

The other datasets are single-domain and significantly smaller. Evaluations on these mainly serve as sanity check to show that we don’t overfit to a particular problem. Some slots in sim-M and sim-R show a high out-of-vocabulary rate, making them particularly interesting for evaluating value independent DST.

The single domain datasets come with span labels. However, MultiWOZ 2.1 does not. We therefore generate our own span labels by matching the ground truth value labels to their respective utterances.

4.2 Evaluation

We compute the joint goal accuracy (JGA) on all test sets for straightforward comparison with other approaches. The joint goal accuracy defined over a dataset is the ratio of dialog turns in that dataset

Models	MultiWOZ 2.1
DST-reader	36.40%
DST-span	40.39%
SUMBT	42.40%**
TRADE	45.60%
MA-DST	51.04%
DS-DST	51.21%
SOM-DST	52.57%
DS-picklist	53.30%
TripPy	55.29±0.28%

Table 1: DST Results on MultiWOZ 2.1 in JGA (\pm denotes the standard deviation. ** MultiWOZ 2.0 result.

for which all slots have been filled with the correct value according to the ground truth. Note that *none* needs to be predicted if a slot value is not present in a turn. In addition to JGA, we compute the accuracy of the slot gates (joint and per-class) and various other metrics for a more detailed analysis of model design decisions.

We run each test three times with different seeds and report the average numbers for more reliable results. MultiWOZ 2.1 is in parts labeled inconsistently. For a fair evaluation, we consider a value prediction correct, if it matches any of its valid labels (for instance "centre" and "center" for the slot-value *hotel-area=centre*) as being correct. We semi-automatically analyzed value label inconsistencies in the training portion of the dataset in order to identify all label variants for any given value. During testing, these mappings are applied as is.

4.3 Training

We use the pre-trained *BERT-base-uncased* transformer (Vaswani et al., 2017) as context encoder front-end. This model has 12 hidden layers with 768 units and 12 self-attention heads each. The maximum input sequence length is set to 180 tokens after WordPiece tokenization (Wu et al., 2016), except for MultiWOZ 2.1, where we set this parameter to 512. We compute the joint loss as

$$\mathcal{L} = 0.8 \cdot \mathcal{L}_{\text{gate}} + 0.1 \cdot \mathcal{L}_{\text{span}} + 0.1 \cdot \mathcal{L}_{\text{refer}}. \quad (6)$$

The function for all losses is joint cross entropy. As there is no coreferencing in the evaluated single-domain datasets, the refer loss is not computed in those cases and the loss function is

$$\mathcal{L} = 0.8 \cdot \mathcal{L}_{\text{gate}} + 0.2 \cdot \mathcal{L}_{\text{span}} \quad (7)$$

instead. Span predictors are presented only spans from the user utterances U_i to learn from (includ-

Models	WOZ 2.0
NBT	84.2%
BERT-DST	87.7%
GLAD	88.1%
GCE	88.5%
StateNet	88.9%
SUMBT	91.0%
TripPy	92.7±0.2%

Table 2: DST Results on WOZ 2.0.

Models	sim-M	sim-R
SMD-DST	96.8%*	94.4%*
LU-DST	50.4%	87.1%
BERT-DST	80.1%	89.6%
TripPy	83.5±1.2%	90.0±0.2%

Table 3: DST Results on sim-M and sim-R. * should be considered as oracle because the value candidates are ground truth labels.

ing the user utterances in the history portion H_i of the input). During training we set the span prediction loss to zero for all slots that are not labeled as *span*. Likewise, the coreference prediction losses are set to zero if slots are not labeled as *refer*. For optimization we use Adam optimizer (Kingma and Ba, 2014) and backpropagate through the entire network including BERT, which constitutes a fine-tuning of the latter. The initial learning rate is set to $2e^{-5}$. We conduct training with a warmup proportion of 10% and let the LR decay linearly after the warmup phase. Early stopping is employed based on the JGA of the development set. During training we use dropout (Srivastava et al., 2014) on the BERT output with a rate of 30%. We do not use slot value dropout (Xu and Sarikaya, 2014) except for one dataset (sim-M), where performance was greatly affected by this measure (see Section 5.1).

5 Experimental Results

Tables 1, 3 and 2 show the performance of our model in comparison to various baselines. TripPy achieves state-of-the-art performance on all four evaluated datasets, with varying distance to the runner-up. Most notably, we were able to push the performance on MultiWOZ 2.1, the most complex task, by another 2.0% absolute compared to the previous top scoring method, achieving 55.3% JGA. The improvements on the much smaller datasets WOZ 2.0, sim-M and sim-R demonstrate that the model benefits from its design on single-domain

Model	JGA
Span prediction only (entire turn)	42.63%
+ triple copy mechanism	49.23%
+ dialog history	52.58%
+ auxiliary features	54.08%
+ masking	54.29%
TripPy (full sequence width)	55.29%

Table 4: Ablation experiments for our model.

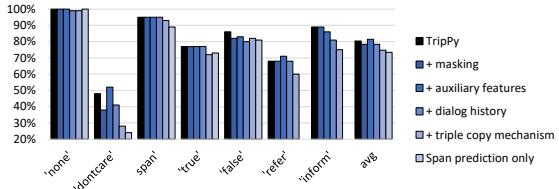


Figure 3: Per class performance of the slot gates for different versions of our model (ablation study).

tasks as well. The following analysis serves a better understanding of our model’s strengths.

5.1 Analysis

We analyse the performance of TripPy on ablation experiments on MultiWOZ 2.1 (see Table 4). Our baseline model is best compared to BERT-DST (Chao and Lane, 2019); we only take single turns as input, and only use span prediction to extract values from the turn. The resulting performance is comparable to other span-based methods such as DST-reader and DST-span and confirms that the dialogs in MultiWOZ are too complex to only be handled by this information extracting mechanism alone.

Impact of the triple copy mechanism Using our proposed triple copy mechanism pushes the performance close to 50%, surpassing TRADE and closing in on the leading hybrid approaches. Especially the performance of the slot gates benefits from this change (see Figure 3). When looking at the F1 score for the individual classes, one can see that the *span* class benefits from the distinction. It is important to point out that none of the coreferences that our model handles can be resolved by span-prediction alone. This means that otherwise guaranteed misses can now be avoided and coreferences can be resolved by copying values between slots. What’s more, using the dialog state memory to resolve coreferences helps value detection across multiple turns, as a value that has been referred to in the current turn might have been assigned to

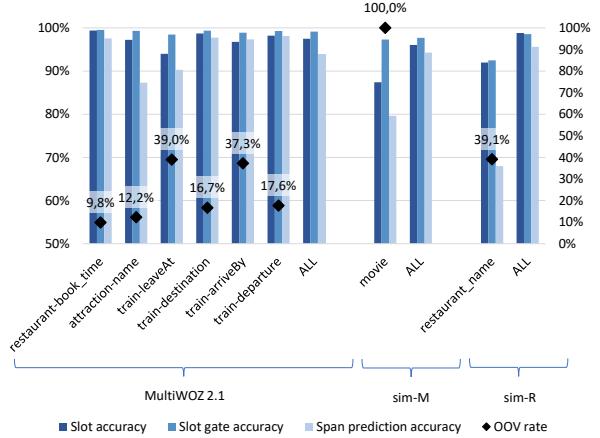


Figure 4: Performance of TripPy on slots with high OOV rate. *ALL* denotes the average of all slots of the respective dataset.

another slot multiple turns before.

Impact of the dialog history We found that using the dialog history as additional context information is critical to a good performance, as it reduces contextual ambiguity. This is clearly reflected in the improved performance of the slot gates (see Figure 3, which has two positive effects. First, the presence and type of values is recognized correctly more often. Especially the special value *dontcare*, and boolean slots (taking values *true* and *false*) benefit from the additional context. This is only logical, since they are predicted by the slot gate using the representation vector of the [CLS] token. Second, values are assigned to the correct slot more often than without the additional contextual information. With the additional dialog history, we outperform DS-DST and match SOM-DST, which set the previous state-of-the-art.

Impact of the auxiliary features SOM-DST uses single turns as input, but preserves additional contextual information throughout the dialog by using the dialog state as auxiliary input. By adding our memory based auxiliary features in a late fusion approach, we surpass SOM-DST, and ultimately DS-picklist, which performs slot-filling with the knowledge of the full ontology. Even though our features carry less information, that is, only the identities of the informed slots – tracked by the system inform memory – and the identities of the previously seen slots – tracked by the DS memory –, we see substantial improvement using them. Obviously, more information about the progress of the dialog helps the slot gates and the referral gates in

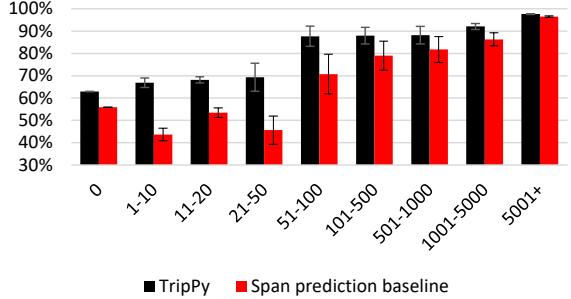


Figure 5: Recall of values depending on the amount of samples seen during training. 0 seen samples means the value is OOV during test time.

their classification tasks.

Impact of partial masking We found that masking the informed values in past system utterances does not give a clear benefit, but it also does not harm performance of the slot gates. While the *inform* cases are detected more accurately, some other cases suffer from the loss of information in the input. Overall, there is a minor overall improvement observable. We report the numbers for MultiWOZ in Table 4 and Figure 3, but would like to note that we have seen the same trend on all other datasets as well.

Impact of the context width Our best model utilizes the full width of BERT (512 tokens). This is a clear advantage for longer dialogs. Maximal context width is not a decisive factor for the single-domain datasets, since their dialogs tend to be shorter. As expected, we have not seen any change in performance on these. For MultiWOZ, we gain 1% absolute by maximizing the history length to preserve as much of the dialog history as possible, achieving 55.3% JGA.

5.2 Generalization Study

It is important that a DST model generalizes well to previously unseen values. We looked at the performance of our model on slots with exceptionally high out-of-vocabulary rates, of which we identified 8 across the evaluated datasets. Figure 4 plots performance measures for these slots and compares them to the average performance for all slots in the respective datasets. Generally, the slots that expect named entities as values show the lowest accuracy. However, the below-average performance of these slots does not seem to be caused by a particularly high OOV rate. Even at 100%, the *movie* slot of sim-M still performs comparably well. Other slots

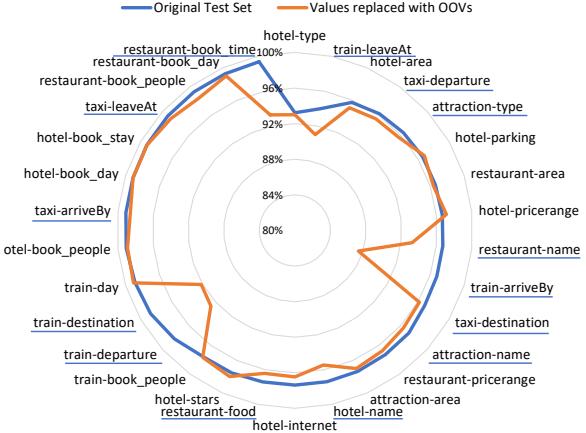


Figure 6: Per-slot accuracy of TripPy on the original test set and the OOV test set. Underlined slot names indicate slots with at least one OOV value.

with relatively high OOV rate still perform close to or better than the average.

Figure 5 plots the recall of values depending on the number of samples seen during training. To our surprise, it does not seem to matter whether a particular value has never been seen during training in order to be detected correctly. OOV values are detected just as well as generally less common values. Our observations however indicate that the model benefits tremendously by seeing a certain minimal amount of training samples for each value, which is somewhere around 50. In other words, if such amounts of data are available, then the model is able to effectively utilize them. In the same Figure we compare TripPy to the span prediction baseline. The latter clearly struggles with OOVs and rare values and generally seems to require more training samples to achieve a good recall. The higher recall on OOV values is likely caused by the fact that many unseen values are of the category time of day, which mostly follows a strict format and is therefore easier to spot. Overall, TripPy clearly generalizes better over sample counts.

To test the limits of our model’s generalization capacities, we manually replaced most of the values in the MultiWOZ test set by (fictional but still meaningful) OOV values. Of the over 1000 unique slot-value pairs appearing in the modified test set, about 84% are OOV after the replacement. Figure 6 compares the per-slot accuracy of our model on the original test set and the OOV test set. Underlined slot names indicate slots with at least one OOV value. Their average OOV rate is 90%. Surprisingly, most of these slots maintain their high

accuracy and only few suffer from the high OOV count. Mainly it is one particular domain, *train*, which suffers above-average performance drops. However, the remainder of the slots maintain their performance. This demonstrates that our model is well equipped to handle OOV values, regardless of the type (e.g., named entity, time of day).

6 Conclusion

We have demonstrated that our approach can handle challenging DST scenarios. Having to detect unseen values does not considerably impair our model’s general performance. The information extraction capabilities of our proposed model are rooted in the memory-based copy mechanisms and perform well even in extreme cases as discussed in Section 5.2. The copy mechanisms are not limited by a predefined vocabulary, since the memories themselves are value agnostic.

To further improve the DST capabilities of TripPy, we hope to introduce slot independence as at present its tracking abilities are limited to slots that are predefined in the ontology. For that, We would like to expand our approach towards the schema-guided paradigm for dialog modeling. We also would like to employ a more sophisticated update strategy, for example by adding the option to partially forget. There already exists an intriguing set of works focusing on these issues and we hope to incorporate and expand upon it in the future.

Acknowledgments

M. Heck, C. van Niekerk and N. Lubis are supported by funding provided by the Alexander von Humboldt Foundation in the framework of the Sofja Kovalevskaja Award endowed by the Federal Ministry of Education and Research, while C. Geishauser, H-C. Lin and M. Moresi are supported by funds from the European Research Council (ERC) provided under the Horizon 2020 research and innovation programme (Grant agreement No. STG2018_804636).

References

- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.
- Guan-Lin Chao and Ian Lane. 2019. BERT-DST: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. *arXiv preprint arXiv:1907.03040*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tür. 2019. MultiWOZ 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.
- Shuyang Gao, Abhishek Sethi, Sanchit Aggarwal, Tagyoung Chung, and Dilek Hakkani-Tür. 2019. Dialog state tracking: A neural reading comprehension approach. *arXiv preprint arXiv:1908.01946*.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.
- Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2019. Efficient dialogue state tracking by selectively overwriting memory. *arXiv preprint arXiv:1911.03906*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Adarsh Kumar, Peter Ku, Anuj Kumar Goyal, Angeliki Metallinou, and Dilek Hakkani-Tür. 2020. MADST: Multi-attention based scalable dialog state tracking. *arXiv preprint arXiv:2002.08898*.
- Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. *arXiv preprint arXiv:1907.07421*.
- Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. *arXiv preprint arXiv:1708.05956*.
- Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2016. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*.
- Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking model. *arXiv preprint arXiv:1812.00899*.
- Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. *arXiv preprint arXiv:1807.06517*.

- Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568. IEEE.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.
- Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. 2018. Towards universal dialogue state tracking. *arXiv preprint arXiv:1810.09587*.
- Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. *arXiv preprint arXiv:1805.01555*.
- Puyang Xu and Ruhi Sarikaya. 2014. Targeted feature dropout for robust slot filling in natural language understanding. In *Fifteenth Annual Conference of the International Speech Communication Association*.
- Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.
- Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.
- Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive dialogue state tracker. *arXiv preprint arXiv:1805.09655*.

Conversational Agents for Intelligent Buildings

Weronika Sieńska, Nancie Gunson, Christopher Walsh,
Christian Dondrup, and Oliver Lemon

School of Mathematical and Computer Sciences,

Heriot-Watt University,

Edinburgh, EH14 4AS, UK

{w.sieinska, n.gunson, c.dondrup, o.lemon}@hw.ac.uk
c.walsh.1@research.gla.ac.uk

Abstract

We will demonstrate a deployed conversational AI system that acts as a host of a smart-building on a university campus. The system combines open-domain social conversation with task-based conversation regarding navigation in the building, live resource updates (e.g. available computers) and events in the building. We are able to demonstrate the system on several platforms: Google Home devices, Android phones, and a Furhat robot.

1 Introduction

The combination of social chat and task-oriented dialogue has been gaining more and more popularity as a research topic (Papaioannou et al., 2017c; Pecune et al., 2018; Khashe et al., 2019). In this paper, we describe a social bot called Alana¹ and how it has been modified to provide task-based assistance in an intelligent building (called the GRID) at the Heriot-Watt University campus in Edinburgh. Alana was first developed for the Amazon Alexa Challenge in 2017 (Papaioannou et al., 2017b,a) by the Heriot-Watt University team and then improved for the same competition in 2018 (Curry et al., 2018). The team reached the finals in both years. Now Alana successfully serves as a system core for other conversational AI projects (Foster et al., 2019).

In the GRID project, several new functionalities have been added to the original Alana system which include providing the users with information about:

- the GRID building itself (e.g. facilities, rooms, construction date, opening times),
- location of rooms and directions to them,
- events happening in the building,
- computers available for use – updated live.

¹See <http://www.alanaai.com>

Currently, our intelligent assistant is available for users on several Google Home Mini devices distributed in the GRID – a large university building with multiple types of users ranging from students to staff, and visitors from business/industry. It is also available on Android phones via Google Actions as part of the Google Assistant. The system is reconfigurable for other buildings, via a graph representation of locations and their connectivity. It connects to live information about available resources such as computers and to an event calendar.

2 System Architecture

Figure 1 presents the architecture of the system.

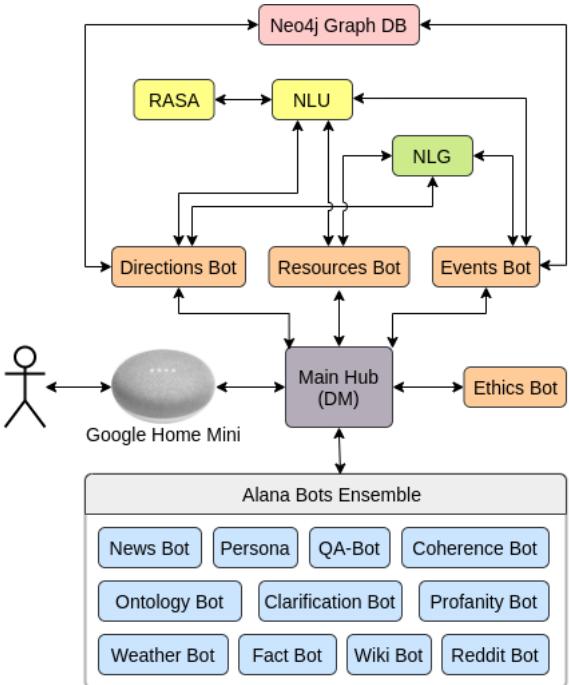


Figure 1: System architecture.

The Alana system is an ensemble of several different conversational bots that can all potentially

produce a reply to the user’s utterance. Each bot uses different information resources to produce its reply. Example resources are: Wikipedia, Reddit, many different News feeds, a database of interesting facts etc. Additionally, there are also conversational bots that drive the dialogue in case it has stalled, deal with profanities, handle clarifications, or express the views, likes, and dislikes of a virtual Persona. The decision regarding which bot’s reply is selected to be verbalised is handled by the Dialogue Manager (DM).

ASR/TTS In the GRID project, the audio stream is handled using the Google Speech API and the system is therefore also available as an action on Google Assistant on Android phones.

NLU In the Alana system, users’ utterances are parsed using a complex NLU pipeline, described in detail in (Curry et al., 2018), consisting of steps such as Named Entity Recognition, Noun Phrase extraction, co-reference and ellipsis resolution, and a combination of regex-based and deep-learning-based intent recognition. In the GRID project, an additional NLU module has been implemented for building-specific enquiries which uses the RASA² framework. In the Persona bot we use AIML patterns for rapid reconfigurability and control.

NLG The NLG strategy depends on the different conversational bots. It ranges from the use of complex and carefully designed templates to automatically summarised news and Wikipedia articles (Curry et al., 2018).

DM In every dialogue turn each of the bots attempts to produce a response. Which response will be uttered to the user is determined by a selection strategy which is defined by a bot priority list and can also be learned from data (Shalymnov et al., 2018). If several bots produce a response in a given turn, the response of the bot with the highest priority is selected. The Directions, Events, and Resources bots are placed at the top of the priority list in order to increase the chance of their responses being selected because all of them provide information regarding the GRID building. The Coherence Bot has the lowest priority since its role is to handle turns when none of the other bots produce a response so that the conversation can be driven forward.

²See <https://rasa.com/>

The Neo4j Graph Database Figure 2 presents the graph representation of how the rooms are connected with each other in the GRID building. To port the system to different buildings, this representation would be changed to reflect the locations and their spatial connectivity.

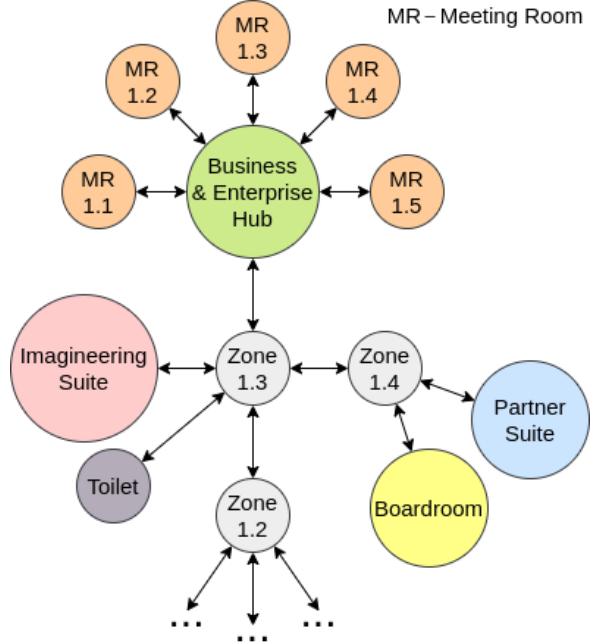


Figure 2: Graph representation of spatial connectivity – example for the GRID building.

3 Example Dialogues

General information about the GRID building is encoded in Persona’s AIML templates. It is this bot that responds to questions whose answers are generally fixed over time – for example about opening hours and where one can buy a cup of coffee. For examples of responses generated by Persona, see Dialogue 1.

The Ethics Bot has been implemented to obtain users’ consent for storing and processing transcriptions of their conversations. The request for user consent is omitted in the example dialogues.

The Directions Bot handles requests for locations of rooms and directions to them. It connects to the Neo4j graph database where it retrieves the shortest paths. The Events Bot responds whenever events information is requested for either a specific room or the building in general. The information is stored in the Neo4j database as well where the events are assigned to particular rooms. It is retrieved from a university timetable system and updated every day.

A: Welcome to the GRID building! How can I help you today?

U: What does GRID stand for?

A: (persona) GRID stands for Global, Research, Innovation and Discovery. Is there anything else I can help you with?

U: What time does this building close?

A: (persona) This building never closes. It is open 24/7. Did you know that you can have a chat with me about music? Also, if you're looking for a specific room, you can ask me how to get there.

U: Actually, I really need a cup of coffee. Do you know where I can get one?

A: (persona) You can have a cup of coffee at the cafe on the ground floor. It looks like a tuk tuk. Can I help you with anything else?

U: What is a tuk tuk?

A: (wiki_bot) I think that an auto rickshaw is a motorized version of the pulled rickshaw or cycle rickshaw. Most have three wheels and do not tilt. They are known by many terms in various countries including auto, baby taxi, bajaj, chand gari, lapa, tuk-tuk, 3wheel or tukxi.

U: Wow, that's interesting. Thanks, goodbye!

A: It was lovely talking to you! Goodbye!

Dialogue 1: Questions about the GRID building. U – User, and A – Alana system.

The role of the Resources Bot is to provide the user with information about how many computers are currently free in the GRID building's labs. The bot scrapes the data live from a university website. Dialogue 2 presents example responses produced by the Directions, Events, and Resources Bots.

Dialogue 3 shows an example of social chat one can have with our system. Alana offers conversation about various topics such as music, movies, and sports. It provides current news as well.

4 Future Work

4.1 Deployment on a Robot

We can already deploy Alana on a Furhat³ robot during special events in the GRID building. We believe that having Alana deployed on a robot makes it even more attractive and encourages more people to interact with it.

This will also be useful for a future project which will aim to further extend Alana's functionalities so that for example, it can become a virtual receptionist in the National Robotarium (a smart research building for Human-Robot Interaction) which is soon to be built at the Heriot-Watt University campus in Edinburgh.

³See <https://www.furhatrobotics.com/>

A: Hello, and welcome to the GRID building! What can I assist you with?

U: Hi! What can you do?

A: (persona) If you're looking for a room, I can tell you where it is. I can also tell you what events are happening in this building and where you can find a free computer.

U: Are there any available computers in the Flex Lab?

A: (resources_bot) Sorry, there is Flex Lab 1 and Flex Lab 2 in this building. Which one do you mean?

U: Number 2.

A: (resources_bot) I'm afraid there are no free computers in the Flex Lab 2 at the moment, but I know that the room Digital Lab has the most free computers, it has 97 PCs available.

U: Where is the Digital Lab?

A: (directions_bot) The Digital Lab is on the first floor. Would you like directions?

U: Yes, please.

A: (directions_bot) From the entrance, head across the atrium to the right and take the accessible lift to the first floor, then turn right, the door to the Digital Lab will be in front of you. Don't forget that you can ask me about events happening in this building.

U: Ok, is there anything on today?

A: (events_bot) There are 2 events taking place in the Boardroom today. An event called Negotiate Training is scheduled from 10:30 until 12:30. Another event called Virtual Reality Workshop is taking place from 13:15 until 19:15. Is there anything else I can help you with?

U: No, I'm good. Thanks, goodbye!

A: It was lovely talking to you! Goodbye!

Dialogue 2: Task-based dialogue. U – User, and A – Alana system.

A: Welcome to the GRID building! How can I help you today?

U: Let's talk about music.

A: (coherence_bot) I'm into music. I'm really into rap and rock. What about you?

U: I like Bob Dylan a lot.

A: (news_bot) A friend of mine told me about this news headline from MTV News about Bob Dylan. Timothée Chalamet will play Bob Dylan in his next Oscar worthy role. Do you want to hear some more about this?

U: Who is Timothée Chalamet?

A: (wiki_bot) As far as I know, Timothée Hal Chalamet is an American actor.

U: What movies do you like?

A: (persona) My favourite movie is Star Wars. I love the two androids in that film. Which movie do you like?

U: The new Star Trek movies are pretty good.

A: (news_bot) Related to Star Trek, I heard this news headline from a friend, 'Star Trek: Picard': Everything you need to watch before the new CBS series. Do you want to learn more?

U: No, thanks. I've gotta go. Bye!

A: It was lovely talking to you! Goodbye!

Dialogue 3: Open-domain chat about music, movies, and news. U – User, and A – Alana system.

4.2 Graphical App

Furthermore, we collaborate with Speech Graphics⁴ which is a company that provides facial animation for speaking game characters from audio input. Together we aim to create a mobile app with Alana and its animated avatar which will be available to students and academics at Heriot-Watt University in Edinburgh. Figure 3 presents two of the avatars available for Alana. We believe that the interaction with the graphical app will be more appealing for users than talking to the Google Home devices.



Figure 3: Example Speech Graphics avatars.

4.3 Evaluation

We are conducting experiments where we compare two versions of the developed system. One of them is the full version of the Alana-GRID system implemented in this project and the other is Alana deprived of its open-domain conversational skills i.e. only capable of providing information about the GRID building which the user requests. Our hypothesis is that open-domain social chat adds value to virtual assistants and makes it more pleasurable and engaging to talk to them.

Acknowledgments

This work has been partially funded by an EPSRC Impact Acceleration Award (IAA) and by the EU H2020 program under grant agreement no. 871245 (SPRING)⁵.

References

Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov,

Xinnuo Xu, Ondřej Dušek, Arash Eshghi, Ioannis Konstas, Verena Rieser, et al. 2018. Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking. *Alexa Prize Proceedings*.

Mary Ellen Foster, Bart Craenen, Amol Deshmukh, Oliver Lemon, Emanuele Bastianelli, Christian Dondrup, Ioannis Papaioannou, Andrea Vanzo, Jean-Marc Odobez, Olivier Canévet, Yuanzhouhan Cao, Weipeng He, Angel Martínez-González, Petr Motlicek, Rémy Siegfried, Rachid Alami, Kathleen Belhassen, Guilhem Buisan, Aurélie Clodic, Amandine Mayima, Yoan Sallami, Guillaume Sarthou, Phani-Teja Singamaneni, Jules Waldhart, Alexandre Mazel, Maxime Caniot, Marketta Niemelä, Päivi Heikkilä, Hanna Lammi, and Antti Tammela. 2019. MuMMER: Socially Intelligent Human-Robot Interaction in public spaces.

Saba Khashe, Gale Lucas, Burcin Becerik-Gerber, and Jonathan Gratch. 2019. Establishing social dialog between buildings and their users. *International Journal of Human-Computer Interaction*, 35(17):1545–1556.

Ioannis Papaioannou, Amanda Cercas Curry, Jose Part, Igor Shalyminov, Xu Xinnuo, Yanchao Yu, Ondřej Dušek, Verena Rieser, and Oliver Lemon. 2017a. An ensemble model with ranking for social dialogue. NIPS 2017 Conversational AI Workshop; Conference date: 08-12-2017 Through 08-12-2017.

Ioannis Papaioannou, Amanda Cercas Curry, Jose L. Part, Igor Shalyminov, Xinnuo Xu, Yanchao Yu, Ondřej Dušek, Verena Rieser, et al. 2017b. Alana: Social dialogue using an ensemble model and a ranker trained on user feedback. *Alexa Prize Proceedings*.

Ioannis Papaioannou, Christian Dondrup, Jekaterina Novikova, and Oliver Lemon. 2017c. Hybrid chat and task dialogue for more engaging HRI using reinforcement learning. In *26th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2017, Lisbon, Portugal, August 28 - Sept. 1, 2017*, pages 593–598. IEEE.

Florian Pecune, Jingya Chen, Yoichi Matsuyama, and Justine Cassell. 2018. Field trial analysis of socially aware robot assistant. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS ’18, page 1241–1249, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Igor Shalyminov, Ondřej Dušek, and Oliver Lemon. 2018. Neural response ranking for social conversation: A data-efficient approach. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, Brussels, Belgium. Association for Computational Linguistics.

⁴See <https://www.speech-graphics.com/>

⁵See <http://spring-h2020.eu/>

RETICO: An incremental framework for spoken dialogue systems

Thilo Michael
Quality and Usability Lab
Technische Universität Berlin
`thilo.michael@tu-berlin.de`

Abstract

In this paper, we present the newest version of retico - a python-based incremental dialogue framework to create state-of-the-art spoken dialogue systems and simulations. Retico provides a range of incremental modules that are based on services like Google ASR, Google TTS, and Rasa NLU. Incremental networks can be created either in code or with a graphical user interface. In this demo, we present three example systems that are implemented in retico: a spoken translation tool that translates speech in real-time, a conversation simulation that models turn-taking, and a spoken dialogue restaurant information service.

1 Introduction

Classical architectures of spoken dialogue systems rely on a pipeline approach, where data is passed through and transformed by a set of modules. These modules perform a specific task on the data, for example, convert speech signals into text (ASR modules) or extracting domain-specific information from text (NLU modules). While this architecture separates the concern between the modules and modularizes the development of spoken dialogue systems, the resulting agents are slow to process data and cannot react quickly to changes in the input.

Incremental processing, an architecture where modules work on small increments of data and forward hypotheses based on those increments to the next module, increases the processing speed and reactivity of dialogue systems while still retaining the modularized approach of the pipeline architecture (Schlangen and Skantze, 2011). However, due to the overhead of creating and revoking hypotheses and processing on incomplete data, the complexity of each module in an incremental dialogue system increases. For researchers, it can

be a challenge to implement and evaluate incremental modules, as they do not have the time and knowledge to implement a complete incremental dialogue system, just to evaluate the part they are researching.

The incremental processing toolkit (InproTK) is an open-source toolkit written in Java that provides an interface for incremental modules and defines an architecture for incremental units, hypothesis handling, and connections between incremental modules (Baumann and Schlangen, 2012). However, the toolkit does not provide an integrated framework that allows for the design and evaluation of networks.

In this paper, we present the current version of retico, an incremental framework for spoken dialogue that was first published in (Michael and Möller, 2019). Retico is a framework written in python and published as an open-source project¹. We demonstrate three types of speech and dialogue systems that are implemented in this framework. First, we showcase an incremental translation service that utilizes Google Translate to recognize, translate, and synthesize speech. Also, we showcase a simulation of a conversation with turn-taking, where two agents interact with each other. Finally, we showcase a spoken dialogue system in the restaurant information domain. All demo systems are visualized in a graphical user interface, and the networks can be adjusted live (e.g., speech synthesis modules can be switched).

2 Related Work

The incremental model has been formalized by Schlangen and Skantze in (Schlangen and Skantze, 2009, 2011). The resulting framework InproTK (Baumann and Schlangen, 2012) has been used for incremental speech recognition and syn-

¹Available at www.github.com/uhlo/retico

thesis and dialogue systems, among others. Based on this, InproTK_S extends the toolkit for the use of situated dialogue(Kennington et al., 2014).

Recent work in modules of spoken dialogue systems like speech recognition (Selfridge et al., 2011) and end-of-turn prediction (Skantze, 2017) focused on incremental processing, and a state-of-the-art natural language understanding module has been incrementalized (Rafla and Kennington, 2019).

Incremental processing cannot only be used in spoken dialogue system, but it also can be useful for research regarding conversation simulation (Michael and Möller, 2020).

3 Architecture

The architecture of retico is written in python based on the conceptual model of incremental processing described in (Schlangen and Skantze, 2009). Core of this framework are the abstract definitions of an *incremental module* (IM) and an *incremental unit* (IU). Both definitions provide interfaces and processing routines to handle concurrent processing of modules and the flow of IUs between the modules. Each IM has a *left buffer*, where IUs of other modules are placed to be processed and a *right buffer* where new hypotheses are placed and sent to IMs further down the incremental pipeline. Usually, an IM defines one or more types of IU that it is able to process and one type of IU it produces and produces or revokes hypotheses based on every incoming unit.

Besides these modules, retico provides interfaces for information exchange apart from IUs by changing meta-information of IMs and by calling “*trigger*” modules, that produce IUs on-demand and insert them to the buffers of modules.

3.1 Incremental Units

Incremental units are mainly defined by their payload, which differs widely depending on the type of data the IU is carrying. For example, an `AudioIU` stores chunks of audio data that is captured by a microphone, while a `TextIU` stores text recognized by an ASR module or generated by an NLG module.

IUs also manage references to IUs they are based on, as well as IUs that precede it. This information is automatically collected and added to the IU when it is created as part of the processing routine of an Incremental Module.

Additionally, IUs retain information on their

hypothesis-status, that is, if they are *committed* (no further changes to the hypothesis will be made) or if they are *revoked* (the hypothesis is no longer valid and may be replaced with a newer hypothesis). Also, meta-data in the form of key-value-pairs can be attached to an IU. In contrast to the payload of an IU, the meta-data is not standardized for a type of an IU and is not guaranteed to be present. However, it is a useful tool for debugging or storing information used for visualization.

3.2 Incremental Modules

Incremental modules represent the core functionality of retico. Their connectivity is defined by one or more input IU types and one output IU type. However, there are special *producer* modules that do not accept any input IUs because they obtain information from other sources (e.g., the `MicrophoneModule`) and *consumer* modules that do not output any IUs (e.g., the `SpeakerModule`). The primary processing method of an incremental module is invoked every time there is a new IU in the left buffer, and it may return a new IU for the right buffer. Like IUs, incremental modules are also able to hold meta-data, which is used for debugging and visualization purposes.

Retico already includes modules from various fields of a spoken dialogue system. Most notably, there exists modules that handle Audio input and output, online and offline speech recognition (CMUSphinx, Google ASR), natural language understanding (rasa NLU), dialogue management (agenda-based, rasa RNN-based, n-gram-based), speech synthesis (Mary TTS, Google TTS) as well as translation services (Google Translate). Additional modules and integrations from other frameworks are in planning.

3.3 Logging and Persistence

The IUs that are defined in retico generalize via the python inheritance structure, so that standard data types like audio, text, and dialogue acts are supported. This allows retico to persist IUs of these types with so-called “recorder” modules.

Retico modules are serializable so that networks can be stored into a file to be loaded and initialized again later.

3.4 Graphical User Interface

While modules can be created, connected, and run purely in python code, it also provides a GUI that

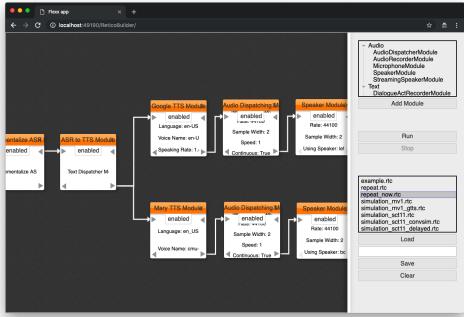


Figure 1: Screenshot of the graphical user interface to create, save and load incremental networks.

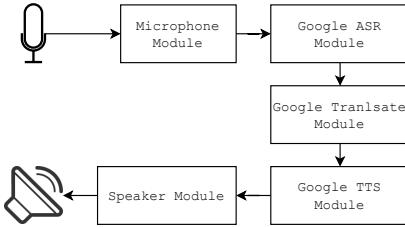


Figure 2: The schema of the translation service.

visualized networks and the flow of IUs. Figure 1 shows the user interface that runs in a browser. Available modules are shown in a tree list and can be added onto the canvas. Connections between modules can be made by clicking on the sockets, and the interface highlights only modules that can process the specific IU type. When a network is run in the GUI, the modules show basic information about the IU they are currently processing.

Networks created with the GUI can be saved to a file and be loaded again with the GUI or with python code. The position and size information of the modules are stored in the module’s metadata, which allows retico to retain the layout of the network when loading it from file.

4 Demonstrations

In this section, we present three different projects that are created entirely in retico. Due to the modular approach of retico, these systems are not fixed regarding the modules they use for a given task. For example, retico is able to use two different speech synthesis modules that can be interchanged.

4.1 Spoken Translation Service

The translation service utilizes speech recognition, a text translation service, and speech synthesis to translate sentences spoken into the system. As can be seen in Figure 2, the main components used in this setup are the Google ASR, Google TTS, and the Google Translate modules. While the ASR

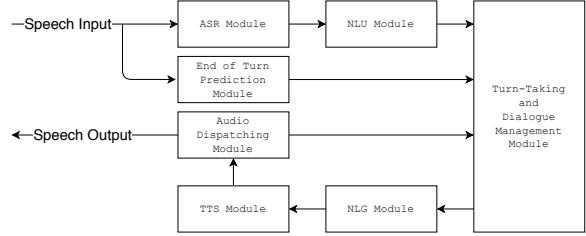


Figure 3: The schema of one agent in the turn-taking conversation simulation network. The complete simulation includes two agents whose speech-input and -output are connected, as well as a recording modules that stores the conversation onto disk for further evaluations.

module works on word and sub-word level, the translation module collects multiple words so that a potential translation stabilizes. The translated sentences are synthesized with Google TTS and transmitted to the speakers.

The languages that can be translated by this service are only limited by the capabilities of Google ASR, TTS, and translation services. However, we tested the system with German-English, English-French, and German-French translations.

Because there is no echo suppression implemented in this version of the service, the loudspeaker, and the microphone have to be acoustically separated (e.g., via a headset).

4.2 Conversation Simulation

The conversation simulation consists of two spoken dialogue systems that are connected and can communicate through an audio channel. Because of the incremental implementation, the agents can predict the end-of-turn of their interlocutors and perform rudimentary turn-taking. As can be seen in Figure 3, an Audio Dispatching Module is used to control when an agent speaks and when it is silent, and it also provides feedback of the status of the current utterance back to the dialogue manager. The simulated conversation itself models a short conversation test as standardized by ([ITU-T Recommendation P.805, 2007](#)). Concretely, the scenario describes a telephone conversation between a worker at a pizzeria and a customer. The customer inquires about available dishes and their toppings, selects an item from the menu, and the pizzeria worker requests information like telephone number and address.

The modules in this network (ASR, TTS, NLG, NLU, end-of-turn) are based on recorded data from real conversations performed in laboratory conditions that were transcribed and annotated

with dialogue acts and turn markers. The incremental modules in the simulation make use of meta-information transmitted through retico’s side-channel to perform their tasks. The utterances produced by the agents are sliced from the empirical conversations. However, other synthesis methods can be used.

4.3 Restaurant Information System

The restaurant information system is a spoken dialogue system that finds restaurants based on user-given criteria like location of the restaurant, as well as the type and price of food. Once every slot is filled, the dialogue system queries a database and recommends restaurants that match the criteria. Depending on the complexity of the query, the request to the database can be slow. The incremental processing, together with a caching-mechanism implemented into the database connector, allows for faster response times of the dialogue system.

The speech recognition and synthesis are realized with Google ASR, and TTS modules, and rasa NLU is used for the natural language understanding. The dialogue manager used in this system is rule-based and uses slot-filling to query restaurants.

5 Conclusion

In this paper, we presented the newest version of retico, a framework for incremental dialogue processing. We described the incremental architecture and highlighted the logging and persistence features as well as the graphical user interface. We also showcased three application ideas created with the framework, that span a wide range of possible speech dialogue systems. We described a service that translates speech in increments, a conversation simulation that is able to perform turn-taking, and a dialogue system that processes increments to decrease the time used to query a database.

While we focus on applications in the area of spoken dialogue, the incremental approach of this framework can be applied to other areas of research as well. For example, modules for video input and object detection can be used to reference positions of objects in the dialogue, and robotics features may be integrated so that a dialogue system can interact with its environment.

The framework is published as open source and available online at

<https://www.github.com/uhlo/retico>.

Acknowledgements

This work was financially supported by the German Research Foundation DFG (grant number MO 1038/23-1).

References

- Timo Baumann and David Schlangen. 2012. The inprotk 2012 release. In *NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*, pages 29–32. Association for Computational Linguistics.
- ITU-T Recommendation P.805. 2007. *Subjective Evaluation of Conversational Quality*. International Telecommunication Union, Geneva.
- Casey Kennington, Spyros Kousidis, and David Schlangen. 2014. Inprotks: A toolkit for incremental situated processing. *Proceedings of SIGdial 2014: Short Papers*.
- Thilo Michael and Sebastian Möller. 2019. Retico: An open-source framework for modeling real-time conversations in spoken dialogue systems. In *30th Konferenz Elektronische Sprachsignalverarbeitung (ESSV)*, pages 238–245, Dresden. TUDpress.
- Thilo Michael and Sebastian Möller. 2020. Simulating turn-taking in conversations with varying interactivity. In *31th Konferenz Elektronische Sprachsignalverarbeitung (ESSV)*, pages 208–215, Dresden. TUDpress.
- Andrew Rafla and Casey Kennington. 2019. Incrementalizing rasa’s open-source natural language understanding pipeline. *arXiv preprint arXiv:1907.05403*.
- David Schlangen and Gabriel Skantze. 2009. A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 710–718. Association for Computational Linguistics.
- David Schlangen and Gabriel Skantze. 2011. A general, abstract model of incremental dialogue processing. *Dialogue and Discourse*, 2(1):83–111.
- Ethan O Selfridge, Iker Arizmendi, Peter A Heeman, and Jason D Williams. 2011. Stability and accuracy in incremental speech recognition. In *Proceedings of the SIGDIAL 2011 Conference*, pages 110–119. Association for Computational Linguistics.
- Gabriel Skantze. 2017. Towards a general, continuous model of turn-taking in spoken dialogue using lstm recurrent neural networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 220–230.

MC-Saar-Instruct: a Platform for Minecraft Instruction Giving Agents

Arne Köhn and Julia Wichlacz and Christine Schäfer

Álvaro Torralba and Jörg Hoffmann and Alexander Koller

koehn@coli.uni-saarland.de, wichiacz@cs.uni-saarland.de, cschaef@coli.uni-saarland.de,
torralba@cs.uni-saarland.de, hoffmann@cs.uni-saarland.de, koller@coli.uni-saarland.de

Saarland University

Abstract

We present a comprehensive platform to run human-computer experiments where an agent instructs a human in Minecraft, a 3D blocksworld environment. This platform enables comparisons between different agents by matching users to agents. It performs extensive logging and takes care of all boilerplate, allowing to easily incorporate new agents to evaluate them. Our environment is prepared to evaluate any kind of instruction giving system, recording the interaction and all actions of the user. We provide example architects, a Wizard-of-Oz architect and set-up scripts to automatically download, build and start the platform.

1 Introduction

Collaborative human-computer interaction can occur in different environments. While interaction in the physical world is often a desirable goal, it places a huge burden on automatic agents as perception is a hard problem, raising the barrier of setting up such experiments significantly. On the other end, interactions on a custom-built platform may be a good fit to explore specific phenomena, but they do not scale easily to different or complex problems. A good example for a custom-built virtual 3D world is the GIVE challenge, where an instruction system must guide a player to press a specific sequence of buttons in a 3D environment while avoiding to step into traps (Byron et al., 2009; Striegnitz et al., 2011). We instead use a *general-purpose* 3D environment.

We release an experimentation platform based on Minecraft (see Figure 1). Minecraft is a game in which the players are situated in a 3D world, which mainly consists of blocks. The game can either be played locally as a single-player game or one can join an online server and play with others. The players can move around, place and remove blocks, and even craft new blocks or items. As such, Minecraft



Figure 1: Example of instructions provided to a user. New instructions appear at the bottom of the chat text and old ones fade out after five seconds. In this case, the world is initialized with a tiny river; the stripes are the barriers for the user.

can be seen as a classic blocksworld that can be scaled up a lot in complexity: Blocks can have different types (wood, earth, stone, glass, lamps, ...), they can be combined into high-level objects, and special blocks even enable building circuits, resulting in Turing-complete machinery. Minecraft contains different game modes: a survival mode, which focuses on exploration and survival in the game world, and the creative mode, focusing on building. We make use of the creative mode.

This feature-richness makes Minecraft a perfect environment for the evaluation of all kinds of intelligent agents (Johnson et al., 2016), from reinforcement learning agents (Guss et al., 2019), to instruction receiving (Sziyal et al., 2019) and instruction giving assistants (Narayan-Chen et al., 2019). Its popularity (Minecraft is the most sold game of all time), together with the client-server architecture make Minecraft a tool well-suited for crowd-sourcing with volunteers from all over the world. Moreover, there are tons of instruction videos for Minecraft on the internet which could be used as auxiliary datasets for offline instruction giving. This addresses several of the limitations

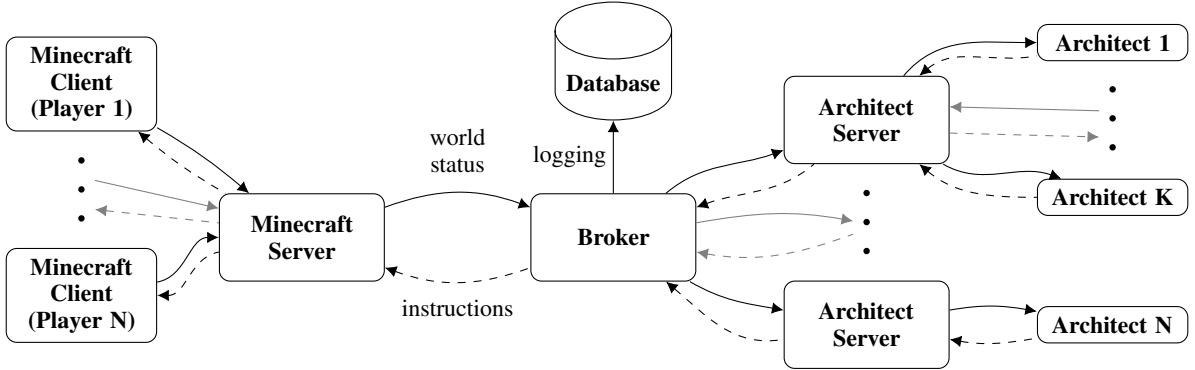


Figure 2: Overview of the services in MC-Saar-Instruct. Updates on the world state are passed along the full lines, instructions are forwarded along the dashed lines.

that previous frameworks like GIVE had: attracting an even larger number of users for the experiments, being more engaging, and allowing for a variety of experiments of increasing complexity.

The platform presented here makes it easy to set up and run instruction giving experiments in Minecraft. In our research, we focus on instructing the user to build complex objects (Wichlacz et al., 2019; Köhn and Koller, 2019), but our platform can easily be used for other generation tasks.

2 System Overview

MC-Saar-Instruct is implemented as a distributed platform which is shown in Figure 2. It consists of the following components, which can each run on their own server:

- The Minecraft server accepts connections from users.
- The Broker decides which scenario will be played by the user, tells the Minecraft server how to initialize the user’s world, pairs the user with an architect and logs all interactions.
- The Architect is the agent with which the user interacts. It receives status updates of the world through the broker and sends natural language instructions back.

While there is only one Minecraft server and one broker, there can be several different kinds of architects, each hosted by its own Architect Server. All interactions between these components are handled using the grpc library,¹ abstracting away the low-level networking and providing a succinct and type-safe remote procedure call (RPC) interface.

We provide example Architect Servers in Java, but they can be written in any language with grpc bindings, such as Python, Go, and many more.

2.1 The Minecraft Server

In contrast to other experimentation systems, such as Johnson et al. (2016) (who modify the Minecraft client) or Szlam et al. (2019) (who use the third-party server Cuberite), we make use of the official Minecraft server, which means that users can use an unmodified up-to-date Minecraft client. Experiments can also make use of all features introduced by new Minecraft releases, if they wish. All functionality in Minecraft, including building Turing-complete apparatuses, can be used.

Upon entering the server, each player is teleported into their own world, which is automatically set up to reflect the start state of the scenario selected by the broker (see Figure 1). All interaction between players is inhibited and all changes made by players are reset once they disconnect. Movement is restricted to a square area and players cannot remove the bottom-most layer of the world and fall into the void. World changes not caused by the player (e.g. weather, time) are disabled. The Minecraft Server runs in creative mode so players have infinite access to building blocks and no decreasing hunger or health bars.

Every 100ms, the server sends the current player position and orientation to the broker. It also sends updates whenever the state of the world changes, i.e. whenever a block is placed or destroyed.

Whenever the architect or the broker sends a message to a user, it is shown as a standard chat message (see Figure 1). Players can also send chat messages to the broker. This can be used for responses in experiment surveys (see Section 5) or for

¹<https://grpc.io/>

an architect that can handle clarification questions.

Because all modifications are implemented in a server plugin, players can connect with an unmodified Minecraft client over the internet.

2.2 The Broker

The broker is the centerpiece of the whole system. It connects to all Architect Servers and provides an RPC interface for the Minecraft server. Whenever a player joins the Minecraft server, the broker gets a message and decides which scenario should be played and what kind of architect the user should be paired with. It then sends a request to the corresponding Architect Server to initialize a new architect and matches that architect to the player. Other than these decisions, the broker is mostly passive. All communication between architect and player is routed over the broker. The broker logs all messages to a database, i. e. block additions and deletions, text messages sent to and from the user and position and camera orientation updates. It also logs the start and end times of experiments and each questionnaire.

The broker provides a web interface to monitor the experiments. It shows the status of the newest experiments and can show a complete list of all database records from a specific game. An in-memory database can be used for development purposes so that no local database needs to be set up and the database is clean on every start.

2.3 The Architect

The architect generates the instructions for the users. Each kind of architect is hosted by an Architect Server. Every time an experiment is supposed to start with this type of architect, the Architect Server instantiates a new architect. The server keeps track of which architect is connected to which game and forwards messages from the broker to the correct architect.

The architect is what a researcher developing and evaluating a new instruction-giving agent needs to implement, using e.g. our high-level Java API. The Architect Server, which manages different architects, can then be reused without changes. Architects could also be implemented in other language with grpc bindings; this would then require reimplementing the Architect Server in the new language.

In our Java API, an architect must implement four functions (see Figure 3): one is called when a block is placed, one when a block is destroyed, one for every update of the position and orientation

```
handleStatusInformation(StatusM);  
handleBlockPlaced(BlockPlacedM);  
handleBlockDestroyed(BlockDestroyedM);  
String getArchitectInformation();
```

Figure 3: Interface to implement for a new architect. The base class provides a method to send text messages.

```
3,2,3,BLUE_WOOL  
1,1,4,WATER  
2,1,4,WATER  
...
```

Figure 4: Excerpt from world file for Figure 1. Each line has the X, Y and Z coordinates plus the block type.

of the player and one to obtain the name of the architect. The architect can then send a string to the user at any time, to be displayed in their Minecraft client. A basic architect can be implemented in 80 lines of Java code.

The architect also determines when the player has reached the objective, as it is the only component keeping track of the state of the game. This design means that all experiment-specific logic is encapsulated in the architect and both broker and Minecraft server can always stay unchanged.

3 Defining and Running Experiments

An experiment is defined by two components: the scenarios that the players are supposed to work on and the architects that should be evaluated.

A scenario consists of a definition of an initial state of the world and architect-specific information instructing the architect of the goal. The initial world state is given by a list of blocks with their location and type (see Figure 4). Each scenario is identified by a unique name. We use a shared dependency for all components that contains the necessary descriptions of the world state when starting a scenario as well as the scenario-specific data for the architects, ensuring that the architects and the Minecraft server use the same initial setup.

4 Wizard-of-Oz Architect

We also ship a Wizard-of-Oz architect (woz) to perform human-human interaction experiments. This architect runs in a second Minecraft server where only one player may log in. That player can neither move nor place or destroy blocks. Once this architect is paired with a player by the broker, the viewpoint of the woz player is synchronized with the player, i. e. the woz player always sees exactly

what the player sees. The woz player may send text messages and these are forwarded as instructions in the same manner as those generated by an automatic agent.

We conducted initial experiments with spoken interaction and noticed that the instruction givers used patterns only possible with spoken interaction such as exactly timing single words to the instruction follower’s actions and self-correction. The text-based Wizard-of-Oz setup on the other hand mirrors the setup with an automatic architect as closely as possible.

5 Post-experiment Questionnaires

After finishing an experiment, the participants fill out a questionnaire using the in-game chat. Once the architect determines that a game is over (hopefully in a successful way), the broker takes over the communication channel and asks the user a series of configurable questions. The questions and answers to this post-experiment questionnaire are logged to the database.

The in-game questionnaire allows to keep all interaction with the experiment platform inside a single medium by removing the need to e. g. open a website. It also ensures that the questionnaires and experiment data are always correctly matched. Finally, the questionnaire mechanism can be used for fraud prevention (Villalba, 2019).

6 Conclusions

We introduced a system for researching situated human-computer dialogue in the Minecraft domain. While primarily focused on instruction giving, it can potentially also be used for two-way text interaction. The framework abstracts away from most of the low-level system, providing a clean and easy to use interface for implementing instruction givers. The system also takes care of matching study participants with different architects and logging of all interactions. We ship several example architects, including a Wizard of Oz architect.

We plan to implement a replay viewer which streams the previously recorded actions by a participant to a Minecraft server. All necessary data is already being stored in the database.

MC-Saar-Instruct as well as scripts to automatically download, build and run specific versions of it for reproducible experiments are available from <https://minecraft-saar.github.io>.

Acknowledgements We thank the reviewers for their comments. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102.

References

- Donna Byron, Alexander Koller, Kristina Striegnitz, Justine Cassell, Robert Dale, Johanna D. Moore, and Jon Oberlander. 2009. [Report on the first NLG challenge on generating instructions in virtual environments \(GIVE\)](#). In *ENLG 2009 - Proceedings of the 12th European Workshop on Natural Language Generation, March 30-31, 2009, Athens, Greece*, pages 165–173. The Association for Computer Linguistics.
- William H. Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. 2019. [MineRL: A large-scale dataset of Minecraft demonstrations](#). *International Joint Conference on Artificial Intelligence (IJCAI)*.
- Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. 2016. [The Malmo platform for artificial intelligence experimentation](#). In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4246–4247.
- Arne Köhn and Alexander Koller. 2019. [Talking about what is not there: Generating indefinite referring expressions in Minecraft](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 1–10, Tokyo, Japan. Association for Computational Linguistics.
- Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. [Collaborative dialogue in Minecraft](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.
- Kristina Striegnitz, Alexandre Denis, Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Mariët Theune. 2011. [Report on the second second challenge on generating instructions in virtual environments \(GIVE-2.5\)](#). In *ENLG 2011 - Proceedings of the 13th European Workshop on Natural Language Generation*, pages 270–279. The Association for Computer Linguistics.
- Arthur Szlam, Jonathan Gray, Kavya Srinet, Yacine Jernite, Armand Joulin, Gabriel Synnaeve, Douwe Kiela, Haonan Yu, Zhuoyuan Chen, Siddharth Goyal, et al. 2019. [Why build an assistant in Minecraft?](#) *arXiv preprint arXiv:1907.09273*.
- Martin Villalba. 2019. [Prediction, detection, and correction of misunderstandings in interactive tasks](#). Ph.D. thesis, Saarland University.
- Julia Wichlacz, Alvaro Torralba, and Jörg Hoffmann. 2019. [Construction-planning models in Minecraft](#). In *Proceedings of the ICAPS Workshop on Hierarchical Planning*, pages 1–5.

ConvoKit: A Toolkit for the Analysis of Conversations

Jonathan P. Chang

Cornell University

jpc362@cornell.edu

Caleb Chiam

Cornell University

cc982@cornell.edu

Liye Fu

Cornell University

liye@cs.cornell.edu

Andrew Z. Wang

Stanford University

anwang@cs.stanford.edu

Justine Zhang

Cornell University

jz727@cornell.edu

Cristian Danescu-Niculescu-Mizil

Cornell University

cristian@cs.cornell.edu

Abstract

This paper describes the design and functionality of ConvoKit, an open-source toolkit for analyzing conversations and the social interactions embedded within. ConvoKit provides an unified framework for representing and manipulating conversational data, as well as a large and diverse collection of conversational datasets. By providing an intuitive interface for exploring and interacting with conversational data, this toolkit lowers the technical barriers for the broad adoption of computational methods for conversational analysis.

1 Introduction

The NLP community has benefited greatly from the public availability of standard toolkits, such as NLTK (Bird et al., 2009), StanfordNLP (Qi et al., 2018), spaCy (Honnibal and Montani, 2020), or scikit-learn (Pedregosa et al., 2011). These toolkits allow researchers to focus on developing new methods rather than on re-implementing existing ones, and encourage reproducibility. Furthermore, by lowering the technical entry level, they facilitate the export of NLP techniques to other fields.

Although much of natural language is produced in the context of conversations, none of the existing public NLP toolkits are specifically targeted at the analysis of conversational data. In this paper, we introduce ConvoKit (<https://convokit.cornell.edu>), a Python package that provides a unified open-source framework for computationally analyzing conversations and the social interactions taking place within, as well as a large collection of conversational data in a compatible format.

In designing a toolkit for analyzing conversations, we start from some basic guiding principles. Firstly, conversations are more than mere ‘bags of utterances’, so we must capture what connects utterances into meaningful interactions. This trans-

lates into native support of reply and tree structure as well as other dependencies across utterances.

Secondly, conversations are inherently social. People often engage in multiple conversations, and how we understand interactions is contingent on what we know about the respective interlocutors. Similarly, the way we understand each speaker is contingent on their entire conversational history. Thus, a conversational analysis toolkit must allow for the integration of speaker information and behaviors across different conversations.

Thirdly, conversations occur in vastly different contexts, from dyadic face-to-face interactions, to discussions and debates in institutional settings, to online group discussions, and to large-scale threaded discussions on social media. This means that the toolkit must offer a level of abstraction that supports different interaction formats.

Finally, since conversational data is key to many social science fields (e.g. political science, sociology, social psychology), the framework should be accessible to a broad audience: not only experienced NLP researchers, but anyone with questions about conversations who may not necessarily have a high degree of NLP expertise.

In this paper, we describe how these principles guided our design of ConvoKit’s framework architecture (Section 2), describe some of the analysis methods (Section 3) and datasets (Section 4) included in ConvoKit, and conclude with some high-level remarks on future developments (Section 5).

2 Framework Architecture

The current state of the software and data ecosystem for conversational research is fragmented: popular conversational datasets are each distributed in different data formats, each using their own task-specific schemas, while similarly, code for reproducing various conversational methods

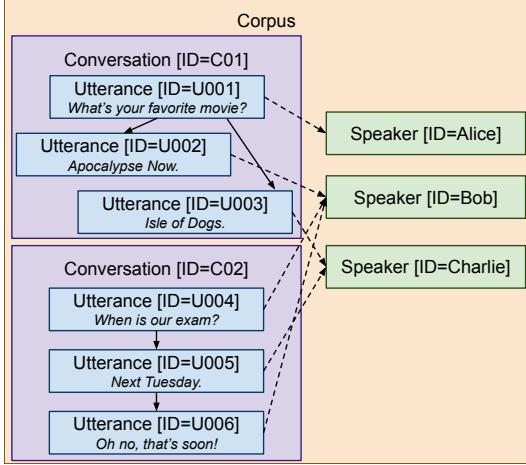


Figure 1: Visualization of the relationship between the four core classes of the Corpus hierarchy. Solid arrows denote reply-to relationships between Utterances, while dashed arrows denote attribution of each Utterance to its authoring Speaker.

tends to be ad-hoc with no guarantee of overlapping functionality or cross-compatibility. This combination of factors poses a barrier to both reproducibility and broader adoption.

To address these issues, a *unified* framework for analyzing conversations must provide both a standardized format for *representing* any conversational data, and a general language for describing *manipulations* of said data. Furthermore, as described in Section 1, the representation must go beyond a mere “bag-of-utterances” and natively capture the structure of conversations, while the language of manipulations must be expressive enough to describe actions at different levels of the conversation: individual utterances, entire conversations, speakers in and across conversations, and arbitrary combinations of the above.

These criteria directly lead to the two core abstractions underlying ConvoKit: the **Corpus**, representing a collection of one or more conversations, and the **Transformer**, representing some action or computation that can be done to a **Corpus**. To draw an analogy to language, **Corpus** objects are the nouns of ConvoKit, while **Transformers** are the verbs.

Representing conversational data. The main data structure for organizing conversational data in ConvoKit is the **Corpus**, which forms the top of a hierarchy of classes representing different levels of a conversation (Figure 1): A **Corpus** is a collection of **Conversations**, each **Conversation** is made up of one or more **Utterances**, and

each **Utterance** is attributed to exactly one **Speaker** (but each **Speaker** can own multiple **Utterances**). **Conversations**, **Utterances** and **Speakers** are identified by unique IDs. Conversation structure is represented by the `reply_to` field of the **Utterance** class, which specifies the ID of the other **Utterance** it replies to (i.e., its parent node in the conversation tree). ConvoKit leverages the relationships between **Utterances**, **Speakers**, and **Conversations** to provide rich navigation of a **Corpus**, such as tree traversal of **Utterances** within a **Conversation** or chronological iteration over all of a **Speaker**’s **Utterance** history.

Custom metadata. Objects in the **Corpus** hierarchy contain some basic information that is generally useful for most operations on conversational data, such as the text content and timestamp of each **Utterance**. However, any use of ConvoKit beyond basic analyses will likely require additional task-specific information. This is supported by ConvoKit in the form of *metadata*. Each of the four classes in the hierarchy contains a field called `meta`, which is a lookup table that may be used to store additional information about the **Corpus**, **Conversation**, **Utterance**, or **Speaker** under some descriptive name. In practice, metadata ranges in complexity from speaker ages to sub-utterance level DAMSL speech act tags.

Manipulating conversational data. ConvoKit supports conversational analyses centered on any level of the hierarchy; for instance, one may wish to examine linguistic characteristics of **Utterances**, characterize a **Conversation** in terms of the structure of its **Utterances**, or track a **Speaker**’s behavior across the **Conversations** they have taken part in throughout their lifetime.

Such flexibility in analysis is achieved by abstracting manipulations of conversational data through the **Transformer** class. At a high level, a **Transformer** is an object that takes in a **Corpus** and returns the same **Corpus** with some modifications applied. In almost all cases, these modifications will take the form of changed or added metadata. For example, the **PolitenessStrategies Transformer** annotates every **Utterance** with a feature vector that counts the presence of politeness features from [Danescu-Niculescu-Mizil et al. \(2013\)](#), while **UserConvoDiversity** annotates every **Speaker** with a measure of their linguistic diversity across the whole **Corpus**.

The key to ConvoKit’s flexibility is that, while a Transformer can represent any arbitrary manipulation of a Corpus and operate at any level of abstraction, all Transformer objects share the same syntax—that is, the Transformer class API represents a general language for specifying actions to be taken on a Corpus. This interface is directly modeled after the scikit-learn class of the same name: a Transformer provides a `fit()` function and a `transform()` function. `fit()` is used to prepare/train the Transformer with any information it needs beforehand; for example, a Transformer that computes bag-of-words representations of Utterances would first need to build a vocabulary. `transform()` then performs the actual modification of the Corpus.

In addition to these standard functions, Transformers also provide a `summarize()` helper function that offers a high-level tabular or graphical representation of what the Transformer has computed. For example, `PolitenessStrategies` offers a `summarize()` implementation that plots the average occurrence of each politeness feature. This can be helpful for getting a quick sense of what the Transformer does, for simple exploratory analyses of a Corpus, or for debugging.

A single Transformer on its own might not make significant changes, but because Transformers return the modified Corpus, multiple Transformers can be chained together, each one taking advantage of the previous one’s output to produce increasingly complex results (see Figure 2 for an example).

3 Transformers

In this section, we introduce some of the built-in Transformers that are available for general use. Broadly speaking, we can group the functionality of Transformers into three categories: preprocessing, feature extraction, and analysis.

Preprocessing refers to the preliminary processing of the Corpus objects prior to some substantive analysis. For example, at the Utterance-level, preprocessing steps can include converting dirty web text into a cleaned ASCII representation (implemented in `TextCleaner`) or running a dependency parse (implemented in `TextParser`). At the Conversation-level, preprocessing steps might include merging consecutive utterances by the same speaker, while at the Speaker-level, they

```

1 corp = Corpus(filename=download(
2   'movie-corpus'))
3
4 # Preprocessing step
5 tc = TextCleaner()
6 tc.transform(corp)
7
8 # Constructing new metadata
9 for c in corp.iter_conversations():
10   genders = [s.meta['gender'] for s
11     in c.iter_speakers()]
12   convo.meta['mixed'] = 'M' in
13   genders and 'F' in genders
14
15 # Analysis step
16 fw = FightingWords()
17 fw.fit(corp,
18   class1_func=lambda utt: utt.
19     get_conversation().meta['mixed'],
20   class2_func=lambda utt: not utt.
21     get_conversation().meta['mixed'])
22 fw.summarize(corp)

```

Figure 2: Basic example code demonstrating how combining different Transformers, and leveraging the Corpus hierarchy’s rich navigation features and metadata functionality, can be used to study conversational data—in this example, comparing the language used in mixed-gender and single-gender movie dialogs.

might include merging contributions from speakers with multiple user accounts.

Feature extraction refers to transformation of conversational data, such as utterance text or conversational structure, into (numerical) features for further analysis and applications. An example of an Utterance-level feature extractor is the previously described `PolitenessStrategies`, while an example of a Conversation-level feature extractor is `HyperConvo`, which constructs a hypergraph representation of the Conversation and extracts features such as (generalized) reciprocity, indegree and outdegree distributions, etc.

Analysis is the process of combining Utterance, Conversation and Speaker features and metadata into a statistical or machine learning model to achieve a higher-level understanding of the Corpus. For example, `FightingWords` implements Monroe et al. (2008)’s method for principled comparison of language used by two subsets of a Corpus; `Classifier` acts as a wrapper around any scikit-learn machine learning model and can be used to classify Utterances, Conversations, or Speakers based on the output of feature extraction Transformers; and `Forecaster`

implements Chang and Danescu-Niculescu-Mizil (2019)’s method for modeling the future trajectory of a Conversation.

Figure 2 illustrates how Transformers belonging to each category can be combined in sequence to perform a practical conversational task: comparing the language used in movie dialogs containing characters of different genders to that used in dialogs containing only one gender.¹

4 Datasets

ConvoKit ships with a diverse collection of datasets already formatted as Corpus objects and ready for use ‘out-of-the-box’. These datasets cover the wide range of settings conversational data can come from, including face-to-face institutional interactions (e.g., supreme court transcripts), collaborative online conversations (e.g., Wikipedia talk pages), threaded social media discussions (e.g., a full dump of Reddit), and even fictional exchanges (e.g., movie dialogs).²

The diversity of these datasets further demonstrates the expressiveness of our choice of conversation representation. We also provide guidelines and code for transforming other datasets into ConvoKit format, allowing ConvoKit’s reach to extend beyond what data is already offered.

5 Conclusions and Future Work

In this paper, we presented ConvoKit, a toolkit that aims to make analysis of conversations accessible to a broad audience. It achieves this by providing intuitive and user friendly abstractions for both representation and manipulation of conversational data, thus promoting reproducibility and adoption.

ConvoKit is actively being developed. While it is currently heavily centered around text analysis (with other modalities being only indirectly supported as metadata), providing first-class support for spoken dialogs is considered as an important line for future extension. In addition, we aim to continue to incorporate new datasets, analysis methods, and integrate with other parts of the NLP software ecosystem that could benefit from ConvoKit’s abstractions, including dialog generation toolkits such as ParlAI (Miller et al., 2018).

¹This example, together with its output and other examples, can be found at <https://convokit.cornell.edu/documentation/examples.html>.

²A complete list of datasets can be found at <https://convokit.cornell.edu/documentation/datasets.html>.

ConvoKit is an open-source project and we welcome contributions of any kind, ranging from bug-fixes and documentation, to augmenting existing corpora with additional useful metadata, to entirely new datasets and analysis methods.³

Acknowledgments

We thank the anonymous reviewers for their thoughtful comments and are grateful to all ConvoKit contributors. This work was supported by an NSF CAREER award IIS-1750615. Zhang was supported in part by a Microsoft PhD Fellowship.

References

- Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O’Reilly Media Inc.
- Jonathan P. Chang and Cristian Danescu-Niculescu-Mizil. 2019. **Trouble on the Horizon: Forecasting the Derailment of Online Conversations as they Develop**. In *Proceedings of EMNLP*.
- Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A Computational Approach to Politeness with Application to Social Factors. In *Proceedings of ACL*.
- Matthew Honnibal and Ines Montani. 2020. spaCy: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing.
- Alexander H. Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, and Jason Weston. 2018. **ParlAI: A Dialog Research Software Platform**. *arXiv:1705.06476 [cs]*.
- Burt L. Monroe, Michael P. Colaresi, and Kevin M. Quinn. 2008. Fightin’ Words: Lexical Feature Selection and Evaluation for Identifying the Content of Political Conflict. *Political Analysis*, 16(04).
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12.
- Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal Dependency Parsing from Scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.

³See contribution guidelines on the ConvoKit webpage.

Commonsense Evidence Generation and Injection in Reading Comprehension

Ye Liu¹, Tao Yang², Zeyu You², Wei Fan² and Philip S. Yu¹

¹Department of Computer Science, University of Illinois at Chicago, IL, USA

²Tencent Hippocrates Research Lab, Palo Alto, CA, USA

{yliu279, psyu}@uic.edu, {tytaoyang, davidwfan}@tencent.com, youz@onid.orst.edu

Abstract

Human tackle reading comprehension not only based on the given context itself but often rely on the commonsense beyond. To empower the machine with commonsense reasoning, in this paper, we propose a Commonsense Evidence Generation and Injection framework in reading comprehension, named **CEGI**. The framework injects two kinds of auxiliary commonsense evidence into comprehensive reading to equip the machine with the ability of rational thinking. Specifically, we build two evidence generators: one aims to generate textual evidence via a language model; the other aims to extract factual evidence (automatically aligned text-triples) from a commonsense knowledge graph after graph completion. Those evidences incorporate contextual commonsense and serve as the additional inputs to the reasoning model. Thereafter, we propose a deep contextual encoder to extract semantic relationships among the paragraph, question, option, and evidence. Finally, we employ a capsule network to extract different linguistic units (word and phrase) from the relations, and dynamically predict the optimal option based on the extracted units. Experiments on the CosmosQA dataset demonstrate that the proposed CEGI model outperforms the current state-of-the-art approaches and achieves the highest accuracy (83.6%) on the leaderboard.

1 Introduction

Contextual commonsense reasoning has long been considered as the core of understanding narratives (Hobbs et al., 1993; Andersen, 1973) in reading comprehension (Charniak and Shimony, 1990). Despite the broad recognition of its importance, the research of reasoning in narrative text is limited due to the difficulty of understanding the causes and effects within the context. Comprehending reasoning requires not only understanding the explicit mean-

P: I was walking home from the store, when I saw an old man laying on the sidewalk, bleeding. The right side of his face was all covered in blood. He was conscious but seemed dazed and probably intoxicated. Nearby there was a young man dialing his cell phone.

Q: What may happen after the young man makes his call?

A: An ambulance would likely come to the scene.

B: The taxi would pick up the young man.

C: None of the above choices.

D: The bus would arrive at the stop soon.

Generated Evidence:

Textual: He will call for medical attention.

Factual: <Blood, AtLocation, emergency room>

<Blood, AtLocation, hospital>, <Ambulance, AtLocation, hospital>

Figure 1: Example of generated evidence helping answer the commonsense question.

ing of each sentence but also making inferences based on implicit connections between sentences.

To answer a contextual commonsense question correctly, two important characteristics need to be well considered. First, the information that is required to infer a correct answer may be beyond the context, and hence adding external commonsense knowledge to guide the reasoning is necessary. Second, how to use external knowledge to gain contextual understanding between the paragraph, question and option set is difficult but important. Despite the great success of large pre-trained models such as BERT (Devlin et al., 2018), GPT (Radford et al., 2018) and RoBERTa (Liu et al., 2019), recent studies suggest that those models fail to capture sufficient knowledge and provide commonsense inference. For example, Poerner et al. (2019) show that language models perform well in reasoning about entity names, but fail to capture rich factual knowledge. Moreover, Talmor et al. (2019) state that language models fail on half of the reasoning tasks which require symbolic operations such as comparison, conjunction and composition.

To this end, we introduce a Commonsense Evidence Generation and Injection framework in reading comprehension, named **CEGI**, which generates useful evidence from textual and factual knowledge and injects the generated evidence into pre-trained models such as RoBERTa. We propose

to generate evidence regarding the facts and their relations. More specifically, we use language models to generate textual evidence and extract factual evidence from a knowledge graph after graph completion. We then inject both evidences into the proposed contextual commonsense reasoning model to predict the optimal answer. As shown in Figure 1, the *Textual Generated Evidence* “He will call for medical attention” and *Factual Generated Evidence* “both blood & ambulance locate at hospital” can help the model find the correct answer “An ambulance would likely come to the scene”.

To capture relations between the paragraph and question, many reading comprehension models (Zhang et al., 2019a; Tang et al., 2019) have been proposed. However, those reasoning models are essentially based on the given context without understanding the facts behind. Moreover, in many situations, the candidate option set contains distractors that are quite similar to the correct answer. In other words, understanding the relations among the option set is also important. We employ a capsule network (Sabour et al., 2017), which uses a routing-by-agreement mechanism to capture the correlations among different options and make the final decision.

Our proposed CEGI framework not only utilizes external commonsense knowledge to generate reasoning evidence but also adopts a capsule network to make the final answer prediction. The explainable evidence and the ablation studies indicate that our method has a large impact on the performance of the commonsense reasoning in reading comprehension. The contributions of this paper are summarized as follows: 1) We introduce two evidence generators which are learned from textual and factual knowledge sources; 2) We provide an injection method that can infuse both evidences into the contextual reasoning model; 3) We adapt a capsule network to our reasoning model to capture interactions among candidate options when making a decision; 4) We show our **CEGI** model outperforms current state-of-the-art models on the CosmosQA dataset and generates richer interpretive evidence which helps the commonsense reasoning.

2 Related Work

2.1 Multi-choice Reading Comprehension

To model the relation and alignment between the pairs of paragraph, question and option set, various approaches seek to use attention and pursue deep

representation for prediction. Tang et al. (2019) and Wang et al. (2018b) model the semantic relationships among paragraph, question and candidate options from multiple aspects of matching. Zhu et al. (2018a) propose a hierarchical attention flow model, which leverages candidate options to capture the interactions among paragraph, question and candidate options. Chen et al. (2019) merge various attentions to fully extract the mutual information among the paragraph, question and options and form the enriched representations.

2.2 Commonsense Knowledge Injection

To empower the model with human commonsense reasoning, various approaches have been proposed on the context-free commonsense reasoning task. The majority of the approaches are focusing on finding the question entity and a reasoning path on the knowledge graph to obtain the answer entity (Huang et al., 2019; Zellers et al., 2018; Talmor et al., 2018). For an instance, Lin et al. (2019) construct graphs to represent relevant commonsense knowledge, and then calculate the plausibility score of the path between the question and answer entity. Lv et al. (2019) extract evidence from both structured knowledge base and unstructured texts to build a relational graph and utilize graph attention to aggregate graph representations to make final predictions. However for contextual commonsense reasoning, it’s hard to find a single most relevant entity from the paragraph or question to obtain the correct answer.

Other approaches focus on enhancing the pre-trained language models through injecting external knowledge into the model and updating the model parameters in multi-task learning (Zhang et al., 2019b; Lauscher et al., 2019; Levine et al., 2019). A knowledge graph injected ERNIE model is introduced in (Zhang et al., 2019b) and a weakly supervised knowledge-pretrained language model (WkLM) is introduced in (Xiong et al., 2019). They both inject the knowledge through aligning the source with the fact triplets in WikiData. However, the parameters need to be retrained when injecting new knowledge, which could lead to the catastrophic forgetting (McCloskey and Cohen, 1989).

3 Task Definition

In multi-choice reading comprehension, we are given a paragraph \mathbf{P} with t tokens $\mathbf{P} = [p_1, p_2, \dots, p_t]$, a question \mathbf{Q} containing n tokens

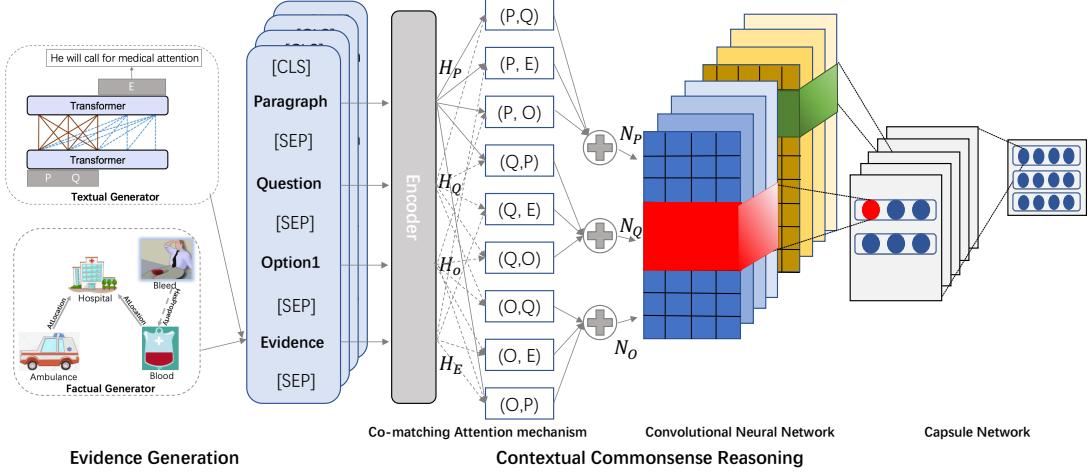


Figure 2: The proposed commonsense evidence generation and injection (**CEGI**) framework.

$\mathbf{Q} = [q_1, q_2, \dots, q_n]$ and the option set with m candidate options $\mathbf{O} = \{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_m\}$, where each candidate option is a text with h tokens $\mathbf{O}_i = [o_1, o_2, \dots, o_h]$. The goal is to select the correct answer \mathbf{A} from the candidate option set. For simplicity, we denote $\mathcal{X} = \{\mathbf{P}, \mathbf{Q}, \mathbf{O}\}$ as one data sample and denote $\mathbf{y} = [y_1, y_2, \dots, y_m]$ as a one-hot label, where each scale $y_i = \mathbf{1}(\mathbf{O}_i = \mathbf{A})$ is an indicator function. In the training stage, we are given N set of $(\mathcal{X}, \mathbf{y})^N$, the goal is to learn a model $f : \mathcal{X} \rightarrow \mathbf{y}$. In the testing, we need to predict \mathbf{y}^{test} given test samples $\mathcal{X}^{\text{test}}$.

When answering a question according to the paragraph, we observe that the context itself often does not provide enough clues to guide us to the correct answer. To this end, we need to know comprehensive information beyond the context and perform commonsense reasoning. Hence, we split the task into two parts: evidence generation and answer prediction, respectively. Our proposed **CEGI** model addresses both parts accordingly by two generators: textual evidence generator and factual evidence generator. In textual evidence generator, our goal is to generate relevant evidence text $\mathbf{E} = [e_1, e_2, \dots, e_k]$ given question \mathbf{Q} and paragraph \mathbf{P} . Note that the number of evidence tokens k may vary in different question and paragraph pair. In factual evidence generator, the goal is to generate relevant text that describes the relations between facts where the facts are the entities from paragraph, question and options. In the second part, we aim to learn a classifier $P(\mathbf{y}|\mathbf{P}, \mathbf{Q}, \mathbf{O}, \mathbf{E})$ that predicts the correct option when a new data sample is given. By using the evidence generated from the first part, we expect the reasoning model can

be enhanced with the auxiliary information, especially for those questions that require contextual commonsense reasoning.

4 Methodology

To tackle reading comprehension task with commonsense reasoning, we introduce a commonsense evidence generation and injection (**CEGI**) framework. The system diagram of the **CEGI** framework is shown in Fig. 2. First, the evidence generation module produces textual evidence and factual evidence. Those generated evidences will be used as auxiliary inputs for the reasoning model. Second, the contextual commonsense reasoning module generates deep contextual features for the paragraph, question, option set and evidence. Meanwhile, a bidirectional attention mechanism is applied to the features to capture representations of the pair of paragraph, question, option set and evidence. Next, all pairs are concatenated and fed into a convolutional neural network for extracting different linguistic units of the options. At least, a capsule network is then applied to dynamically update the representation vector of the candidate options. The final answer is one of the options with the largest vector norm. We describe more details of each component in the following subsections.

4.1 Evidence Generation

It is worthy to mention that many commonsense reasoning types, such as causes of events and effects of events, are important factors of understanding the context in reading comprehension. While those factors are often not explicit or given in the paragraph and option set, answering such may be-

come difficult. To this end, we seek to learn relevant evidence that contains commonsense knowledge. Specifically, we leverage pretrained language models to learn from both context and knowledge graph that may contain reasoning relations. We exploit two kinds of generators, textual evidence generator and factual evidence generator.

4.1.1 Textual Evidence Generator

We observe that daily life events often follow a common routine such that when one event happened, the resulting event or the cause of such an event follows a specific pattern. For an example, in Figure 1, the given paragraph describes a scenario that the old man is hurt and the young man is making a phone call. If we know that he is calling for medical attention, answering the question would become easy. Hence, the goal of our proposed textual evidence generator is to generate the text that follows daily life event routines. We rely on a pretrained language model to acquire the textual evidence by using GPT2 (Radford et al., 2018) and Uniml (Dong et al., 2019). Specifically, in the training, we concatenate the paragraph, question and the correct answer as the input to the standard language model (Liu et al., 2018). Accordingly, the textual evidence generated from the language model is the following sentence after the question text. Note that we stack $[\mathbf{P} \text{ [SEP]} \mathbf{Q} \text{ [SEP]} \mathbf{A}]$ as the input to train the language model. Formally, let $[\mathbf{w}^1, \dots, \mathbf{w}^T] = [\mathbf{P} \text{ [SEP]} \mathbf{Q} \text{ [SEP]} \mathbf{A}]$. The language generation model aims to maximize the following likelihood (Radford et al., 2018):

$$\mathcal{L}_{gen} = \sum_{i=1}^T p(\mathbf{w}^i | \mathbf{w}^1, \dots, \mathbf{w}^{i-1}), \quad (1)$$

where the conditional probability $p(\mathbf{w}^i | \mathbf{w}^1, \dots, \mathbf{w}^{i-1}) = f(\mathbf{w}^1, \dots, \mathbf{w}^{i-1})$ and f is a sequence of operations that (i) converts each token \mathbf{w}^i into token embedding \mathbf{W}_e^i and position embedding \mathbf{W}_p^i ; (ii) transforms them into features with L layers where each layer feature is $\mathbf{H}^l(\mathbf{w}^i) = h^l(g(\mathbf{W}_e^{i-1}, \mathbf{W}_p^{i-1}), \mathbf{H}^l(\mathbf{w}^{i-1}))$, and (iii) converts the feature into a probability using a linear classifier by predicting the next token \mathbf{w}^i .

Moreover, we aim to generate evidence that can discriminate the correct answer from option distractors. Hence, we add the answer prediction loss into the objective to fine-tune the language model. The text input for the j th option is $\mathbf{x}_j = [\mathbf{P} \text{ [SEP]} \mathbf{Q} \text{ [SEP]} \mathbf{O}_j]$. We use all N samples to optimize the

following objective (with a regularization term λ):

$$\mathcal{L}_{class} = \sum_{(x,y) \in \{\mathcal{X}, \mathcal{Y}\}} \log(\text{Softmax}(\mathbf{H}^L(\mathbf{w}^0) \mathbf{W}_y)), \quad (2)$$

$$\mathcal{L}_{total} = \mathcal{L}_{gen} + \lambda * \mathcal{L}_{class}, \quad (3)$$

where $\mathbf{H}^L(\mathbf{w}^0)$ is the last layer feature of the first token and \mathbf{W}_y is the parameters to learn to predict label y .

Test stage: we only use $[\mathbf{P} \text{ [SEP]} \mathbf{Q}]$ as the input to the language model and use the model to generate the next sentence as an evidence which means model is agnostic to the correct answer.

4.1.2 Factual Evidence Generator

Aside from the textual evidence that contains information about the facts of daily life routine, relations between the facts are also important for question answering. In this section, we propose to utilize a factual knowledge graph to extract facts and relations and use them as additional evidence. Specifically, we use the ConceptNet (Speer et al., 2017)¹ as the base model. We use a knowledge graph completion algorithm Bosselut et al. (2019) to find new relations to further improve the quality of the generated factual evidence.

We define $X^s = \{x_0^s, \dots, x_{|s|}^s\}$ as the subject, $X^r = \{x_0^r, \dots, x_{|r|}^r\}$ as the relation, and $X^o = \{x_0^o, \dots, x_{|o|}^o\}$ as the object. We use the $[X^s \text{ [SEP]} X^r \text{ [SEP]} X^o]$ triplets as the input to the knowledge graph completion language model in Bosselut et al. (2019) to generate additional triplets that contain new subject and object relations. To generate factual evidence, we first extract entities from the given data \mathcal{X} . We then select the related entities that match the subject X^s in forms of subject-relation-object triplets. After that, we filter the triplets by selecting the subject X^{s*} that follows: (i) part-of-speech (POS) tag of X^{s*} word matches the POS tag of the entity word; (ii) subject X^{s*} word frequency is less than the word frequency of the object X^o plus a threshold K^o ; (iii) subject X^{s*} word is not in the top- K frequent words based on the word frequency table²; and (iv) the relation X^r in the (X^{s*}, X^r, X^o) triplets connects no more than K^r objects from the same subject X^{s*} . K , K^o and K^r are the hyper-parameters. Finally, we convert the

¹ConceptNet is a knowledge graph, which consists of triples obtained from the Open Mind Common Sense entries.

²<https://www.wordfrequency.info/free.asp>

filtered triplets into a nature language sequences as our factual evidences. For example, “(trouble, Partof, life)” would be converted to “trouble is part of life”.

4.2 Model Learning with Contextual Commonsense Reasoning

After the relevant reasoning evidences are generated, the goal is to combine the evidence with the given data and then build a reasoning model to make a selection for the correct answer. In the following, we introduce our proposed contextual commonsense reasoning module, which utilizes contextual encoding, evidence injection and a capsule network components to make the prediction.

Contextual Encoding Recently, RoBERTa (Liu et al., 2019) has shown to be effective and powerful in many natural language processing tasks and it is potentially beneficial for generating deep contextual features as well. Here, we use RoBERTa as an intermediate component to generate hidden representation of paragraph, question, the ith option and evidence $[\mathbf{H}_{\text{cls}}^i, \mathbf{H}_P^i, \mathbf{H}_{\text{sep}}^i, \mathbf{H}_Q^i, \mathbf{H}_{\text{sep}}^i, \mathbf{H}_{O_i}^i, \mathbf{H}_{\text{sep}}^i, \mathbf{H}_E^i] = \text{Encode}([[CLS], \mathbf{P}, [SEP], \mathbf{Q}, [SEP], \mathbf{O}_i, [SEP], \mathbf{E}])$. We use the last layer of the RoBERTa model to encode, and thus the function $\text{Encode}(\cdot)$ returns the last layer features for each token. The corresponding features of paragraph, question, option and evidence are $\mathbf{H}_P^i \in \mathcal{R}^{d \times t}$, $\mathbf{H}_Q^i \in \mathcal{R}^{d \times n}$, $\mathbf{H}_{O_i}^i \in \mathcal{R}^{d \times h}$ and $\mathbf{H}_E^i \in \mathcal{R}^{d \times k}$, where d is the dimension of the feature. Since we have m options, we have m set of features.

Evidence Injection Given the previously generated evidence representation \mathbf{H}_E^i . We aim to integrate it with the paragraph \mathbf{H}_P^i , question \mathbf{H}_Q^i and option $\mathbf{H}_{O_i}^i$. Here, we adopt the attention mechanism used in QANet (Yu et al., 2018) to model the interaction between \mathbf{H}_E^i and the paragraph \mathbf{H}_P^i :

$$\mathbf{S}_{iP}^E = \text{Att}(\mathbf{H}_E^i, \mathbf{H}_P^i) = \text{Softmax}(\mathbf{H}_P^{iT} \mathbf{W}_g \mathbf{H}_E^i) \quad (4)$$

$$\mathbf{G}_{iP}^E = \mathbf{H}_E^i \mathbf{S}_{iP}^{ET}, \quad (5)$$

where $\mathbf{W}_g \in \mathcal{R}^{d \times d}$ is the bi-linear model parameter matrix. Since $\mathbf{S}_{iP}^E \in \mathcal{R}^{t \times k}$ is the activation map (attention weights) between each token in \mathbf{P} and each token in \mathbf{E} , the learned relation representation $\mathbf{G}_{iP}^E \in \mathcal{R}^{d \times t}$ of the paragraph \mathbf{P} contains evidence information \mathbf{E} . The other two relations \mathbf{G}_{iP}^Q and $\mathbf{G}_{iP}^{O_i}$ regarding \mathbf{P} can be generated accordingly. Similarly, we can model the other interactions for

question \mathbf{Q} as $\mathbf{G}_{iQ}^P, \mathbf{G}_{iQ}^E, \mathbf{G}_{iQ}^{O_i}$, and each option \mathbf{O}_i as $\mathbf{G}_{iO_i}^Q, \mathbf{G}_{iO_i}^E$ and $\mathbf{G}_{iO_i}^P$.

To incorporate the relation information, we use the co-matching algorithm introduced in Wang et al. (2018b) to generate the final representation of the input. First, we obtain the matching result between the paragraph and the question as follows:

$$\mathbf{M}_{iP}^Q = (W_m [\mathbf{G}_{iP}^Q \ominus \mathbf{H}_P^i; \mathbf{G}_{iP}^Q \odot \mathbf{H}_P^i] + b_m \otimes \mathbf{1})^+, \quad (6)$$

where $(\cdot)^+$ denotes ReLU function, $\mathbf{1} = [1, 1, \dots, 1]^T \in \mathcal{R}^{t \times 1}$ is vector of all ones, and $W_m \in \mathcal{R}^{d \times 2d}$ and $b_m \in \mathcal{R}^{d \times 1}$ are the model parameters. Following Tai et al. (2015) and Wang et al. (2018b), we use notation \ominus and \odot as the element-wise subtraction and multiplication between two matrices and \otimes as outer product of two vectors. Similarly, we can obtain the other pairs as $\mathbf{M}_{iP}^E, \mathbf{M}_{iP}^{O_i}, \dots, \mathbf{M}_{iO_i}^P$. In the next step, we concatenate all the pairs regarding \mathbf{P} as

$$\mathbf{C}_{iP} = [\mathbf{M}_{iP}^Q : \mathbf{M}_{iP}^{O_i} : \mathbf{M}_{iP}^E] \in \mathcal{R}^{3d \times t}, \quad (7)$$

where $[:]$ denotes the vertical concatenation operation. Each column \mathbf{c}_i is the co-matching state that concurrently matches a paragraph token with the question, candidate option and the evidence. Accordingly, we can obtain the question representation \mathbf{C}_{iQ} and option representation \mathbf{C}_{iO_i} . Finally, we concatenate them all to obtain the final representation $\mathbf{F} = [\mathbf{C}_1, \dots, \mathbf{C}_m] \in \mathcal{R}^{3d \times m(t+n+h)}$, where each $\mathbf{C}_i = [\mathbf{C}_{iP}, \mathbf{C}_{iQ}, \mathbf{C}_{iO_i}] \in \mathcal{R}^{3d \times (t+n+h)}$.

Since the final representation only contains the fine-grid token-level information, we employ a convolutional neural network (CNN) to extract higher level (phrase-level) patterns. To generate phrase patterns with different size, we use two convolutional kernels: size 1×2 with stride 2 and size 1×4 with stride 4 to convolve with \mathbf{F} along the dimension of hidden state. In other words, such an operation extracts non-overlapping moving windows on \mathbf{F} with window size 2 and 4.

$$\begin{aligned} \mathbf{R}_1 &= \text{MaxPooling}_{1 \times 2}\{\text{CNN}_{1 \times 2}(\mathbf{F})\} \\ \mathbf{R}_2 &= \text{MaxPooling}_{1 \times 1}\{\text{CNN}_{1 \times 4}(\mathbf{F})\} \end{aligned}$$

To ensure \mathbf{R}_1 and \mathbf{R}_2 have the same dimension, we use a max pooling of size 1×2 with stride 2 for \mathbf{R}_1 and a max pooling of size 1×1 with stride 1 for \mathbf{R}_2 . We concatenate \mathbf{R}_1 and \mathbf{R}_2 to generate phrase-level representation $\mathbf{L} = [\mathbf{R}_1, \mathbf{R}_2] \in \mathcal{R}^{3d \times m((t+n+h)/2)}$.

With \mathbf{L} , to predict the final answer, one of the commonly applied operation is to simply take the maximum over the hidden dimension of length $(t + n + h)/2$. However, the max operation only consider the most significant phrase for each candidate without aware of the others. To explore the correlation between options and dynamically select the optimal one, we use dynamic routing-by-agreement algorithm represented in Sabour et al. (2017). Specifically, we convert \mathbf{L}_i to a capsule \mathbf{v}_j using the following steps:

$$\hat{\mathbf{L}}_{j|i} = W_{ij}\mathbf{L}_i, \quad \mathbf{s}_j = \sum_{i=1}^{(t+n+h)/2} c_{ij} \cdot \hat{\mathbf{L}}_{j|i},$$

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2} \frac{\mathbf{s}_j}{\|\mathbf{s}_j\|},$$

where \mathbf{L}_i is the i th column vector of \mathbf{L} , affine transformation matrix W_{ij} and weighting c_{ij} are the capsule network model parameters. The learned $\hat{\mathbf{L}}_{j|i}$ denotes the “vote” of the capsule j for the input capsule i . The agreement of “prediction vector” $\hat{\mathbf{L}}_{j|i}$ between the current j th output and i th parent capsule is captured by the coupling coefficients c_{ij} . The value of c_{ij} would increase if higher level capsule \mathbf{s}_j and lower lever capsule \mathbf{L}_i highly agreed.

Model Learning If an option \mathbf{O}_j is the correct answer, we would like the top-level capsule \mathbf{v}_j to have a high energy, otherwise, we expect the energy of \mathbf{v}_j to be low. Since the L_2 -norm (square root of the energy) of the capsule vector \mathbf{v}_j represents the scoring of how likely the j th candidate is the correct answer, we use the following loss function (Sabour et al., 2017) to learn the model parameters:

$$\mathcal{L}_{\text{pre}} = \sum_{j=1}^m \{y_i \cdot \max(0, m^+ - \|\mathbf{v}_j\|)^2 + \lambda_1(1 - y_i) \max(0, \|\mathbf{v}_j\| - m^-)^2\} \quad (8)$$

where λ_1 is a down-weighting coefficient, m^+ and m^- are margins. In our experiments, we set $m^+ = 0.9$, $m^- = 0.1$, $\lambda_1 = 0.5$.

5 Experiments

In the experiment, we evaluate the performance of our proposed CEGI framework from different aspects, including evidence generation tasks and the answer prediction of contextual commonsense reasoning tasks.

5.1 Dataset and Baseline

CosmosQA is the dataset that is designed for reading comprehension with commonsense reasoning

(Huang et al., 2019). Samples are collected from people’s daily narratives and the type of questions are concerning the causes or effects of events. Particularly answering the questions require contextual commonsense reasoning over the considerably complex, diverse, and long context. In general, the dataset contains a total of 35.2K multiple-choice questions, including 25262 training samples, 2985 development samples, and 6963 testing samples.³ **Baseline** We categorize baseline methods into the following three groups: 1. Co-Matching (Wang et al., 2018b), Commonsense-RC (Wang et al., 2018a), DMCN (Zhang et al., 2019a), Multiway (Huang et al., 2019). 2. GPT2-FT (Radford et al., 2018), BERT-FT (Devlin et al., 2018), RoBERTa-FT (Liu et al., 2019). 3. Commonsense-KB (Li et al., 2019), K-Adapter (Wang et al., 2020). The baseline details are in appendix A.2.

Table 1: Comparison of approaches on CosmosQA (Accuracy %) from the AI2 Leaderboard. T+F means using generated textual and factual evidence together.

Model	Dev	Test
Co-Matching (Wang et al., 2018b)	45.9	44.7
Commonsense-RC (Wang et al., 2018a)	47.6	48.2
DMCN (Zhang et al., 2019a)	67.1	67.6
Multiway (Huang et al., 2019)	68.3	68.4
GPT-FT (Radford et al., 2018)	54.0	54.4
BERT-FT (Devlin et al., 2018)	66.2	67.1
RoBERTa-FT (Liu et al., 2019)	79.4	79.2
Commonsense-KB (Li et al., 2019)	59.7	\
K-Adapter (Wang et al., 2020)	81.8	\
CEGI(T+F)	83.8	83.6
Human	\	94.0

5.2 Experimental Results and Analysis

Table 1 shows the performance of different approaches reported on the AI2 Leaderboard.⁴ Comparing to all methods, our proposed model CEGI(T+F) has the highest accuracy on both development set and test set. Most of the reading comprehension approaches utilize the attention mechanism to capture the correlations between paragraph, question and option set, therefore, the model tends to select the one option that is semantically closest to the paragraph. Among all of the group 1 methods, Multiway has the highest accuracy of 68.3%.

³The CosmosQA dataset can be obtained from <https://leaderboard.allenai.org/cosmosqa/>

⁴<https://leaderboard.allenai.org/cosmosqa/> The test dataset is hidden by the AI2 and methods like Commonsense-KB and K-Adapter are not reported on the Leaderboard.

Group 2 methods consider deep contextual representation of the given paragraph, question and option set, and increase the performance. Comparing group 2 methods with group 1 methods, RoBERTa-FT, which uses dynamic masking and large mini-batches strategy to train BERT, gains 11.1% accuracy increase compared to Multiway.

However, it is worthy to mention that more than 83% of correct answers are not in the given passages in the CosmosQA dataset. Hence, multi-choice reading comprehension models do not gain big improvement as they tend to select the choice which has the most overlapped words with the paragraph without commonsense reasoning. Even though, group 2 methods consider connecting the paragraph with question and option through a deep bi-directional strategy, the reasoning for question answering is still not well-addressed in the models. By utilizing additional knowledge, Commonsense-KB or K-Adapter teach pretrained models with commonsense reasoning. K-Adapter gains 2.4% accuracy increase than RoBERTa-FT. Those approaches leverage the structured knowledge but fail to produce a prominent prediction improvement. Comparing our CEGI approach with RoBERTa, we gain a 4% increase and 2% increase than K-Adapter, which demonstrates that injecting evidence is beneficial and incorporating interactive attentions can further enhance the model.

5.3 Evidence Evaluation

In this section, we investigate the generated evidence from the textual generator and factual generator. Moreover, we study the quality of the generated evidence on another dataset—CommonsenseQA.

5.3.1 Textual Evidence Generator

Dataset Open Mind Common Sense (OMCS) corpus (Singh et al., 2002) is a crowd-sourced knowledge database of commonsense statements⁵, where its English dataset contains a million sentences from over 15,000 contributors. We consider using this dataset to pretrain the textual evidence generator and using CosmosQA to fine-tune the generator.

Setup We use both BERT and GPT2 model to generate evidence and compare the results. To obtain a language model that contains representation of facts, we first pretrain both models with the OMCS data using the loss function in Eq. 1. Then we use

⁵<https://github.com/commonsense/conceptnet5/wiki/Downloads>

CosmosQA data to fine-tune the pretrained model using multi-task loss in Eq. 3.

Metrics In line with prior work (Wang and Cho, 2019), we evaluate the performance of evidence generation based on quality and diversity. In terms of quality, we follow Yu et al. (2017) and compute the BLEU score between the generated evidence and the ground truth evidence to measure the similarity. The perplexity (PPL) score is also reported as a proxy for fluency. In terms diversity, we consider using self-BLEU (Zhu et al., 2018b), which measures how similar between two generated sentences. Generally, a higher self-BLEU score implies that the model has a lower diversity.

Results From Table 2, we observe that, compared to CEGI-GPT2, the CEGI-BERT generator has higher diversity (Self-BLEU decreases 4 for bi-gram and decreases 2.1 for tri-gram) but lower quality (BLEU decreases 1.3 for tri-gram and PPL increases 27.1). Even though the perplexity on CEGI-BERT is as good as CEGI-GPT2, after reading the samples, we find out that many of the generated language are fairly coherent. For a more rigorous measure of generation quality, we collect human judgments on sentences for 100 samples using a four-point scale (the higher the better). For each sample, we ask three annotators to rate the sentence on its fluency and take the average of the three judgments as the sentence’s fluency score. For CEGI-BERT and CEGI-GPT2, we get mean scores of 3.21, 3.17 respectively. Those results imply that generated evidence are semantically consistent with the correct evidence and can be used as auxiliary knowledge for the reasoning step.

Table 2: Generation performance on CosmosQA.

Model	Quality		Diversity	
	BLEU(\uparrow) n=2	PPL(\downarrow) n=3	Self-BLEU(\downarrow) n=2	n=3
CEGI-BERT	40.8	32.2	153.8	30.5 14.7
CEGI-GPT2	39.8	33.5	126.7	34.2 16.6

Table 3: Generation performance on ConceptNet

Model	PPL	Score	N/T sro	N/T o
LSTM-s	\	60.83	86.25	7.83
CKBG	\	57.17	86.25	8.67
CEGI-BERT	4.89	92.19	65.32	4.12
CEGI-GPT2	4.58	93.89	61.72	3.90

5.3.2 Factual Evidence Generator

Dataset ConceptNet⁶ is a commonsense knowledgebase of the most basic things a person knows. We use the 100K version of the training set in ConceptNet, which contains 34 relation types, to train the factual evidence generator. Tuples within the data are in the standard s, r, o form.

Setup We set s and r as input for both GPT2 and BERT and use them to generate the new object o . To compare with our proposed GPT2 model and BERT model, we include a LSTM model (LSTM-s) and the BiLSTM model (CKBG) in (Saito et al., 2018). We train the LSTM model to generate o , and we train the CKBG model from both directions: s, r as input and o, r as input.

Metrics Similar to the textual evidence generation task, we use PPL to evaluate our model on relation generation. To evaluate the quality of generated knowledge, we also report the number of generated positive examples that are scored by the Bilinear AVG model (Li et al., 2016). “N/T sro” and “N/T o” are the proportions of generated tuples and generated objects which are not in the training set.

Results As we observed from Table 3, CEGI-GPT2 has the lowest PPL (4.58) and highest score (93.89), which indicates the CEGI-GPT2 model is confident and accurate at the generated relations. Even though the generated tuples on LSTM-s and CKBG model has high “N/T sro” (both are 86.25%) and “N/T o” (7.83% and 8.67% respectively), which means they generate novel relations and expand the knowledge graph, the generated nodes and relations may not be correct. We still need to rely on the Score to evaluate and they do poorly (60.83% and 57.17% respectively) in terms of Score. Since our proposed CEGI-GPT2 and CEGI-BERT model have high Score and low PPL, we believe that both models can produce high-quality knowledge and still be able to extend the size of the knowledge graph.

5.3.3 Evidence Evaluation on CommonsenseQA

CommonsenseQA⁷ is a multi-choice question answering dataset, which contains roughly 12K questions with one correct answer and four distractor answers. Since the CommonsenseQA data only requires different types of commonsense knowledge to predict the correct answers, it does not contain

paragraphs compared to CosmosQA. We use our textual generator and factual generator to generate evidence using CommonsenseQA data and use that to test the performance on answer prediction. To train our proposed textual evidence generator, we use Cos-e⁸ as the ground truth evidence. Cos-e uses Amazon Mechanical Turk to provide reasoning explanations for the CommonsenseQA dataset. To train our proposed factual evidence generator, we follow the same procedure as described in subsection 4.1.2. To predict the answer based on both evidence, we prepare the input as [Q [SEP] O_i [SEP] E] to the RoBERTa model.

Baselines KagNet (Lin et al., 2019), Cos-E (Rajani et al., 2019), DREAM (Lv et al., 2019), RoBERTa + KE, RoBERTa + IR and RoBERTa + CSPT (Lv et al., 2019). All baselines use extracted knowledge from ConceptNet or Wikipedia. The details are in the appendix A.2.

Table 4: Accuracy (%) of different models on CommonsenseQA development set

Model	Acc
KagNet (Lin et al., 2019)	62.4
Cos-E (Rajani et al., 2019)	64.7
DREAM (Lv et al., 2019)	73.0
RoBERTa+CSPT (Lv et al., 2019)	76.2
RoBERTa+KE (Lv et al., 2019)	77.5
RoBERTa+IR (Lv et al., 2019)	78.9
RoBERTa + T	78.8
RoBERTa + F	77.6
RoBERTa + (T+F)	79.1

Result Results on CommonsenseQA datasets are summarized in Table 4. RoBERTa + T, RoBERTa + F and RoBERTa + (T+F) includes textual evidence, factual evidence and both evidence together respectively. We observe that our model RoBERTa + T and RoBERTa + F can produce competitive performance compared to all baselines. By utilizing both textual knowledge and factual knowledge, our approach outperforms RoBERTa+IR and achieves the highest accuracy 79.1%.

5.4 Ablation Study

To evaluate the contributions of individual components of our proposed framework, we use an ablation study. Table 5 summarizes ablation studies on the development set of CosmosQA from several aspects: the influence of the generated evidence; which evidence is better, textual or factual; the influence of the capsule network.

⁶<https://ttic.uchicago.edu/~kgimpel/commonsense.html>

⁷<https://www.tau-nlp.org/commonsenseqa>

⁸<https://github.com/salesforce/cos-e>

Result We can see that injecting generated explainable evidence can help the model achieve a better performance in terms of accuracy. Using generated textual evidence and factual evidence together can benefit more. Using capsule network significantly improves the reasoning performance, we doubt that is due to the hierarchical structure information from both token-level and phrase-level are extracted by capsule network.

Table 5: Accuracy (%) of different models on Cosmos development set. ✓ means selecting the module.

Model	Text	Fact	Capsule	Co-Att	Acc
CEGI	✓	✓	✓	✓	83.8
CEGI-V1	✓		✓	✓	83.4
CEGI-V2		✓	✓	✓	83.2
CEGI-V3			✓	✓	82.6
CEGI-V4	✓	✓			82.2
RoBERTa-FT					79.4

6 Conclusion

In this paper, we proposed a commonsense evidence generation and injection model to tackle reading comprehension. Both textual and factual evidence generators were used to enhance the model for answering questions which requires commonsense reasoning. After the evidences were generated, we adopted attention mechanism to find the relation and match between paragraph, question, option and evidence. We used convolutional network to capture the multi-grained features. To capture diverse features and iteratively make a decision, we proposed using a capsule network that dynamically capture different features to predict the answer. The AI2 Leaderboard of CosmosQA task demonstrated that our method can tackle commonsense-based reading comprehension pretty well and it outperformed the current state-of-the-art approach K-Adapter with a 2% increase in term of accuracy. Experiments regarding the evidence generators showed that the generated evidence is human-readable and those evidences are helpful for the reasoning task.

7 Acknowledge

This work is supported in part by NSF under grants III-1526499, III-1763325, III-1909323, and CNS-1930941.

References

- Henning Andersen. 1973. Abductive and deductive change. *Language*, pages 765–793.
- Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.
- Eugene Charniak and Solomon Eyal Shimony. 1990. *Probabilistic semantics for cost based abduction*. Brown University, Department of Computer Science.
- Zhipeng Chen, Yiming Cui, Wentao Ma, Shijin Wang, and Guoping Hu. 2019. Convolutional spatial attention model for reading comprehension with multiple-choice questions. *Proceedings of the AAAI*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.
- Jerry R Hobbs, Mark E Stickel, Douglas E Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial intelligence*, 63(1-2):69–142.
- Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.
- Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2019. Informing unsupervised pretraining with external linguistic knowledge. *arXiv preprint arXiv:1909.02339*.
- Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2019. Sensebert: Driving some sense into bert. *arXiv preprint arXiv:1908.05646*.
- Shiyang Li, Jianshu Chen, and Dian Yu. 2019. Teaching pretrained models with commonsense reasoning: A preliminary kb-based approach. *arXiv preprint arXiv:1909.09743*.
- Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th ACL (Volume 1: Long Papers)*, pages 1445–1455.
- Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

- Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Dixin Jiang, Gui-hong Cao, and Songlin Hu. 2019. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. *arXiv preprint arXiv:1909.05311*.
- Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.
- Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *arXiv preprint arXiv:1911.03681*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.
- Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*.
- Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *NeurIPS*, pages 3856–3866.
- Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. 2018. Commonsense knowledge base completion and generation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 141–150.
- Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences “On the Move to Meaningful Internet Systems”*, pages 1223–1237. Springer.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceddings of the AAAI*.
- Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.
- Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2019. olmpics-on what language model pre-training captures. *arXiv preprint arXiv:1912.13283*.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.
- Min Tang, Jiaran Cai, and Hankz Hankui Zhuo. 2019. Multi-matching network for multiple choice reading comprehension. *Proceddings of the AAAI*.
- Alex Wang and Kyunghyun Cho. 2019. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094*.
- Liang Wang, Meng Sun, Wei Zhao, Kewei Shen, and Jingming Liu. 2018a. Yuanfudao at semeval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. *arXiv preprint arXiv:1803.00191*.
- Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Dixin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.
- Shuohang Wang, Mo Yu, Shiyu Chang, and Jing Jiang. 2018b. A co-matching model for multi-choice reading comprehension. *arXiv preprint arXiv:1806.04068*.
- Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv preprint arXiv:1912.09637*.
- Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.
- Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceddings of the AAAI*.
- Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.
- Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019a. Dcmn+: Dual co-matching network for multi-choice reading comprehension. *arXiv preprint arXiv:1901.09381*.
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019b. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Haichao Zhu, Furu Wei, Bing Qin, and Ting Liu. 2018a.
Hierarchical attention flow for multiple-choice reading comprehension. In *Proceedings of the AAAI*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo,
Weinan Zhang, Jun Wang, and Yong Yu. 2018b.
Txygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference*, pages 1097–1100. ACM.

A Appendix

A.1 Training Details

In CosmosQA experiments, we use pretrained weight of RoBERTa_large. We run experiments on a 24G Titan RTX for 5 epochs, set the max sequence length to 256. For hyper-parameters, we set the routing iterations of capsule network as 3, batch size is chosen from {8, 16, 24, 32}, learning rate is chosen from {2e-5, 1e-5, 5e-6} and warmup proportion is chosen from {0, 0.1, 0.2, 0.5}. For CEGI(F+L), the best performance is achieved at batch size=24, lr=1e-5, warmup_proportion=0.1 with 16-bit float precision. GPT2 with 12-layer and BERT_base model are used in evidence generation. In textual evidence generation, we set λ in Eq. 3 to 0.5, max sequence length to 40, batch size to 32 and the learning rate to 6.25e-0.5. In factual evidence generation, we set max sequence length to 15, batch size to 64, the learning rate to 1e-5. For both generators, we train 100000 iterations with early stop.

A.2 Baseline Methods

Cosmos Baselines

1. **Co-Matching** (Wang et al., 2018b) captures the interactions between paragraph with question and option set through attention. **Commonsense-RC** (Wang et al., 2018a) performs three-way unidirectional attention to model interactions between paragraph, question, and option set. **DMCN** (Zhang et al., 2019a) applies dual attention between paragraph and question or option set using BERT encoding output. **Multiway** (Huang et al., 2019) uses BERT to learn the semantic representation and uses multiway bidirectional interaction between each pair of input paragraph, question and option set.
2. **GPT2-FT** (Radford et al., 2018), **BERT-FT** (Devlin et al., 2018) and **RoBERTa-FT** (Liu et al., 2019) are the pretrained transformer language models with additional fine-tuning steps on CosmosQA.
3. **Commonsense-KB** (Li et al., 2019) uses logic relations from a commonsense knowledge base (e.g., ConceptNet⁹) with rule-based method to generate multiple-choice questions as additional training data to fine-tune the pretrained BERT model. **K-Adapter** (Wang et al., 2020) infuses commonsense knowledge into a large pre-trained

network.

CommonsenseQA Baselines

KagNet (Lin et al., 2019) uses ConceptNet as extra knowledge and proposes a knowledge-aware graph network and finally scores answers with graph representations. **Cos-E** (Rajani et al., 2019) constructs human-annotated evidence for each question and generates evidence for test data. **DREAM** (Lv et al., 2019) adopts XLNet-large as the baseline and extracts evidence from Wikipedia. **RoBERTa + KE**, **RoBERTa + IR** and **RoBERTa + CSPT** (Lv et al., 2019) adopt RoBERTa as the baseline and utilize the evidence from Wikipedia, search engine and OMCS, respectively.

A.3 Case Study

To verify the generated evidence performance, we perform case studies on textual generator and factual generator. In addition, we also show a case that the proposed capsule network can help to select the answer by comparing with the other options.

P: My favorite part of the job is training and handling the animals. I really like that they are trusting me to be able to handle some of the animals without supervision. There is always someone around if I need help, but they are n't overseeing it like they were in the beginning, which makes me feel like they trust me which is important for a work environment. It makes me feel like I 've earned a place there ... and I believe I have.

Q: What may be your reason for thinking you 've earned your place there ?

- A: None of the above choices .
- B: They told me that they trust me with the animals.
- C: They delegate supervision tasks to me.
- D: They delegate tasks to me without supervision.

Evidence:

CEGI-BERT: I am an expert at handling animals.
CEGI-GPT2: They trust me and I handle animal without supervision.

P: You fell asleep in my arms, a few hours later and I took to watching you the whole night. My mind was filled with nothing but thoughts of you, how holding you makes me forget everything else and suddenly I realize. We are meant to be together.

Q: What might have happened had you not shared an intimate moment with her?

- A: I would have realized how much I love her at a later stage because I love her anyway.
- B: I would have gone to bed without her.
- C: None of the above choices.
- D: I would not have realized how much I love her and want to be with her.

Evidence:

CEGI-BERT: I would not have experienced the feelings that I had for her.
CEGI-GPT2: He might not have gotten to have a romantic moment with her.

Figure 3: Examples of textual evidence generator.

Case Study on Textual Generator We show examples of automatically generated evidences by

⁹<http://conceptnet.io/>

CEGI-GPT2 and CEGI-BERT in Figure 3. We observe that using the multi-tasking loss, CEGI-BERT and CEGI-GPT2 generate more accurate evidence. Moreover, using those generated evidences is helpful for predicting the correct answer. In the first example, the evidence generated by CEGI-GPT2 “They trust me and I handle animal without supervision.” can help select the Answer D “They delegate tasks to me without supervision.” In the second example, the evidence generated by CEGI-BERT “I would not have experienced the feelings that I had for her.” is close to the Answer D.

P: My nephew hates bees, and he moved over to my sisters ' house. He was trying to tell me the world did n't need bees. I told him that most plant and animal life would die within a decade of bees disappearing from the planet.

Q: What 's a possible reason why the nephew hates bees ?

- A: Because he moved to the writer 's sister 's house.
- B: Because most plant and animal life would die within decade of bees disappearing from the planet .
- C: Because he got bite before.**
- D: None of the above choices.

Evidence:

<bee, Desires, flower> <bee, Capableof, sting> <bee, Capableof, buzz> <bee, AtLocation, any garden> <planet, HasProperty, beautiful> <planet, IsA, orbitting sun> <planet, AtLocation, solar system>, <planet, ReceivesAction, fill with sand>

P: Also, if he were to clean the interior first, he wud hafta remove the body kits and put them aside as the interior is the hardest part of any vehicle wash, having to remove oil, dirts, grease and what - nots. Soon, he stepped up to do the spray job first. Everything seemed well as he was left to spray the finished smooth surface. He began spraying right under the sun.

Q: Why does he feel that he must perform the spray job while the sun beats down?

- A: None of the above choices.
- B: The sun assists in warming the paint so it is easy to apply.
- C: The sun keeps the paint from spilling off the car.
- D: The sun dries the paint which is sprayed on quickly.**

Evidence:

<sun, Capableof, dry something that be wet>, <spray, HasProperty, wet>, <spray Atlocation, waterfall>, <interior, HasProperty, inside>, <vehicle, UsedFor, transportation>, <vehicle, Capableof, travel>, <vehicle, UsedFor, mobility>

Figure 4: Examples of factual evidence generator.

Case Study on Factual Generator Figure 4

shows the examples of evidences generated by the factual generator. In the first example, from evidence, we know “bee is capable of sting”, so option C “Because he got bite before” will be the correct answer. Some options like B “Because most plant and animal life would die within decade of bees disappearing from the planet” appear in the context “I told him that most plant and animal life would be die within a decade of bees disappearing from the planet”, and thus without the evidence it could puzzle the model to select B. In the second example, we have the evidence “sun has capable of drying something that be wet” and “spray has property wet”, so it is easy to reach the correct answer D “The sun dries the paint which is sprayed on quickly”.

Case Study on Capsule Network We investigate the case with and without capsule network in the model. As shown in Figure 5, it is hard to answer the question simply by reading through the paragraph. However, after comparing with the other options, option A will be the best answer. In this case, the generated evidence is not useful to predict the correct answer A. But the capsule network considering all other candidate options when answering the question can help predict “She wanted her to look at a pretty rock” as answer.

P: Last night just at twilight to be exact my daughter , her little head all sweaty from running around like a little maniac with the other children in the neighborhood came bounding into the house skidding to a halt in the kitchen , \" Mommy ! Look ! Is n't it beautiful ! \"

Q: What did her daughter want her to look at ?

- A: She wanted her to look at a pretty rock.**
- B: The daughter wanted her to look at the sun going down.
- C: None of the above choices.
- D: She wanted her to see the other pretty children.

Figure 5: Example of capsule network predict correctly while without capsule network predict wrongly.

Identifying Collaborative Conversations using Latent Discourse Behaviors

Ayush Jain, Maria Leonor Pacheco, Steven Lancette,
Mahak Goindani and Dan Goldwasser

Department of Computer Science, Purdue University

{jain207, pachecog, slancett, mgoindan, dgoldwas}@purdue.edu

Abstract

In this work, we study collaborative online conversations. Such conversations are rich in content, constructive and motivated by a shared goal. Automatically identifying such conversations requires modeling complex discourse behaviors, which characterize the flow of information, sentiment and community structure within discussions. To help capture these behaviors, we define a hybrid relational model in which relevant discourse behaviors are formulated as discrete latent variables and scored using neural networks. These variables provide the information needed for predicting the overall collaborative characterization of the entire conversational thread. We show that adding inductive bias in the form of latent variables results in performance improvement, while providing a natural way to explain the decision.

1 Introduction

Online conversations are rampant on social media channels, news forums, course websites and various other discussion websites consisting of diverse groups of participants. While most efforts have been directed towards identifying and filtering negative and abusive content (Wang and Cardie, 2014; Wulczyn et al., 2017; Zhang et al., 2018), in this paper we focus on characterizing and automatically identifying the positive aspects of online conversations (Jurafsky et al., 2009; Niculae and Danescu-Niculescu-Mizil, 2016; Nápoles et al., 2017a). We specifically focus on *collaborative conversations*, which help achieve a shared goal such as gaining new insights about the discussion topic like response informativeness, engagement etc.

Rather than looking at the outcomes of such conversations (e.g., task completion (Niculae and Danescu-Niculescu-Mizil, 2016)), we analyze conversational behaviors, specifically looking at indications of *collaborative* behavior that is conducive to

group learning and problem-solving. These include purposeful interactions centered around a specific topic, as well as open and respectful exchanges that encourage participants to elaborate on previous ideas. To help clarify these concepts, consider the following conversation snippet.

User A : We should invest in more resources to encourage young people to be responsible citizens.

Response Option 1 : I wonder if more initiatives at grassroots level can help them to identify and understand issues of their local community more deeply.

Response Option 2 : Good point, I agree.

We compare the two possible responses to User A's post. Option 1 offers a balanced contribution, developing the idea presented in the original post and allowing the conversation to proceed. Option 2, while polite and positive, is *not* collaborative as the initial idea is not expanded on. In fact, agreement is often used as a polite way to end conversations without contributing additional content. Despite the positive sentiment, capturing the absence of balanced content contribution and the absence of idea development as different discourse behaviors, one can infer that it is not a collaborative conversation.

While humans could tell the two apart by detecting constructive discourse behaviors, automatically capturing these behaviors is highly challenging. Anecdotal evidence, collected by extracting features from conversation transcripts, can lead to conflicting information, as identifying collaborative behavior relies on complex interactions between posts. Our main intuition in this paper is that reasoning and enforcing consistency over these behaviors can help capture the conversational dynamics and lead to more accurate predictions.

Our technical approach follows this intuition. We design a hybrid relational model that combines neural networks and declarative inference

to capture high-level discourse behaviors. Since we only have access to the raw conversational text, we model these behaviors as discrete latent variables, used to support and justify the final decision – whether the conversation is collaborative or not.

Explicitly modeling discourse behaviors as latent variables allows us to add inductive bias, constraining the representation learned by the neural model. It also provides a natural way to “debug” the learning process, by evaluating the latent variables activation. Our experiments show that the joint model involving global learning of different latent discourse behaviors improves performance. We use the Yahoo News Annotated Comments Corpus (Napoles et al., 2017b), and expanded the annotation for the collaborative task.¹

2 Task Definition

Collaborative conversations are purposeful interactions, often revolving around a desired outcome, in which interlocutors build on each others’ ideas to help move the discussion forward. Collaborative conversations are an important tool in collaborative problem solving (Greiff, 2012) and require collaboration skills (Flor et al., 2016; Hao et al., 2016). We focus on identifying indicators of successful collaboration. We build on the work of Napoles et al. 2017a, who released a dataset annotated for engaging, respectful and informative conversations, and annotate it for collaborative conversations, in which participants build on each other’s words, provide constructive critique, elaborate on suggested ideas, generalizing them and synthesizing new ideas and knowledge in the process.

During the annotation process, we identified several repeating behaviors (detailed below) that helped characterize and separate between collaborative and non-collaborative conversations.

2.1 Non-Collaborative Discourse Behaviors

(A) Low Idea Development users who: (1) deviate from the thread topic and change the topic, (2) ignore previously raised ideas and give preference to their own, (3) repeat or reinforce previous viewpoints. **(B) Low User Engagement** users who: (1) show little interest, (2) add shallow contributions, such as jokes or links. **(C) Negative Sentiment** relevant when disagreements are not resolved politely and respectfully. **(D) Rudeness** use of abusive, rude or impolite words.

¹Annotated dataset available at <https://gitlab.com/ayush-jain/collaborative-yahoo-discourse>

2.2 Collaborative Discourse Behaviors

(A) High Idea Development when users stay on topic (with respect to the original post) and new ideas are formed and developed based on preceding turns. **(B) Reference to Previous Posts** users refer to the previous post to advance the conversation. **(C) Back and Forth** users support and appreciate the ideas shared by others, and are polite when expressing disagreements. **(D) Positive Sentiment** resulting in positive interactions among users, expressed through polite conversation or informal emoticons. **(E) High User Engagement** leading to insightful discussions, meaningful to its participants. **(F) Balanced Content Distribution** between all members in the group. **(G) Questions** raised by participants to advance the conversation.

Annotation Process Two annotators labeled the conversations based on these guidelines, with an accuracy in inter-annotator agreement of 81%.

3 Modeling Collaborative Behaviors

Identifying collaborative conversations requires characterizing nuanced behaviors. In previous work, this analysis was defined by extracting social and discourse features directly from the raw data. In contrast, we view this decision as a probabilistic reasoning process over the relevant conversational behaviors that were identified during the annotation process (Sec. 2.1 and 2.2). Since these behaviors are not directly observed, and have to be inferred from the raw conversational features, we treat them as discrete latent variables which are assigned together-with, and consistent-with, the final classification task.

Each behavior is captured by a binary latent variable, denoted as $\mathbf{h} = \langle h_1, \dots, h_k \rangle$, indicating if it’s active or not in the given thread. These decisions are then connected with the final prediction, denoted y , a binary output value. This results in a factor graph (Figure 1). Each individual decision is scored by a neural net, and uses a set of features capturing relevant properties in the input conversation. To learn this model, we extend DRaiL (Zhang et al., 2016), a recently introduced framework for combining declarative inference with neural networks, described briefly in the following section. Our extension allows for the introduction of discrete latent predicates into the model.

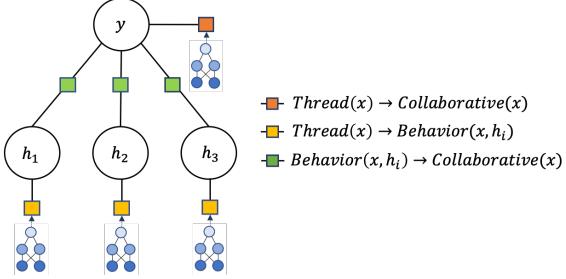


Figure 1: Factor Graph for Collaborative Conversations

3.1 Learning and Inference with DRaiL

DRaiL uses a first-order logic template language to define structured prediction problems. A task is defined by specifying a finite set of *predicates*, corresponding to observed or output information. Decisions are modeled using rule templates, formatted as horn clauses: $A \Rightarrow B$, where A (*body*) is a conjunction of observations and predicted values, and B (*head*) is the output variable to be predicted. The collection of rules represents the global decision, taking into account the dependencies between the rules using a set of constraints C . Rule instances are represented by variables r_i , and they are scored using neural nets, defined over a parameter set \mathbf{w} .

$$\mathbf{y} = \arg \max_{r_i} \sum_i r_i \cdot \text{score}(x, \mathbf{w}, r_i)$$

subject to $C, \quad \forall i; r_i \in \{0, 1\}$ (1)

We define two models using this representation. The first, **DRaiL Local**, trains a single neural net, represented by the rule: $\text{THREAD}(T) \Rightarrow \text{ISCOLLABORATIVE}(T)$, mapping the thread to the predicted value directly. The input layer to the neural net is the union of word indicators and all the features used to capture conversational behavior (Table 1). This approach is similar in spirit to previous works, classifying conversational threads using aggregated features.

The second, **DRaiL Global**, builds on the previous model, augmenting it with rules capturing individual discourse behaviors, and then associating the predictions of these rules with the final prediction task. We define the set of latent conversational behaviors $B \in \{\text{Idea Development, Reference to Previous Post, Sentiment, Balanced Content, Back and Forth, Questioning Activity, User Engagement, Rudeness and Controversial}\}$.

We define two rules for each behavior in B , as follows: $\text{THREAD}(T) \Rightarrow \text{LATENTBEHAVIOR}(T, B)$, corresponding to a neural net predicting the occurrence of the specific behavior

Behavior	Features
S	Degree of sentiment and intensity
B.C	Sentences per post, words per posts, post depth
C	Upvote/downvote ratio, $u - d, u + d, u/(u + d)$
R.P.P	2 per. pronouns, quotes of prev. posts, @username tags
B.F	(Dis)agreement markers, content indicators, post references
I.F	Lexical chains (Barzilay and Elhadad, 1997)
R	Profanity, bad words, short posts indicators
U.A	Number of posts, number of threads
Q.A	Question marks, question forms, question types

Table 1: Features per Behavior. Sentiment (S), Balanced Content (B.C), Controversial (C), Reference to Previous Posts (R.P.P) Back and Forth (B.F), Idea Flow (I.F), Rudeness (R), User Activity (U.A), Questioning activity (Q.A)

B in conversational thread T . We also add the rule: $\text{LATENTBEHAVIOR}(T, B) \Rightarrow \text{ISCOLLABORATIVE}(T)$, capturing the relationship between the latent behavior and the collaborative prediction.

Each rule template is associated with an initial feature representation and a neural architecture to learn its scoring function. After scoring factors, values are assigned to the output variables by running an inference procedure. DRaiL uses Integer Linear Programming (ILP) to solve the inference problem. In our setup, we compare two models, with and without inference, corresponding to the global and local models.

Global Learning When multiple rules are defined in DRaiL, each has its own neural architecture and parameters. Since these rules are interconnected, DRaiL learns a globally normalized model which uses inference to ensure that the scoring functions for all rules result in a globally consistent decision. We adapted the structured hinge loss used in DRaiL to handle latent predicates. The loss function is defined over all neural parameters \mathbf{w} , and the error is back-propagated to update all networks.

$$L_D(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{n} \sum_{i=1}^n \xi_i \quad (2)$$

Where ξ_i is the slack variable, capturing the margin violation penalty for a given training example, and defined as follows:

$$\begin{aligned} \xi_i &= \max_{y, \mathbf{h}} (f(\mathbf{x}_i, \mathbf{h}, y, \mathbf{w}) + \Delta(y, y_i)) \\ &\quad - \max_{\mathbf{h}} f(\mathbf{x}_i, \mathbf{h}, y_i, \mathbf{w}) \end{aligned}$$

Here, \mathbf{x}_i and y_i are the inputs and gold labels for the i -th instance and \mathbf{h} denotes the active DRaiL rules corresponding to latent discourse behaviors.

4 Empirical Evaluation

4.1 Dataset and Experimental Settings

We annotate conversations on the Yahoo News Annotated Comments Corpus (Napoles et al., 2017b) following the guidelines specified in section 2, with 81% inter-annotator accuracy. The dataset consists of 2130 conversations for training, 97 for validation and 100 for testing. The data is imbalanced, with more conversations being non-collaborative (64%, 69% and 67% for training, validation and testing, respectively). Additionally, we annotated the fine-grained discourse behaviors for a sample set of 103 conversations.

We used feedforward networks for all rules, with one hidden layer and a softmax on top. All hidden layers use sigmoid activations. The number of hidden units are: 400 for the local rule, 50 for idea flow and 100 for all remaining behaviors. Rules that map a latent behavior to a final decision did not have a hidden layer. We used a learning rate of 0.01. All of these parameters, as well as the weights for the different rules, were tuned using the validation set.

4.2 Experiments

We compare the model that explicitly reasons about conversational behaviors and their relationships (*DRaiL Global*), with a local model that predicts whether a conversation is collaborative or not by using all discourse features as inputs to a single rule (*DRaiL Local*). To motivate the use of neural networks, we include two *Linear SVM* baselines, using bag-of-words and the set of all discourse features (Table 1). These results (Table 2) demonstrate the advantage of modeling competing discourse behaviors as latent variables and making a joint decision using inference, as opposed to just representing them using input features.

Model	Prec.	Rec.	F1
Linear SVM(BoW)	0.60	0.58	0.59
Linear SVM(BoW + disc.)	0.63	0.61	0.62
DRaiL Local(single NN)	0.65	0.64	0.64
DRaiL Global (latent vars.)	0.69	0.68	0.69

Table 2: Predicting Collaborative Conversations (Fixed splits)

We conduct an additional experiment to evaluate the quality of the predicted latent behaviors. To do this, we annotated the discourse behaviors based on the definitions provided in section 2, and evaluate the activations produced by our global model. We

compare their correctness **before** learning (based on initialization parameters) and **after** global learning. Inference is used in both cases. Table 3 describes the results. We can see that performance consistently improved after global training compared to the initialization point, a clear indication of the connection between the latent information and the predicted conversational outcome. Identifying rude behaviors yields the highest F1 score (0.62), which can be expected as the decision relies on lexical information (negative and abusive words). Similarly, it is relatively easy to identify balanced content behavior, given that structural features (outlined in table 1) are very informative. Lexical chains, representing the repeated occurrence of a single word or of several closely related words over the course of a post (Barzilay and Elhadad, 1997), are also successful at capturing idea flow behaviors. However, controversial and back and forth behaviors are more challenging.

Individual Behavior	F1 (before)	F1 (after)
Idea Flow	0.371	0.574
Controversial	0.390	0.420
Balanced Content	0.541	0.610
Sentiment	0.462	0.548
User Activity	0.521	0.570
Reference to Previous Posts	0.299	0.427
Questioning Activity	0.427	0.511
Rudeness	0.514	0.620
Back and Forth	0.470	0.520

Table 3: Predicting Individual Latent Behaviors on Annotated Sample Set Before and After Global Learning

We performed an ablation study to see if the global model is driven by any particular discourse behavior (Table 4). We observe that performance drops significantly if the sentiment behavior is removed. Just using rules related to idea flow, sentiment and balanced content behaviors leads to an F1 score of 0.62.

Model	Precision	Recall	F1
All	0.690	0.680	0.687
All except S	0.483	0.495	0.489
All except I.F	0.635	0.554	0.591
All except B.C	0.581	0.593	0.586
All except Q.A	0.578	0.588	0.582
I.F + S + B.C	0.645	0.607	0.625
I.F + S + U.A	0.665	0.404	0.502
S + B.C + C + Q.A	0.693	0.546	0.610

Table 4: Ablation Study. Sentiment (S), Idea Flow (I.F), Balanced Content (B.C), Questioning Activity (Q.A), User Activity (U.A), Controversial (C)

5 Summary and Future Work

In this paper, we introduce the task of identifying collaborative conversations and provide annotations for a subset of the Yahoo News Annotated Comments Corpus. We suggest an approach that combines neural networks with constrained inference for identifying collaborative conversations, and showed how adding additional inductive bias in the form of discrete latent variables can improve learning. Moreover, we show that we are able to capture and explain individual discourse behaviors without additional supervision, which in turn allows us to gain insight into the final decision made by the model. Collaborative interactions help leverage the synergy between team members tackling complex problems, we hope to contribute in the development of automated systems supporting such processes.

References

- Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.
- Michael Flor, Su-Youn Yoon, Jiangang Hao, Lei Liu, and Alina von Davier. 2016. Automated classification of collaborative problem solving interactions in simulated science tasks. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 31–41.
- Samuel Greiff. 2012. From interactive to collaborative problem solving: Current issues in the programme for international student assessment. *Review of psychology*, 19(2):111–121.
- Jiangang Hao, Lei Liu, Alina von Davier, Patrick Kyllonen, and Christopher Kitchen. 2016. Collaborative problem solving skills versus collaboration outcomes: Findings from statistical analysis and data mining. In *Proceedings of the 9th International Conference on Educational Data Mining*.
- Dan Jurafsky, Rajesh Ranganath, and Dan McFarland. 2009. Extracting social meaning: Identifying interactional style in spoken conversation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 638–646. Association for Computational Linguistics.
- Courtney Napoles, Aasish Pappu, and Joel Tetreault. 2017a. Automatically identifying good conversations online (yes, they do exist!). In *Proceedings of the International AAAI Conference on Web and Social Media*.
- Courtney Napoles, Joel Tetreault, Enrica Rosata, Brian Provenzale, and Aasish Pappu. 2017b. Finding good conversations online: The yahoo news annotated comments corpus. In *Proceedings of The 11th Linguistic Annotation Workshop*, pages 13–23, Valencia, Spain. Association for Computational Linguistics.
- Vlad Niculae and Cristian Danescu-Niculescu-Mizil. 2016. Conversational markers of constructive discussions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–578, San Diego, California. Association for Computational Linguistics.
- Lu Wang and Claire Cardie. 2014. A piece of my mind: A sentiment analysis approach for online dispute detection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 693–699, Baltimore, Maryland. Association for Computational Linguistics.
- Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399. International World Wide Web Conferences Steering Committee.
- Justine Zhang, Jonathan Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Dario Taraborelli, and Nithum Thain. 2018. Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1350–1361. Association for Computational Linguistics.
- Xiao Zhang, Maria Leonor Pacheco, Chang Li, and Dan Goldwasser. 2016. Introducing DRAIL – a step towards declarative deep relational learning. In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 54–62, Austin, TX. Association for Computational Linguistics.

A Case Study of User Communication Styles with Customer Service Agents versus Intelligent Virtual Agents

Timothy Hewitt

Verint - Next IT

Spokane Valley, WA USA

timothy.hewitt@verint.com

Ian Beaver

Verint - Next IT

Spokane Valley, WA USA

ian.beaver@verint.com

Abstract

We investigate differences in user communication with live chat agents versus a commercial Intelligent Virtual Agent (IVA). This case study compares the two types of interactions in the same domain for the same company filling the same purposes. We compared 16,794 human-to-human conversations and 27,674 conversations with the IVA. Of those IVA conversations, 8,324 escalated to human live chat agents. We then investigated how human-to-human communication strategies change when users first communicate with an IVA in the same conversation thread. We measured quantity, quality, and diversity of language, and analyzed complexity using numerous features.

We find that while the complexity of language did not significantly change between modes, the quantity and some quality metrics did vary significantly. This fair comparison provides unique insight into how humans interact with commercial IVAs and how IVA and chatbot designers might better curate training data when automating customer service tasks.

1 Introduction

An intelligent virtual agent (IVA) is a subset of chatbots designed for the commercial enterprise realm to mimic a human customer service agent. A popular use case for IVAs is live chat deflection, where they are trained to handle the most common interactions while still allowing for escalation to a human agent when required or requested.

As a company that has designed and built IVAs for enterprise applications for many years, we had intuition that the language we saw in live chat interaction was different from the language we saw coming into the IVA, but the difference had not yet been quantified. After using live chat data for training an IVA, we were occasionally surprised at the gaps in understanding it presented once in production, even though the training data

originated from the same company the IVA was built for. In order to improve analysis and training, we sought a standard to create and gather data more consistent with actual IVA usage and filter out some of the non-representative live chat data.

We also wanted to investigate how the IVA was affecting conversations with live chat operators. While there are differences, a user behaves consistently when he/she is chatting with a human, similarly they are consistent when chatting with an IVA. In this paper we demonstrate that chatting with an IVA has significant impact on language beyond what has been documented by human-to-human computer mediated conversation such as instant messenger or live chat.

The IVA and live chat corpora used in this study originated from a financial services company where customers are interacting with the IVA and live chat on their website for the same purposes making the comparisons extremely relevant. However, due to data use agreements with the financial services company, the identification of the origin and corpora cannot be made public.

2 Related Works

Hill et al. (2015) have done comparisons between inter-human and “toy” chatbot conversations. However, in this comparison the conversations were sampled from completely unrelated domains making the comparison less valid.

While tools to improve the training process of IVAs from live chat or call center transcriptions exist such as (Bouraoui et al., 2019), there has not been a focused linguistic study on the difference in communication styles between human-human and human-machine in service dialogs. Such a study could inform such tools where specific samples may or may not make good training samples due to projected communication differences with IVAs. To our knowledge this is the first study to compare real world language of users with IVAs and live

chat from the same origin.

3 Method

The IVA for this research was originally trained on live chat conversations from the financial sector and continuously refined while in production. It was designed to understand frequently asked questions and conversational work flows around the largest business use case: waiving fees (for example conversations see Appendix B). Besides business intents, the IVA also responds to persona (e.g. asking if the IVA is married), common courtesy, and profanity. Escalation points were designed where human involvement was desired (e.g. credit limit changes, account closure). There was no dynamic response delay, no avatar, and users were informed at the beginning of the conversation that they were speaking with an IVA (see Appendix A).

For our corpus, we selected 16,794 conversations with live chat agents from June through October 2017 and 27,674 conversations with the IVA that occurred in January 2020. Within the IVA conversations, 19,350 conversations were completed with IVA only while 8,324 escalated at some point to a live chat agent.

For the purpose of this work we only looked at the user language and actions and not the IVA or live agent responses. The IVA was launched in 2017 on the company website along side a live chat option. After 2017, access to live chat without first talking to the IVA was disabled due to the success of the IVA at automating a continuously expanding set of use cases. We chose to sample IVA data from 2020 to allow for adequate refinement time to present statistics representative of communicating with a mature IVA implementation.

3.1 Conversational Clicks

When we discuss turn-taking in conversation with a multi-modal IVA, we must consider that there are different methods than typing to elicit more information. For instance, clicking on suggested topic or answer links presented by the IVA will continue the conversation as though the user had typed the text of the link. In our domain, specific actions need to occur if a credit card is stolen. If a user goes to either an IVA or a live chat operator and says, “I need to get a replacement card,” the operator might respond with a “Was the card stolen?” whereas the IVA might present two conversational clicks, <Replace a lost or damaged

card> <Replace a stolen card>. There were a few considerations for counting these interactions in respect to word counts and user turns.

Remove conversational clicks as a word level metric. This metric allows for direct comparison of the complexity of typed user inputs, but hinders the ability to compare at a conversational level. Both IVA and live chat operators can ask a yes or no questions, but if we drop the click of a “yes” response link to the IVA we lose the comparison to the “yes” response in live chat.

Count clicks as a one word turn. In our example, if we assume a conversational click would only solicit a single piece of information a single word turn would be a fair metric. However, conversational clicks are not always of this type. Some present additional information (such as what to do if a stolen card is found) or other suggested topics (such as upsell opportunities).

Count the language in the link text as the user input. In our example, the same information is required, but the method of eliciting that information has changed the user’s interaction from a single word typed input, “yes,” to a four word conversational click.

For any of these metrics, the count would not be representative of the language a user might input if the conversational click was not present.

For all options considered, there were sufficient concerns that any metrics provided on this data set would be implementation dependent, so we chose to present the statistics for all three options outlined so the reader can understand where the differences lie and to what extent noise exists within our IVA data set from conversational link clicks. To control for question complexity between environments, we measured the frequency of yes/no questions and found that they occurred 8% more often in live chat than IVA conversations.

3.2 Turns

For the purposes of this study, if the user clicks on a suggestion by the IVA that advances the conversation (that is, it returns a response in the IVA), it will count as a turn. IVA turns are ABAB, that is, the user (A) takes a turn and the IVA (B) follows. Live chat turns can extend over multiple inputs, such as, ABAAAB. In such cases, these will be joined into a single turn. In other words, we will treat ABAAAB as ABAB.

	Live Chat	IVA Only	Mixed Sessions	Mixed - IVA	Mixed - Live Chat
User Words/Session	68.83 (61.90)	27.91 (22.10)	114.88 (84.67)	32.45 (23.50)	82.51 (80.31)
Words/Session (links = 0 words)	n/a	23.70 (20.97)	107.61 (84.50)	25.16 (22.34)	n/a
Words/Session (links = 1 word)	n/a	24.72 (21.16)	109.15 (84.52)	26.71 (22.58)	n/a
User Turns/Session	5.12 (3.81)	3.06 (2.16)	10.65 (5.56)	4.03 (2.04)	6.62 (2.04)
Type/Token	0.77 (0.11)	0.82 (0.07)	0.79 (0.07)	0.81 (0.06)	0.78 (0.10)

Table 1: Means and standard deviation of session level analysis. Words/Session is raw words including link click text, links = 0 ignores link clicks, and links = 1 treats link clicks as single word inputs. Type/Token is the ratio of unique words over all words in the session.

3.3 Sentences

Successful conversation over chat does not require full, grammatically complete sentences and IVAs are frequently used as keyword searches. Sentence boundaries and punctuation are many times missing or grammatically misused. As such, we ignore sentence-level metrics.

3.4 Metrics

There are 3 session types: Live Chat (human to human conversation), IVA Only (human to IVA conversation) and Mixed Session (sessions that started with the IVA and escalated to a human live chat operator). A mixed session has two subtypes: Mixed - IVA (user inputs to the IVA in a Mixed Session) and Mixed - Live Chat (user inputs to the human live chat operator in a Mixed Session).

We used the L2 Syntactic Complexity Analyzer (L2SCA) (Lu, 2010) to measure complexity. However, we will not be using any of L2SCA’s sentence based metrics for the reasons discussed in 3.3. We also ran the user turns through our own measures for quality and quantity.

For quality, we selected some of the variables selected by Lortie and Guitton (2011) and Hill et al. (2015) from LIWC (Pennebaker et al., 2015) and included a metric for politeness. However, we did not use LIWC due to data security policies. For fair comparison, we used word lists from closed class words and opted out of the more subjective open class word based features, other than profanity. The variables of quality we investigated were misspellings, words with more than 6 characters, pronouns, articles, profanity, and politeness.

Misspellings compared tokens against a list of company products and services first, and, if the token was not found there, it was then spell-checked against the English aspell wordlist¹.

¹<https://ftp.gnu.org/gnu/aspell/dict/en>

Gratitude is a count of the variations of *thank* in a turn. We considered only expressions of gratitude as politeness for this study to reduce potential classification error from approaches such as (Yildirim et al., 2005).

Profanity was checked using a regex of common swear word phrases. There is substantial variation in how people manage to misspell a profane word. The regular expressions are not exhaustive, but broad enough to ensure a quality sample.

Tokens are counted by splitting on white space. Thus punctuation won’t count as unique tokens and contractions will only count as a single token.

Type/Token is the ratio of unique words over all words in a turn or session.

Sentiment was measured using the NLTK implementation of VADER (Hutto and Gilbert, 2014) and is normalized from -1 to 1.

4 Analysis

Conversation Level: We begin with the full conversation level metrics shown in Table 1. Each conversation which escalated to live chat involves a link click where the link text was 4 words. This extra click is included in the IVA session.

Live chat conversations take 1.7 times more turns with more than 2.5 times more words. Where escalation is not required, a user can achieve a more efficient resolution with the IVA. However, if the IVA is in fact deflecting the easier to handle issues this could explain some of the differences.

On the other hand, the user experience for escalation is substantially less efficient. First the user has an average length IVA conversation and then escalates to the human agent for a more involved conversation with an average of 1.5 more turns and 14 more words than the live chat sessions alone. This indicates the user’s tasks presented to the IVA are not being properly reviewed by the live chat agents, requiring substantial additional effort

	Live Chat	IVA Only	Mixed - IVA	Mixed - Live Chat
Tokens	14.33 (14.29)	9.11 (8.00)	8.05 (7.03)	12.54 (13.44)
Tokens (links = 0 words)	n/a	7.74 (9.00)	6.24 (8.16)	n/a
Tokens (links = 1 word)	n/a	8.07 (8.72)	6.62 (7.87)	n/a
Type/Token	0.79 (0.13)	0.80 (0.12)	0.78 (0.12)	0.75 (0.17)
Misspellings	0.61 (1.16)	0.18 (0.53)	0.13 (0.44)	0.58 (1.06)
Six Character Words	3.08 (3.69)	2.45 (2.29)	2.21 (2.20)	2.44 (3.51)
Profanity	0.00 (0.02)	0 (0)	0 (0)	0.00 (0.02)
Gratitude	0.20 (0.41)	0.05 (0.22)	0.01 (0.09)	0.11 (0.35)
Sentiment	0.20 (0.33)	0.08 (0.27)	0.06 (0.24)	0.20 (0.30)

Table 2: Means and standard deviation of language quality metrics per turn. Tokens includes link click text, links = 0 ignores link clicks, and links = 1 treats link clicks as single word inputs. Type/Token is the ratio of unique words over all words in a turn.

on the part of the user to restate them.

Turn Level: Table 2 gives the turn level metrics. Users type substantially shorter inputs (between 1.5 and 1.8 times) when speaking with an IVA. It appears that beside being more concise with IVA, users are also more careful as there were 4.2% of tokens misspelled in live chat vs 2.0% when interacting with the IVA only and 1.6% when interacting with the IVA prior to speaking with a live chat agent. After communicating with an IVA, users increased to a 4.6% misspelling rate.

Human-to-human gratitude is significantly more frequent than with an IVA. However, after continuing to a human after the IVA, gratitude is almost halved. This reflects the more difficult conversations when live chat is tier 2 support.

Sentiment for human-to-human was significantly more positive. IVA turns were neutral. IVA-only turns averaged 0.08 where as live chat conversation turns averaged 0.2. One would expect the live chat conversations that were preceded by the IVA to be more negative reflecting the decrease in user efficiency discussed in the previous section. However, sentiment for live chat after IVA actually remained at 0.2, perhaps indicating that live chat was usually leading to a reasonable (if not always satisfactory) resolution or the additional effort seemed justified to the users as they were in a sense restarting the conversation with a new party.

Hill et al. (2015) showed significant profanity in chatbot language and Burton and Gaskin (2019) showed a self-reported tendency to be less polite to digital assistants. In our data, only live chat sessions had any profanity to speak of. We speculate that the overall lack of profanity has to do with the professional setting of the customer service environment where previous studies were on open domain chatbots and personal assistants such as Amazon’s Alexa.

Pronouns: Live Chat users were almost 2.9 times more likely to refer to the human as ‘you’ than they were with an IVA (Table 3). When a user escalates to live chat, the pronouns increase, but in general pronoun use is less in conversations that escalated. This implies that when a user knows that they aren’t chatting with a human, they remove any references to it as a person, consistent with Burton and Gaskin (2019).

L2SCA returned results that could be explained by shorter turns and fewer words shown between live chat and IVA (Table 3). However, there were two increases worth mentioning in IVA-only conversations. Complex nominals per T-unit (CN/T) increased in IVA usage from a mean of 0.64 to 0.70. The other is mean length of clauses which increased from 5.34 to 5.78. Given the decrease in T-units and Clause/T-unit, this may indicate a tendency of IVA users to rely on conveying information through noun phrases than complete verb phrases. However, these increases were not reflected in users who escalated to Live Chat, the reason for this is unclear.

L2SCA did show that live chat language after IVA was less complex across every measure. This may be part of the explanation for the reduction in gratitude in those conversations: they were less polite because they were more concise. It may be that as the conversation is less efficient, the language becomes more efficient to compensate, but more research is needed to prove this hypothesis.

5 Application and Conclusion

When designing an IVA and when given live chat data for training, it’s tempting to tag random inputs indiscriminately for training. However, indiscriminately adding longer inputs more

	Live Chat	IVA Only	Mixed - IVA	Mixed - Live Chat
Pronouns	1.92 (2.22)	1.60 (1.60)	1.33 (1.55)	1.46 (2.06)
1st Person	1.31 (1.77)	1.28 (1.35)	1.09 (1.30)	0.88 (1.53)
2nd Person	0.31 (0.59)	0.13 (0.37)	0.08 (0.30)	0.30 (0.65)
3rd Person	0.30 (0.59)	0.19 (0.49)	0.16 (0.45)	0.28 (0.72)
Articles	0.73 (1.19)	0.58 (0.88)	0.50 (0.82)	0.64 (1.20)
Verb Phrase (VP)	2.37 (2.54)	1.93 (1.70)	1.67 (1.58)	1.96 (2.26)
Clause (C)	1.92 (1.97)	1.52 (1.30)	1.34 (1.21)	1.60 (1.74)
T-Unit (T)	1.26 (1.13)	1.10 (0.79)	0.99 (0.75)	1.09 (1.02)
Dependent Clause (DC)	0.62 (1.12)	0.40 (0.75)	0.32 (0.68)	4.03 (2.04)
Complex T-Unit (CT)	0.40 (0.70)	0.29 (0.53)	0.23 (0.49)	0.31 (0.61)
Coordinate Phrase (CP)	0.17 (0.46)	0.13 (0.37)	0.10 (0.33)	0.12 (0.42)
Complex Nominal (CN)	1.14 (1.63)	0.88 (1.12)	0.74 (1.03)	0.93 (1.42)
Mean Length of T	8.13 (7.94)	7.90 (6.40)	6.95 (6.26)	7.42 (7.65)
Mean Length of C	5.34 (4.10)	5.78 (3.97)	5.20 (3.96)	5.10 (4.15)
VP/T	1.50 (1.43)	1.46 (1.20)	1.32 (1.14)	1.38 (1.39)
C/T	1.22 (1.10)	1.15 (0.88)	1.05 (0.84)	1.13 (1.05)
DC/C	0.18 (0.27)	0.14 (0.25)	0.12 (0.24)	0.16 (0.27)
DC/T	0.40 (0.78)	0.29 (0.61)	0.24 (0.57)	0.34 (0.74)
CT/T	0.24 (0.39)	0.21 (0.39)	0.18 (0.34)	0.21 (0.39)
CP/T	0.10 (0.31)	0.09 (0.29)	0.08 (0.27)	0.09 (0.29)
CP/C	0.07 (0.22)	0.06 (0.22)	0.05 (0.20)	0.06 (0.21)
CN/T	0.70 (1.06)	0.64 (0.90)	0.55 (0.84)	0.63 (1.00)
CN/C	0.41 (0.54)	0.43 (0.56)	0.38 (0.55)	0.39 (0.55)

Table 3: Means and standard deviation of pronoun and article usage and the results of L2SCA per turn.

common in live chat may introduce unnecessary noise to the data. Given our observations, we recommend that training language be more focused to the task and rely on more direct language. We also recommend designers do not neglect to add training samples in the form of keyword searches for the users who still view the IVA as a search tool.

When live chat data is not available, a synthetic strategy must take place. One such strategy outlined by Leuski et al. (2006) is to give a human a sample input and ask them to synthesize new data. A better plan would be to give the user a task of retrieving information and then asking them what questions they would use to get that information. This would encourage the simple direct language that IVAs are more likely to see in the wild instead of forcing the human to be creative, which may result in language unlikely to be seen by an IVA.

Users are currently more likely to be concise with IVAs than human live chat operators in the same domain. This is an advantage to the user as an IVA can respond more quickly and get them the needed information with less language production.

Modern contact centers use various performance metrics to rate contact center agents, which has a direct impact on their compensation and recognition (Cheong et al., 2008). A successful IVA will significantly reduce the number of conversations coming into a contact center, and the

conversations that do will largely consist of more difficult cases. As these types of conversations now make up a much larger part of the performance metrics, and the operators no longer get the positive feedback from the easy cases, they appear less effective than before the IVA was implemented. This decrease in performance should be expected and these metrics adjusted, perhaps by weighting by the difficulty of the task, so that human contact center agents are not punished by the deployment of IVAs alongside them.

There exists substantial research regarding how to measure the performance of a dialog system, but the study of how people communicate through language with artificial intelligence in the wild is still in its infancy. The nature of the data originating from commercial IVAs means that corpora are seldom shared, making the research more challenging. In spite of these sharing restrictions, in this paper we have presented an approach to analyze the nature of language use between humans and IVAs compared to that of human chat operators in a way that still allows the research community to understand in what way humans currently communicate differently with IVAs than other humans in the same domain. If others with commercially deployed IVAs repeat such experiments we can observe how humans adapt to IVAs over time in the wild, and change the way IVA conversations are designed accordingly.

References

- Jean-Léon Bouraoui, Sonia Le Meitour, Romain Carbou, Lina M Rojas Barahona, and Vincent Lemaire. 2019. Graph2bots, unsupervised assistance for designing chatbots. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 114–117.
- Nathan G Burton and James Gaskin. 2019. “Thank you, Siri”: Politeness and intelligent digital assistants. In *AMCIS 2019*.
- K Cheong, J Kim, and S So. 2008. A study of strategic call center management: relationship between key performance indicators and customer satisfaction. *European Journal of Social Sciences*, 6(2):268–276.
- Leon Ciechanowski, Aleksandra Przegalinska, Mikolaj Magnuski, and Peter Gloor. 2019. In the shades of the uncanny valley: An experimental study of human–chatbot interaction. *Future Generation Computer Systems*, 92:539–548.
- Ulrich Gnewuch, Stefan Morana, Marc TP Adam, and Alexander Maedche. 2018. Faster is not always better: Understanding the effect of dynamic response delays in human–chatbot interaction. In *26th European Conference on Information Systems: Beyond Digitization-Facets of Socio-Technical Change, ECIS 2018*, page 143975.
- Jennifer Hill, W Randolph Ford, and Ingrid G Farreras. 2015. Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. *Computers in human behavior*, 49:245–250.
- Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*.
- Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006. [Building effective question answering characters](#). In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 18–27, Sydney, Australia. Association for Computational Linguistics.
- Catherine L Lortie and Matthieu J Guitton. 2011. Judgment of the humanness of an interlocutor is in the eye of the beholder. *PLoS One*, 6(9).
- Xiaofei Lu. 2010. Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15(4):474–496.
- James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. The development and psychometric properties of liwc2015. Technical report.
- Serdar Yildirim, Chul Min Lee, Sungbok Lee, Alexandros Potamianos, and Shrikanth Narayanan. 2005. Detecting politeness and frustration state of a child in a conversational computer game. In *Ninth European Conference on Speech Communication and Technology*.

A Implementation Considerations

Some research on human-chatbot interactions suggests that dynamic response times increased overall satisfaction with the chatbot (Gnewuch et al., 2018). Ciechanowski et al. (2019) recommended avoiding avatars or otherwise making the chatbot appear human visually. It is unknown how these, or any other measures, will impact user language.

B Example Dialogs

User: hi i did not mean to make a late payment i was in the hospital, can you please waive the charge for me :)

IVA integration checks against user account and eligibility for fee waiver

IVA: Your account is eligible for a fee waiver. Customers can have this fee waived when <list of business conditions>. Are you sure you want to apply this fee waiver now?

Yes (link)

No (link)

User: yes please

IVA: The fee has been waived. Can I help you with anything else?

User closes window

Live Chat Agent: Hello <user>. How can I help you today

User: Hi. My bill came in and I saw a fee.

User: What is that all about?

Live Chat Agent: Let me take a look at that.

Live Chat Agent: It looks like this fee was applied to your account for <business reasons>.

User: I didn't know. Is there any way you can waive the fee this time?

User: That never has happened before.

Live Chat Agent: It looks like your account is eligible for a one time waiver. You will not be eligible again until <business requirements>. Would you like to apply that now?

User: Yes. Please.

Live Chat Agent: Ok. It's done. Is there anything else I can do for you?

User: No Thanks.

User: Bye.

Live Chat Agent: Thank you for contacting <company>. Have a great day.

User: thanks bye.

User: waive the fee

IVA integration checks against user account and eligibility for fee waiver

IVA: Your account is eligible for a fee waiver. Customers can have this fee waived when <list of business conditions>. Are you sure you want to apply this fee waiver now?

Yes (link)

No (link)

User: clicks link "Yes"

IVA processes fee waiver and returns with confirmed status

IVA: The fee has been waived. Can I help you with anything else?

User closes window

Figure 2: An example conversation showing the user advancing the conversation through conversational link clicks.

User: waive the fee

IVA integration checks against user account and eligibility for fee waiver

IVA: Your account is not eligible for a fee waiver. Customers can have this fee waived when <list of business conditions>.

User: Is there any leniency for a one time exception

IVA: Exceptions can be made in the cases of <list of business conditions>.

IVA integration checks if a live chat operator is available

IVA: In order to do this, we need to connect you to a live chat operator. Wait time is currently 2 minutes.

Connect to Live Chat (link)

User: clicks link "Connect to Live Chat"

IVA hands off conversation to live chat operator

Operator: Hello <user>, I see you are asking to waive a fee on your account?

User: Yes. Can you do that for me?

Operator: Give me just a second while I look up your account

Operator: I'm sorry, it looks like your account is not currently eligible for a fee waiver right now.

User: The thing before told me you could make an exception for <business condition>.

User: Can you do that for me?

Conversation continues with live chat operator

Figure 1: An example conversation with an IVA (top) and an example with a human live chat operator (bottom) completing the same task of waiving a fee. Company-specific information has been sanitized.

Figure 3: An example conversation showing integration points for waiving a fee and escalation to live chat, as well as the use of conversational links of more than one word. See Section 3.1 for a discussion on the various ways to count such click interactions.

It's About Time: Turn-Entry Timing For Situated Human-Robot Dialogue

Felix Gervits, Ravenna Thielstrom, Antonio Roque, Matthias Scheutz

Human-Robot Interaction Laboratory

Tufts University, Medford MA 02155

{felix.gervits, ravenna.thielstrom,
antonio.roque, matthias.scheutz}@tufts.edu

Abstract

Turn-entry timing is an important requirement for conversation, and one that few spoken dialogue systems consider. In this paper we introduce a computational framework, based on work from psycholinguistics, which is aimed at achieving proper turn-entry timing for situated agents. Our approach involves incremental processing and lexical prediction of the turn in progress, which allows a situated dialogue agent to start its turn and initiate actions earlier than would otherwise be possible. We evaluate the framework by integrating it within a cognitive robotic architecture and testing performance on a corpus of situated, task-oriented human-robot directives. We demonstrate that: 1) the system is superior to a non-incremental system in terms of faster responses, reduced gap between turns, and the ability to perform actions early, 2) the system can time its turn to come in immediately at a turn transition, or earlier to produce several types of overlap, and 3) the system is robust to various forms of disfluency in the input. Overall, this domain-independent framework can be integrated into existing dialogue systems to improve responsiveness, and is another step toward more natural and fluid turn-taking behavior.

1 Introduction

Behavioral evidence shows that humans are able to exchange turns extremely quickly in conversation – within a few hundred milliseconds on average (Levinson and Torreira, 2015). This is a universal human characteristic, though the nature of the timings varies slightly across languages (Stivers et al., 2009). There is some debate about exactly how humans achieve this performance, but evidence from psycholinguistic studies suggests that it is likely done by processing ongoing utterances incrementally and making lexicosyntactic predictions about the turn in progress (de Ruiter et al., 2006; Magyari and de Ruiter, 2012). This allows a listener to

plan what to say and to anticipate the end of the speaker’s turn accurately so that turn-transitions are seamless, and gaps between turns are minimized. It also allows for the production of speech overlap, to produce conversational behaviors such as backchanneling and repair. Such human behaviors are desirable for spoken dialogue systems (SDSs) where *naturalness* is a priority (Edlund et al., 2008).

SDS research has produced an impressive body of work on turn-taking (e.g. Bohus and Horvitz (2011); Kronlid (2006); Raux and Eskenazi (2009, 2012); Skantze and Schlangen (2009); Zhao et al. (2015)), and some early work on overlap and completions (Baumann and Schlangen, 2011; DeVault et al., 2011b; Gervits and Scheutz, 2018a). However, relatively little focus has been placed on using turn-taking capabilities for **responsive turn-entry timing**, especially for situated agents. One exception is the approach by Baumann and Schlangen (2011) which involves estimating word duration to produce collaborative completions.

We build on this prior work through the development of a framework for achieving responsive turn-entry timing, as well as a full set of adaptive human-like overlap and completion behaviors. Our approach involves using utterance-level predictions from partial input and information from a world modeler to determine *when* to enter the turn (including producing overlap at any of the entry points shown in Figure 1), and whether to initiate actions early. Such capabilities are particularly important for situated dialogue agents, as responses, and especially actions, often involve lengthy processing delays, which can be mitigated by preparing or initiating them during an ongoing turn. Section 2 describes how this framework builds on existing research, including our novel *Turn-Entry Manager* (TEM) described in Section 2.4. In Section 3 we describe implementation details related to integrating

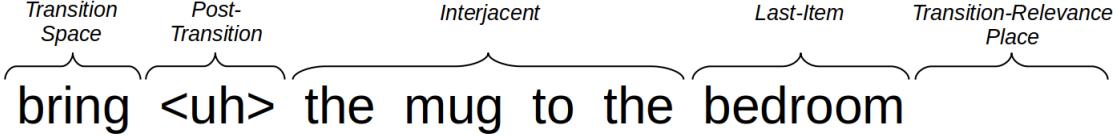


Figure 1: Entry points in a sample utterance based on (Jefferson, 2004). A fluid turn exchange starts at the transition point between turns (transition-relevance place or TRP), whereas earlier entry points indicate various types of overlap.

the framework in a cognitive robotic architecture. Then in Section 4 we evaluate our implementation on a corpus of situated human-robot dialogue utterances. Finally, we close with a discussion of the contributions and directions for future work.

2 A Framework for Turn-Entry Timing

Here we discuss the framework needed to manage turn-entry timing for situated dialogue agents, and the related work that the framework builds on.

2.1 Incremental Processing with Prediction

Obtaining an early understanding of the meaning of an utterance allows for faster feedback, supportive overlap, and faster actions. To achieve this, the SDS needs prediction, which is enabled by incremental processing.

Extensive prior work has supported fast and effective incremental processing with prediction (Paetzel et al., 2015; Skantze, 2017). For example, Schlangen and Skantze (2011) developed the Incremental Unit (IU) framework which supports incrementality with prediction, revision, and management of alternative hypotheses. This and other related approaches (e.g., Heintze et al. (2010); Skantze and Schlangen (2009)) involve interpreting meaning from each partial input rather than trying to predict the complete utterance. Other work has focused on predicting a full utterance (or semantic frame) from partial input using a maximum entropy classification approach (DeVault et al., 2011a; Sagae et al., 2009). These approaches attempt to find the point of maximal understanding at which a response can be initiated, and have been demonstrated to support the production of collaborative completions (DeVault et al., 2009). While these approaches use lexical cues in the input to generate predictions, other cues can also be used for situated interaction, including gesture and gaze (Kennington et al., 2013), and acoustic features (Maier et al., 2017; Ward et al., 2010). Our approach builds on this prior work in incremental processing, using it

as a component in our overall framework.

2.2 Speech Overlap Production

Speech overlap has been shown to serve many useful functions in conversation, including responsiveness and repair (Jefferson, 2004), but historically the SDS community has viewed it as an intrusive property and used the term *barge-in*. Some SDS work exists on the topic of intentional overlap production, including a body of work aimed at producing appropriate backchannel feedback (Lala et al., 2017; Truong et al., 2010). Another example comes from DeVault et al. (2011b), who designed a prototype system using predictive capabilities to perform collaborative completions and backchannel feedback. This work provides a necessary first step, but it only covers a subset of the different types of overlap possible, leaving out those that occur at the transition space, post-transition, and interjacent positions (Drew, 2009). Moreover, this work does not deal with situated dialogue or issues of timing in speech synthesis. Situated dialogue presents additional opportunities for overlap which have yet to be explored, such as coming in mid-utterance to clarify an un-actionable command. Finally, if a system will be producing overlap, then mechanisms to manage and recover from overlap are also needed. A preliminary approach was demonstrated in Gervits and Scheutz (2018a) based on a corpus analysis in Gervits and Scheutz (2018b), but otherwise there is limited work in this area.

2.3 Preemptive Action Execution

For dialogue in real-world or virtual environments with humans, situated agents can use predictive language capabilities to perform actions early or at least begin some processing during an ongoing utterance. This has been explored by Hough and Schlangen (2016), who developed a real-time incremental grounding framework that supports “fluid” interaction using the IU framework. While the system performance is impressive, this work only

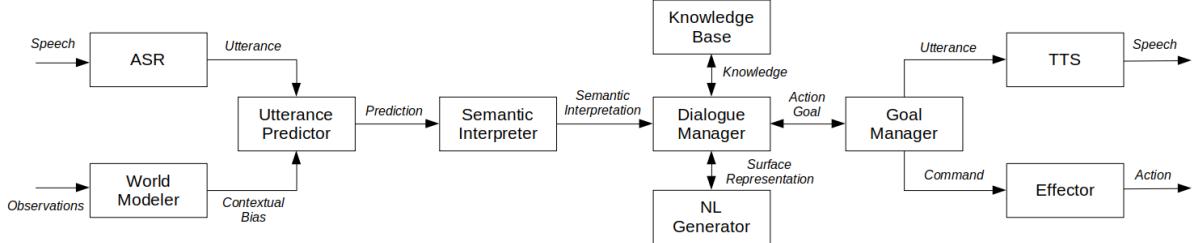


Figure 2: Component diagram of the turn-taking framework as implemented in the DIARC cognitive robotic architecture. Boxes represent architectural components and arrows represent the flow of information.

focused on action and did not involve timing dialogue responses. Moreover, the only actions considered were those that the robot *could* carry out. In human-robot interaction, a robot might be instructed to perform an action that it does not know how to do or that it cannot currently do. In order to respond early, the robot will need to simulate the action to determine if it will be successful. This simulation may involve a cognitive architecture carrying out an actual “mental” simulation of the action, or simply checking if the preconditions for the action are met. This is the type of processing that can be done during an ongoing utterance.

2.4 Turn-Entry Manager

Given the multitude of points for which a system may need to enter a turn (as shown in Figure 1), some process is needed to manage turn entry. We propose a *Turn-Entry Manager* (TEM) component that carries out these tasks. The TEM works as follows: it receives full utterance predictions from partial automatic speech recognition (ASR) results and determines *when* to initiate a follow-up utterance and action based on the confidence in the prediction as well as task context and agent goals. The most intuitive location for the TEM is in the Dialogue Manager (DM), as it uses information only available further along in the pipeline. The TEM will store the following information about its *prediction* of an ongoing utterance: semantics and text of the utterance, remaining words and expected duration of the utterance, response and action associated with the utterance, confidence in the prediction, cost of the action, entry time for the *transition-relevance place* (TRP) (Sacks et al., 1974) and several overlap positions, and latency of various components. Most of this information is updated with each increment received by the parser. Using this information, the TEM determines the timing of when to take a turn so as to achieve fluid turn transitions. Depending on its policy, it can also

come in early to produce various kinds of overlap. While most SDSs have some process that manages turn entry, none, to our knowledge, possess the capabilities described here.

3 Implementation in a Cognitive Robotic Architecture

To effectively interact in a situated environment, robots need to react to and affect the environment, as well as to reason about the task and user; this requires a *cognitive robotic architecture*. We integrated our turn-entry timing framework into the DIARC architecture (Scheutz et al., 2019). We used DIARC due to its emphasis on situated robot dialogue (highlighted in Gervits et al. (2020)), although in principle our framework is general enough to be used with any architecture of its type. Below we discuss each of the key components in our architectural configuration.

3.1 Situated Natural Language Processing

Our work is mostly performed in the language-processing components of DIARC, shown in Figure 2. First, speech is received by the *ASR component*, which converts it into text. For ASR, we use the Sphinx4 recognizer, modified to output incremental results. A text interface can also be used to simulate incremental speech input. The word-by-word results are sent to the *Utterance Prediction component* (described further in Section 3.2), which generates a prediction using a bigram language model and sends the prediction to the *Semantic Interpreter component*. We use a rule-based parser that performs syntactic and semantic parsing, and converts the text of an utterance into a logical predicate form. The predicate is then sent to the *DM component*, which is a goal-based dialogue manager that uses a Prolog knowledge base for storing declarative knowledge, and for performing logical inference over that knowledge to engage

in mixed-initiative dialogue. The DM implements a version of the *TEM* described in Sec. 2.4. The DM also interacts with the *Goal Manager (GM)* component, which contains a database of actions that the robot can perform (including dialogue actions) and facilitates action execution. Actions in DIARC are defined by their pre-, post-, and operating conditions. The post-conditions of an action are goal predicates that describe a state of the world that an agent is trying to achieve, e.g., *did(self,moveTo(self,bookshelf))* for an action goal, and *did(self,speak(okay))* for a dialogue goal. For dialogue actions, the DM obtains the surface form of the response utterance from the *NL Generator* component, and then submits the goal associated with the action to the GM. The GM then calls the *text-to-speech (TTS) component* (which is a wrapper for MaryTTs) to produce speech output. Physical actions are handled in a similar way, except that the *Effector* component corresponding to the action handles the execution.

3.2 Utterance Prediction with Contextual Bias

For utterance prediction, we implemented a bigram language model trained on the frequency distribution of bigrams in the HuRIC corpus (see Sec. 4). More sophisticated prediction algorithms are possible, but given the importance of speed, we chose a simple and effective approach. The prediction is computed as follows: given an initial word as input, the model generates a set of complete utterances based on the most probable follow-up words along with their associated probability. A cumulative probability threshold is used to determine when a prediction is sufficient, at which point the full utterance prediction with the highest probability is sent to the parser. If the threshold is not reached, then the algorithm waits for the next input word and repeats the same process.

A contextual bias is included to represent the influence of the situated environment as observed by the robot and included in a world model. This context influences the utterance predictor by increasing the probability of specific bigrams by a set amount, causing the model to favor those words. In our preliminary implementation, the context is hand-tuned for each utterance in the corpus¹, but situated agents would be able to determine this con-

¹For example, the context for the utterance “Grab the bottle on the kitchen table” may be ‘kitchen’ (describing the environment) and ‘bottle’ (describing an item in the environment)

text by perceiving the environment, through task knowledge, or through the dialogue history.

3.3 An Algorithm for Turn-Entry Management

The *TEM* algorithm works as follows (see Algorithm 1): First, an utterance is received incrementally from the ASR component. In parallel, each word is sent to the Utterance Predictor component, where the bigram language model described in Sec. 3.2 generates predictions based on the frequency distribution of the training corpus and any contextual bias (Algorithm 1, line 3).

If a prediction clears a set threshold, then it is sent to the DM component. The DM first computes a score for the prediction based on the cost of the associated action and the confidence in the probability (line 5). If the score is above a set threshold then it continues. The score threshold can be set to minimize early execution for costly actions (e.g., actions that can cause delay to repair, such as movement) in the case of a wrong prediction. If the score threshold is exceeded, the DM next computes the TRP and last-item entry points based on the utterance start time and expected duration, accounting for the known TTS delay, which was about 40 ms in our system (lines 7-8).

Next, the preconditions for the action associated with the predicted utterance are checked (line 9). If the action exists and the preconditions are met, then a response is set (but not yet generated; line 13); otherwise, a failure explanation is generated and immediately produced (line 11). In the case that the preconditions are met, the DM sets the overlap type (TRP, last-item, or collaborative completion) based on a simple policy (line 14).² The action corresponding to the prediction is then performed (line 15). Finally, once the overlap type is set, a separate thread running every 1 ms waits until the current system time reaches the designated entry point and then produces the associated response (lines 22-26).

4 Evaluation

To evaluate the efficacy of our framework, we used a corpus of directives to a household robot from the S4R dataset of the HuRIC corpus (Bastianelli et al., 2014). The dataset consisted of 96 imperative utterances from a task in which people were

²The current policy is that if the response utterance is an acknowledgment then the system will produce a last-item overlap, otherwise it will aim for the TRP with no overlap.

Algorithm 1 Turn-Entry Manager Algorithm

```
1: procedure TEM(Utterance u)
2:   for all word ∈ u do
3:     Prediction p = generatePrediction(word)
4:     if p.probability > probThreshold then
5:       p.score = p.cost * p.confidence
6:       if p.score >= scoreThreshold then
7:         p.TRP_entry = p.startTime + p.duration - TTS_delay
8:         p.LI_entry = p.startTime + p.duration - TTS_delay - p.lastWord.duration
9:         actionStatus = simulateAction(p.action)
10:        if actionStatus == fail then
11:          generateResponse(failure)
12:        else
13:          p.response = setResponse()
14:          p.setOverlapType(p.response)
15:          performAction(p.action)
16:        end if
17:      end if
18:    end if
19:  end for
20: end procedure
21:
22: procedure WAITTOSPEAK(Prediction p)
23:   if currentTime >= p.TRP_entry then
24:     generateResponse(p.response)
25:   end if
26: end procedure
```

asked to give commands to a physical robot operating in a household environment. The language was unscripted and had few constraints, though people were told about the robot’s capabilities and the locations and objects that it could recognize. While the evaluation corpus contains only directives (no dialogue), it includes the kinds of utterances commonly seen in situated task-based dialogues, to which a robot would need to promptly respond (and potentially initiate early), and serves as a useful benchmark to test our framework.

The central aim of the evaluation is to show how a situated agent given these instructions can make predictions and respond at the TRP compared to a non-incremental baseline system. We also seek to demonstrate the potential for overlap production and preemptive action execution. In addition to the standard directives in the corpus, we also evaluate several variants of them which contain disfluency. It is important that SDSs are resilient to disfluency, as it is common in team communication channels (particularly in remote communication)

and has been implicated in effective team performance (Gervits et al., 2016a,b). Including disfluent utterances in the evaluation was done to show that the algorithm can handle variations in the input and still produce timely responses. Table 1 lists the utterance subsets that were constructed from the original corpus data. These include: 1) the original utterance, 2) utterance-initial non-lexical filler, 3) non-lexical filler after the first word, 4) 200 ms pause before the final word, and 5) repetition self-repair of the first word.

Subset	Example Utterance
1	go to the kitchen
2	<um>go to the kitchen
3	go <uh>to the kitchen
4	go to the <200 ms pause>kitchen
5	go- go to the kitchen

Table 1: Utterance subsets used in the evaluation

4.1 Approach

First, the text strings from the HuRIC corpus were extracted, along with the frequency distribution of bigrams. Parse rules (linking the text string to a semantic form) and actions (specifying the pre-conditions and effect) were defined for each utterance, and we generated the 5 subsets (see Table 1) for each utterance in the corpus.

While the system is capable of processing speech incrementally, we used incremental text input for the evaluation in order to abstract away some of the ASR noise (e.g., latency, errors, etc.)³. To simulate the timing associated with real speech, we added a delay before each word corresponding to 180 ms x the number of syllables in the word. This decision is based on the upper bound of the estimated duration of a syllable from Wilson and Wilson (2005), and is roughly in line with data from the *Switchboard* corpus, in which the mean syllable duration was 200 ms (SD: 103) (Greenberg, 1999). To handle the disfluency in Subsets 2-5, we used a simple keyword-spotting approach to detect fillers and pauses in the input, like most ASRs can do. These fillers were excluded from the recognizer result, but importantly their duration was added to the timing. We assume that fillers such as *uh* and *um* are one syllable in length, and so have a duration of 180 ms. While not all types of disfluencies are handled with these subsets, we leave prolongations and more complex self-repairs for future work.

The turn-taking policy used in the evaluation is that the robot will attempt to come in at the TRP if it made an early prediction and the action status of the prediction was successful. If the action status was a failure then the robot will overlap with the failure explanation immediately. The *score* threshold was set to 0 to maximize data collection. Other policies are, however, possible such as never overlapping, or using a higher score threshold to minimize wrong predictions for costly actions.

4.1.1 Measures and Hypotheses

Our primary measure was the Floor Transfer Offset (FTO), a term introduced by de Ruiter et al. (2006). FTO is defined as the time difference in ms between the start of a turn and the end of the previous turn. Positive values indicate gaps whereas negative values indicate overlap. We also computed

³In particular, we experienced significant delays with incremental speech input. This is likely due to our Sphinx4 configuration, as others have reported much faster performance with the same ASR (Baumann and Schlangen, 2012).

the accuracy of the prediction model, the timing of when a prediction was made, and the point at which an action was initiated.

Overall, we expected the algorithm to perform well for the majority of examples in Subset 1, leading to smaller FTOs compared to a non-incremental system. This gives us:

H1: *Incremental utterance prediction would lead to smaller FTOs and earlier actions than non-incremental processing without online prediction.*

The non-incremental baseline system we used is a similar DIARC configuration, with the main difference being that input is non-incremental and the Utterance Predictor component is bypassed. We ran utterances from Subset 1 in which a correct prediction was made through this non-incremental configuration to compare performance. Next, we expected that the timing in the system would work out such that it can time its turn to come in at or near the TRP for actionable predictions, and much earlier for un-actionable ones. Thus we have:

H2: *Incremental utterance prediction would enable the system to 1) hit the TRP entry point for responses to actionable predictions, 2) initiate those actions early, 3) and produce interjacent (mid-speech) overlap for un-actionable predictions.*

If the system makes an early prediction, subsequent processing takes minimal time, so it should be able to hit the TRP for all but very late predictions. It would also be able to initiate the action shortly after the DM receives the prediction. For early predictions that are not actionable, it should produce an interjacent overlap well before the utterance is finished. Finally, we expect performance on Subsets 2-5 to be dependent on whether a prediction was made before or after the disfluency was detected. This is because the TRP entry point is computed from the expected duration of the predicted utterance, and this duration may be incorrect if the prediction did not incorporate the additional timing of the disfluency. This leads to:

H3: *The approach would be robust to disfluency in the input, but only if the disfluency was detected before a prediction was made.*

Given H3, we expect the FTO for Subsets 1 and 2 to be close to 0 for correct predictions, since these involve either no filler or an utterance-initial filler (which will always be detected before a prediction is made). Subset 4 will likely have a negative FTO, as predictions will usually be made before the final word, and so the 200 ms pause will not be added to

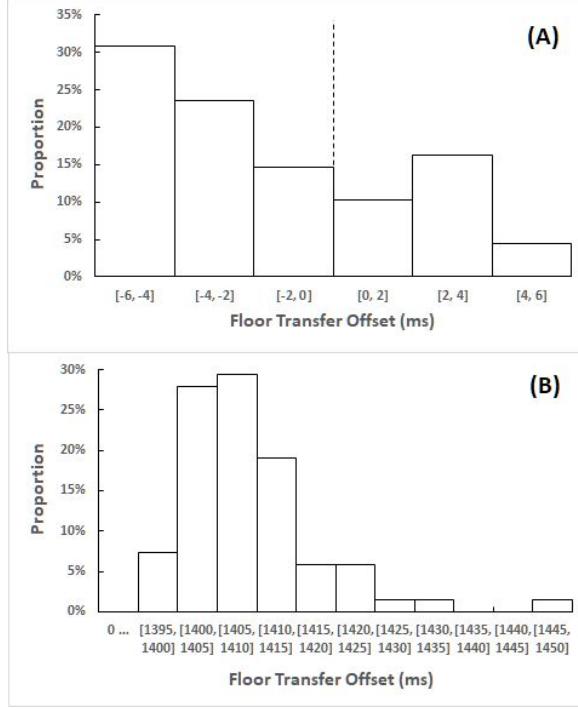


Figure 3: Histograms showing the floor transfer offset for A) predictive system and B) non-incremental (baseline) system for cases in which a correct early prediction was made. N = 68.

the utterance duration, leading to earlier turn entry.

4.2 Results

Below we present the results of the evaluation described in Sec. 4. In general, prediction accuracy of our bigram model was 70.8% with 340 of 480 test utterances predicted correctly. On average, a prediction was made $50.8 \pm 17.7\%$ of the way into an utterance, duration-wise.

4.2.1 H1: Incremental vs Non-Incremental Processing

H1 dealt with the difference in FTOs between our framework implementation and a non-incremental baseline configuration of the same architecture. We compared the correctly-predicted utterances from Subset 1 (N=68) and the same utterances tested on the baseline system. A Welch's independent-samples t-test showed a significant difference between FTOs for the incremental prediction cases ($M = -1.1 \pm 3.2$ ms) compared to the baseline cases ($M = 1409.5 \pm 8.6$ ms), $t(85) = 1259.2$, $p < .001$ (see Figure 3). These results support *H1* in that a system running our framework was able to take a turn significantly earlier than a non-incremental one that did not use the framework.

4.2.2 H2: Timing Turn-Entry

H2 stated that our framework implementation would allow the system to reliably come in at the TRP for actionable predictions, and produce early failure explanations in the form of interjacent overlap for un-actionable (i.e., incorrect) predictions. For Subset 1 (fluent) utterances, we found a mean FTO of -1.1 ± 3.2 ms. Since an FTO of 0 means a seamless transition, these results support *H2* in that the system was able to time its turn to hit the TRP very accurately for actionable predictions. For those predictions that were un-actionable in Subset 1, the system produced a failure explanation with a mean FTO of -683.2 ± 713.7 ms (see Figure 4). The earliest FTO was -2780 ms and the latest was -8 ms. These results provide further support for *H2* in that the system was able to provide early failure explanations (i.e., interjacent overlap) when a predicted action could not be performed. See Table 2 for an overview of the results.

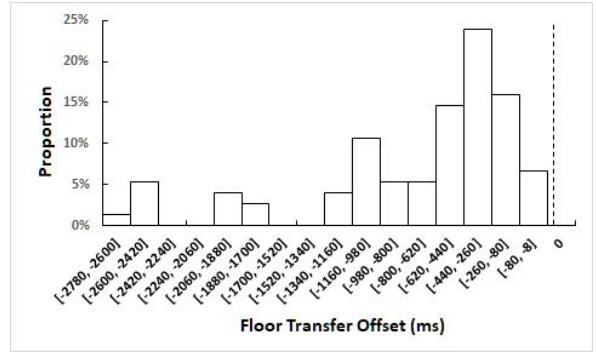


Figure 4: Floor Transfer Offset for cases in which the predicted utterance could not be performed. The system produced an interjacent overlap at the earliest opportunity. N = 15

To demonstrate preemptive action execution, we calculated (for Subset 1) the earliest point at which an action can be initiated. This is the point at which a prediction was sent to the DM and the preconditions for the corresponding action were checked. The difference between the end of the utterance and this point was 635 ± 197 ms, meaning that on average, an action could be initiated 635 ms before the end of an utterance.

As a supplementary analysis and to evaluate performance with varying syllable duration, we tested 10 random utterances from Subset 1 in which each syllable in the input was assigned a random duration between 100 and 200 ms (following Greenberg (1999)). The average syllable count for these utterances was 5.7 ± 1.6 and the average FTO was -16.5

	TRP Entry			Interjacent Entry		
	N	FTO (early prediction)	N	FTO (no prediction)	N	FTO
All	340	-55.7 ± 88.0	65	157.9 ± 47.5	75	-709.3 ± 714.2
Subset 1	68	-1.1 ± 3.2	13	164.8 ± 21.3	15	-683.2 ± 713.7
Subset 2	68	-5.5 ± 31.8	13	148.3 ± 19.4	15	-710.9 ± 703.5
Subset 3	68	-40.7 ± 75.9	13	147.8 ± 15.4	15	-746.4 ± 717.6
Subset 4	68	-191.2 ± 46.1	13	149.9 ± 13.9	15	-687.6 ± 780.1
Subset 5	68	-41.0 ± 75.6	13	176.0 ± 101.4	15	-630.6 ± 661.8

Table 2: Table of evaluation results. Mean values for Floor-Transfer Offset (FTO) are displayed for all evaluation cases ($N = 480$). For a given case, either an early prediction was made, or no prediction was made. If the prediction was correct and actionable, then a TRP entry was selected and an acknowledgment was produced. If the prediction was un-actionable (i.e., incorrect), then an interjacent overlap was selected and a failure explanation was produced.

± 87.9 ms, with a range of -155 to 152 ms. The difference between these results and the original set was that the predicted duration could now be wrong, and this was reflected in the slightly early entry times. Still, the mean FTO was close to 0, suggesting that the model still performs well with variable input.

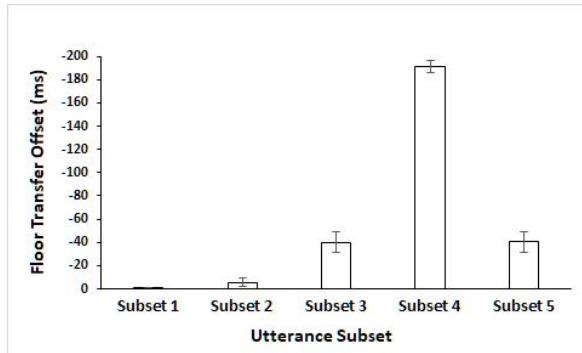


Figure 5: Floor Transfer Offset for correct predictions in each utterance subset. $N = 340$

4.2.3 H3: Robustness to Disfluency

To evaluate $H3$, which involved the robustness of the algorithm to disfluency in the input, we analyzed all of the disfluency cases in which a correct prediction was made (Subsets 2-5; $N = 272$). As expected, a key factor in correct timing here had to do with whether the prediction was made before or after the filler. This was confirmed with an independent-samples t-test, which found a significant difference between FTOs for predictions made after the filler ($M = -2.4 \pm 0.12$ ms) compared to those made before the filler ($M = -188.1 \pm 0.17$ ms), $t(127) = 44.6$, $p < .001$. Predictions made before the filler were most common in Subset 4 (making up 69% of the examples) and predictions made after the filler were made up entirely of Sub-

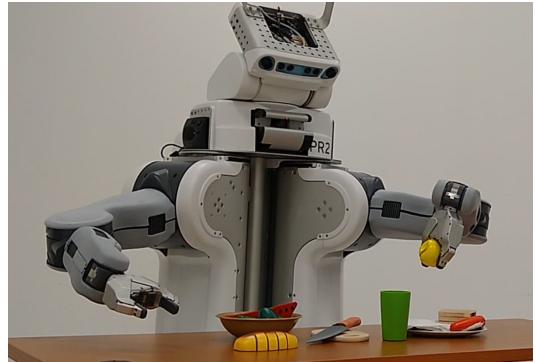


Figure 6: Robot performing a situated interactive task involving dialogue.

sets 2, 3, and 5. In Figure 5, we show the mean FTO for each of the utterance subsets.

4.3 Demonstration

To supplement the evaluation and show a real-world use-case, we ran the framework on a PR2 robot using real speech input (see Figure 6). A video of the interaction is available at <https://vimeo.com/410675260>. This video compares our baseline (non-incremental) system to the system running our turn-entry timing framework, and demonstrates that a robot can reliably make predictions about ongoing utterances using speech input, and that it can initiate actions and responses early.

5 Discussion

5.1 Contributions

Overall, we found support for $H1$, $H2$, and partial support for $H3$.

For $H1$, we demonstrated that our system was able to take a turn significantly faster than a non-incremental version of the same architecture. This

is not surprising, as the advantages of incremental processing are well known (Baumann et al., 2017). However, the comparison quantifies the amount of time that our approach saves.

H2 was also supported in that the system was able to hit the TRP very accurately for correct and actionable predictions (see Figure 3 A). Moreover, those actions were initiated on average 635 ms before the TRP, providing further support for *H2*. For un-actionable predictions, interjacent overlap was produced on average 709 ms before the TRP, suggesting very early turn entry (see Figure 4).

Finally, *H3* was partially supported in that fillers that were processed before a prediction (i.e., utterance-initial fillers) had their duration added to the overall utterance duration, but fillers towards the end of an utterance (after the prediction) were not detected in time. In these latter cases, the system came in earlier than expected (40 - 191 ms early), which would be a last-item overlap, and would likely not require repair (see Figure 5).

Overall, our domain-independent framework can be integrated into various SDSs in order to support responsive dialogue behavior and early actions, as well as to enable certain kinds of overlap that would not be achievable in other approaches.

5.2 Limitations and Future Work

One limitation is that the evaluation involved text rather than real speech and only considered simple directives. More work is clearly needed to evaluate the accuracy of the proposed approach with respect to variable speech input. Nevertheless, state-of-the-art ASRs can display very low recognition latency (e.g., Baumann et al. (2009)), suggesting that this would not significantly change our results.

Another limitation is that a fixed syllable duration was used to estimate timing, which was the same duration used in the input text. Since syllable length is a parameter in the model, this can be adjusted as needed to better estimate spoken syllable length. We have shown in a supplementary analysis on 10 utterances that the approach works reasonably well with variable syllable length. Future work will test other methods of estimating utterance length, including the clever duration modeling technique used in Baumann and Schlangen (2011) involving the ASR and TTS modules. The current results can be thought of as a best case scenario, and we expect that with more accurate duration estimates of real speech, system performance will

approach this upper bound.

Recovering from incorrect predictions is an important area for future work. Currently, when the system makes a prediction it cannot change it, even if new input comes in that contradicts the prediction (this is because the timing is very tight). In future work, it should be possible for the *TEM* to be updated if the prediction changes. This will support the handling of utterances such as those in Subset 4 which were characterized by late pauses.

Finally, the prediction model itself can be improved, perhaps through the use of a neural approach (Maier et al., 2017) or one that incorporates syntactic or prosodic features (Ward et al., 2010). Though we focus on lexico-syntactic cues for prediction, future work could leverage recent findings suggesting that prosody is more important to end-of-turn projection than previously thought (Barthel et al., 2016; Bögels and Torreira, 2015).

6 Conclusion

We have introduced a framework for turn-entry timing in human-robot dialogue which enables a situated agent to make incremental predictions about an ongoing utterance and time its turn to hit a variety of entry points. We implemented the framework in a robotic architecture and evaluated it on a corpus of human-robot directives from a situated interactive task. The system integrating our framework is significantly faster than a non-incremental system, and can produce fluid responses and various types of overlap, as well as execute actions preemptively. Moreover, the approach is robust to several forms of disfluency in the input. This framework offers a number of benefits for situated dialogue agents, including better responsiveness, the ability to produce various types of overlap (interjacent, last-item, backchannels, and collaborative completions), and preemptive action execution. These interactive capabilities are a step toward more natural and flexible turn-taking for situated dialogue agents.

Acknowledgments

This work was partly funded by a NASA Space Technology Research Fellowship under award 80NSSC17K018 and AFOSR Grant FA9550-18-1-0465. We are especially grateful to Brad Oosterveld, Evan Krause, and Zach Munro for their technical assistance.

References

- Mathias Barthel, Sebastian Sauppe, Stephen C Levinson, and Antje S Meyer. 2016. The timing of utterance planning in task-oriented dialogue: Evidence from a novel list-completion paradigm. *Frontiers in Psychology*, 7:1858.
- Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, Luca Iocchi, Roberto Basili, and Daniele Nardi. 2014. Huric: a human robot interaction corpus. In *Proceedings of LREC 2014*, pages 4519–4526.
- Timo Baumann, Michaela Atterer, and David Schlangen. 2009. Assessing and improving the performance of speech recognition for incremental systems. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 380–388.
- Timo Baumann, Casey Kennington, Julian Hough, and David Schlangen. 2017. Recognising conversational speech: What an incremental asr should do for a dialogue system and how to get there. In *Dialogues with Social Robots*, pages 421–432. Springer.
- Timo Baumann and David Schlangen. 2011. Predicting the micro-timing of user input for an incremental spoken dialogue system that completes a user's ongoing turn. In *Proceedings of the SIGDIAL 2011 Conference*, pages 120–129. Association for Computational Linguistics.
- Timo Baumann and David Schlangen. 2012. Inpro_iss: A component for just-in-time incremental speech synthesis. In *Proceedings of the ACL 2012 System Demonstrations*, pages 103–108.
- Sara Bögels and Francisco Torreira. 2015. Listeners use intonational phrase boundaries to project turn ends in spoken interaction. *Journal of Phonetics*, 52:46–57.
- Dan Bohus and Eric Horvitz. 2011. Decisions about turns in multiparty conversation: from perception to action. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 153–160.
- David DeVault, Kenji Sagae, and David Traum. 2009. Can i finish?: learning when to respond to incremental interpretation results in interactive dialogue. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 11–20. Association for Computational Linguistics.
- David DeVault, Kenji Sagae, and David Traum. 2011a. Detecting the status of a predictive incremental speech understanding model for real-time decision-making in a spoken dialogue system. In *Twelfth Annual Conference of the International Speech Communication Association*.
- David DeVault, Kenji Sagae, and David Traum. 2011b. Incremental interpretation and prediction of utterance meaning for interactive dialogue. *Dialogue & Discourse*, 2(1):143–170.
- Paul Drew. 2009. Quit talking while I'm interrupting: a comparison between positions of overlap onset in conversation. In *Talk in Interaction: Comparative Dimensions*, pages 70–93.
- Jens Edlund, Joakim Gustafson, Mattias Heldner, and Anna Hjalmarsson. 2008. Towards human-like spoken dialogue systems. *Speech communication*, 50(8–9):630–645.
- Felix Gervits, Kathleen Eberhard, and Matthias Scheutz. 2016a. Disfluent but effective? a quantitative study of disfluencies and conversational moves in team discourse. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3359–3369.
- Felix Gervits, Kathleen Eberhard, and Matthias Scheutz. 2016b. Team communication as a collaborative process. *Frontiers in Robotics and AI*, 3:62.
- Felix Gervits and Matthias Scheutz. 2018a. Pardon the interruption: Managing turn-taking through overlap resolution in embodied artificial agents. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 99–109.
- Felix Gervits and Matthias Scheutz. 2018b. Towards a conversation-analytic taxonomy of speech overlap. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference (LREC)*.
- Felix Gervits, Dean Thurston, Ravenna Thielstrom, Terry Fong, Quinn Pham, and Matthias Scheutz. 2020. Toward genuine robot teammates: Improving human-robot team performance using robot shared mental models. In *19th International Conference on Autonomous Agents and Multi-Agent Systems*.
- Steven Greenberg. 1999. Speaking in shorthand—a syllable-centric perspective for understanding pronunciation variation. *Speech Communication*, 29(2–4):159–176.
- Silvan Heintze, Timo Baumann, and David Schlangen. 2010. Comparing local and sequential models for statistical incremental natural language understanding. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 9–16. Association for Computational Linguistics.
- Julian Hough and David Schlangen. 2016. Investigating fluidity for human-robot interaction with real-time, real-world grounding strategies. In *Proceedings of the 17th Annual SIGdial Meeting on Discourse and Dialogue*.
- Gail Jefferson. 2004. A sketch of some orderly aspects of overlap in natural conversation. *Pragmatics and Beyond New Series*, 125:43–62.

- Casey Kennington, Spyridon Kousidis, and David Schlangen. 2013. Interpreting situated dialogue utterances: an update model that uses speech, gaze, and gesture information. *Proceedings of SigDial 2013*.
- Fredrik Kronlid. 2006. Turn taking for artificial conversational agents. In *International Workshop on Cooperative Information Agents*, pages 81–95. Springer.
- Divesh Lala, Pierrick Milhorat, Koji Inoue, Masanari Ishida, Katsuya Takanashi, and Tatsuya Kawahara. 2017. Attentive listening system with backchanneling, response generation and flexible turn-taking. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 127–136.
- Stephen C Levinson and Francisco Torreira. 2015. Timing in turn-taking and its implications for processing models of language. *Frontiers in psychology*, 6:731.
- Lilla Magyari and Jan-Peter de Ruiter. 2012. Prediction of turn-ends based on anticipation of upcoming words. *Frontiers in psychology*, 3:376.
- Angelika Maier, Julian Hough, and David Schlangen. 2017. Towards deep end-of-turn prediction for situated spoken dialogue systems. *Proceedings of INTERSPEECH 2017*.
- Maike Paetzl, Ramesh Manuvinakurike, and David DeVault. 2015. so, which one is it? the effect of alternative incremental architectures in a high-performance game-playing agent. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 77–86.
- Antoine Raux and Maxine Eskenazi. 2009. A finite-state turn-taking model for spoken dialog systems. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 629–637.
- Antoine Raux and Maxine Eskenazi. 2012. Optimizing the turn-taking behavior of task-oriented spoken dialog systems. *ACM Transactions on Speech and Language Processing (TSLP)*, 9(1):1–23.
- Jan-Peter de Ruiter, Holger Mitterer, and Nick J Enfield. 2006. Projecting the end of a speaker’s turn: A cognitive cornerstone of conversation. *Language*, 82(3):515–535.
- Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, pages 696–735.
- Kenji Sagae, Gwen Christian, David DeVault, and David Traum. 2009. Towards natural language understanding of partial speech recognition results in dialogue systems. In *Proceedings of The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 53–56.
- Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasantha Sarathy, and Tyler Frasca. 2019. An overview of the distributed integrated cognition affect and reflection diarc architecture. In *Cognitive Architectures*, pages 165–193. Springer.
- David Schlangen and Gabriel Skantze. 2011. A general, abstract model of incremental dialogue processing. *Dialogue & Discourse*, 2(1):83–111.
- Gabriel Skantze. 2017. Towards a general, continuous model of turn-taking in spoken dialogue using lstm recurrent neural networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 220–230, Saarbrcken, Germany. Association for Computational Linguistics.
- Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 745–753. Association for Computational Linguistics.
- Tanya Stivers, Nicholas J Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter de Ruiter, Kyung-Eun Yoon, et al. 2009. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106(26):10587–10592.
- Khiet P Truong, Ronald Poppe, and Dirk Heylen. 2010. A rule-based backchannel prediction model using pitch and pause information. In *Eleventh Annual Conference of the International Speech Communication Association*.
- Nigel G Ward, Olac Fuentes, and Alejandro Vega. 2010. Dialog prediction for a general model of turn-taking. In *Eleventh Annual Conference of the International Speech Communication Association*.
- Margaret Wilson and Thomas P Wilson. 2005. An oscillator model of the timing of turn-taking. *Psychonomic bulletin & review*, 12(6):957–968.
- Tiancheng Zhao, Alan W Black, and Maxine Eskenazi. 2015. An incremental turn-taking model with active system barge-in for spoken dialog systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 42–50.

Learning Word Groundings from Humans Facilitated by Robot Emotional Displays

David McNeill

Boise State University
1910 W. University Dr.,
Boise, ID 83725
davidmcneill@
u.boisestate.edu

Casey Kennington

Boise State University
1910 W. University Dr.,
Boise, ID 83725
caseykennington@
boisestate.edu

Abstract

In working towards accomplishing a human-level acquisition and understanding of language, a robot must meet two requirements: the ability to learn words from interactions with its physical environment, and the ability to learn language from people in settings for language use, such as spoken dialogue. In a live interactive study, we test the hypothesis that emotional displays are a viable solution to the cold-start problem of how to communicate without relying on language the robot does not—indeed, cannot—yet know. We explain our modular system that can autonomously learn word groundings through interaction and show through a user study with 21 participants that emotional displays improve the quantity and quality of the inputs provided to the robot.

1 Introduction

In any first language acquisition task, three questions must be resolved:

1. *What kinds of words to be learned?*
2. *How to model those words' semantics?*
3. *How to overcome the cold-start problem?*

To answer the first question, we note that co-located spoken dialogue interaction is the fundamental setting of first language acquisition for humans (Fillmore, 1981; McCune, 2008) and that children generally tend to focus on physical objects first, as evidenced by age-of-acquisition datasets. For this reason, concrete words that denote physical objects are learned earlier than abstract words (Kuperman et al., 2013). This informs the answer to the second question: the model of semantics should be able to connect language with the physical world, which is part of the goal of *grounded semantics* (e.g., grounding a color word like *green* with visual information).

This still leaves the third question: how can a system learn word groundings in a physical, co-located setting without using words it has yet to learn? In answering this, there is evidence that having a physical body is a requirement for bootstrapping semantic learning of concrete word denotations (Smith and Gasser, 2005; Johnson, 2008). Therefore, a system that can use extra-linguistic cues through physical signals can potentially overcome the cold-start problem and learn words without uttering words it has never heard.

In this paper, we test the hypothesis that *emotional displays*, specifically *confusion* and *understanding* displays performed by an embodied robot, are a viable solution to the cold-start problem. Our reasons are two-fold: emotional displays can relate the robot's state to its human teacher, and emotional displays are developmentally appropriate for the most common language acquisition setting (i.e., an adult teaching a child) (Adolphs, 2002), and would therefore not lead a human user to make incorrect assumptions regarding the robot's level of comprehension.

In an interactive study with 21 participants, our robot independently and autonomously explored a physical setting and elicited relevant word references and feedback from the participants, who were tested both with a robot that displayed emotions and a robot that did not. For grounded semantics, we opted for a model that is incremental (i.e., operates at the word level), that can map individual words to physical features, and that can learn a mapping between a word and physical features using only a few examples—the *words-as-classifiers model* (WAC) (Kennington and Schlangen, 2015). In the WAC model, each word is represented by its own classifier trained on “not / is” examples of real-world referents. The WAC model has been used in interactive dialogue scenarios with robots before (Hough and Schlangen, 2017). Importantly,

our system not only learned word groundings as it interacted with participants, it also incorporated a reinforcement learning model to learn from positive or negative participant feedback which emotional valence (either understanding or confusion) to display. Analyzing the results from the surveys and the learned WAC classifiers, we discovered that the use of emotional displays improved the quantity and quality of the inputs provided to the robot, with the effect modulated by the valence and frequency of the emotional displays.

2 Background & Related Work

It has been shown that people assign anthropomorphic characteristics, social roles and models when interacting with robots (Kiesler and Goetz, 2002), which has implications for the kinds of settings and tasks that robots can carry out with human collaborators. One dimension that people anthropomorphically assign to robots is emotion. We cannot prevent users from making emotional judgements of a robot’s behavior (Novikova et al., 2015). Instead, if a robot’s behavior were designed to take these emotional judgements into account, the robot could be made more predictable and more interpretable by humans in a complex environment (Breazeal, 2005). Indeed, emotional features can make a robot appear more lifelike and believable to humans, thereby making humans more prone to accept and engage with them (Cañamero, 2005). Of course, the choice of emotions must be taken with care; Claret et al. (2017) showed that happiness and sadness emotional displays during primary tasks (e.g., such as transporting an object) could confuse human interlocutors as robot actions (e.g., jerkiness, activity, gaze), and robot movement are also emotionally interpreted.

Similar to *conversational grounding*, Jung (2017) explained how *affective grounding*—the coordination on content and process of affect—occurs between robots and human users. We handle this particular phenomenon by only considering a positive and negative valence of a single affective type (i.e., confusion vs. understanding), and by establishing through an evaluation that they are indeed interpreted the way we expect before we use them in a language learning task.

Robots have been used in many language grounding tasks; Matuszek (2018) gives an overview of the recent literature. In some cases the cold-start problem is handled by Wizard-of-Oz paradigm

studies where a robot that knows no word denotations interacts with human participants, but the robot is in fact being controlled by a confederate. In this paper, our robot is fully autonomous and has no pre-programmed language production capabilities; that is, the robot will never utter words it hasn’t encountered within an interaction.

Beyond word learning, our approach attempts to ground language and learn which emotions to display. This work builds on Ferreira and Lefèvre (2015) which outlined the approach we take for a reinforcement-learning based on “polarized user appraisals gathered throughout the course of a vocal interaction between a machine and a human”. Their work outlined the design of a hypothetical experiment; we have taken this a step further by actually implementing this design in a live interactive study. We take user feedback to be the explicit reward signal (those user inputs that match the explicit positive or negative feedback). Like their work, our approach does require a lengthy explore phase at the outset.

3 System

In this section we explain our choice of robot, and how we modeled the dialogue for language learning with integrated robot modules.

Choice of Robot: Anki Cozmo Cozmo is small, has track wheels for locomotion, a lift, and a head with an OLED display which displays its eyes. The head has a small camera and a speaker with a built-in speech synthesizer (with a “young”-sounding voice). With a Python SDK, we can easily access Cozmo’s sensors and control it. Importantly for our study, we will make use of Cozmo’s camera for object detection, human face recognition, and locomotion functionality for navigation between objects. Cozmo does not have an internal microphone—we make use of an external one.

The choice of robot affects how humans will treat it, and it is important for our study that users perceive the robot as a young language learning child. We opted for the Anki Cozmo robot because Plane et al. (2018) showed that participants in their study perceived Cozmo as young, but with potential to learn. Cozmo’s affordances are likewise consistent with this perceived age and knowledge-level. Cozmo is also a good option for this work because it has been recently demonstrated that humans perceive the same emotions and positive or negative valences from Cozmo’s over 940 pre-scripted be-

haviors (McNeill and Kennington, 2019). Taken together, these studies show that (1) we can safely assume that human participants will treat Cozmo at an appropriate age level, and (2) we can assume that human participants will properly interpret Cozmo’s behaviors as displays of emotion.

Indicating Objects If Cozmo is to learn denotations for physical objects, then objects must be present in the environment that Cozmo and a person share. Also, the person needs to be able to identify the object that Cozmo is attending to. Once these requirements are met, then Cozmo can learn the correct denotations for objects. Noting that Matuszek et al. (2014) has been able to successfully use deictic gestures to isolate objects, we assume participants will denote objects that the robots are already attending to, which is what adults do for children learning their first language (Hollich et al., 2000) (that is, the perspective Cozmo takes is ego-centric). More practically, Cozmo is small, which places its camera very low to the surface of the shared environment. Therefore, Cozmo must be very close to objects to “see” them through its camera, which effectively isolates objects without the need for deictic gestures from the robot. When Cozmo does need to indicate an object, Cozmo moves its lift up and down while directly in front of the object of intended reference.

Social Conventions Motivated by Michaelis and Mutlu (2019), Cozmo needs to exhibit minimal “socially adept” behaviors if language learning is going to take place. We identify two behaviors that we incorporate into Cozmo: (1) *eye contact*; that is, in certain states (e.g., Cozmo is looking for feedback from the user) Cozmo looks up and turns in place until it finds a face, and (2) *motion*; that is, Cozmo must nearly always be moving—for several reasons, first to signal to an interlocutor that Cozmo is still functional and second, children who are learning language rarely sit still. These random motions occur outside of the task actions (explained below) and give priority to those task actions when they occur.

Learning To answer the question *can emotions serve as scaffolding to solve the cold-start language learning problem?*, we take a reinforcement learning (RL) approach. Given a dialogue state and a robot state, the RL regime learns which emotional valence to display: confusion or understanding. This learning takes place at the same time that the

robot is learning grounded word meanings using WAC as it interacts with a person and its environment.

3.1 System Modules

For the balance of this section, we describe the modules that make up our word learning dialogue system and how they are integrated with the Cozmo robot. The modules include:

1. Visual Perception
2. Object Detection
3. Feature Extraction
4. Automatic Speech Recognition
5. Grounded Semantics
6. Action Management
 - Navigation
 - Emotional Displays
 - Word proposals
7. Emotion Management

Visual Perception The Visual Perception module handles the event of a new image being captured by Cozmo’s camera. Cozmo’s camera produces a color image at 30 frames per second (320x240 pixels). The output of this module is a single frame image.¹

Object Detection This module uses the Mask RCNN graph (He et al., 2017) adapted taken from the tensorflow library. We used a model pre-trained on a dataset of sixty separately labeled grocery items from the MVTec D2S dataset (Follmann et al., 2018). We apply this configuration of the Mask RCNN model for drawing bounding boxes around objects in images received from the Visual Perception module. We discard the labels and only make use of the bounding box information. The output of this module is the bounding box information of all detected objects in view.

Feature Extraction The Feature Extraction module contains an image classification model built on the Keras implementation of VGG19 (Simonyan and Zisserman, 2014) which is trained using the ImageNet (Deng et al., 2009) corpus weights.² This module takes an image and bounding box information, extracts each sub-image containing each object, then passes those through the Keras model, thereby extracting features. We use the second-to-last (i.e., $fc2$) layer as the feature

¹For our system, we only considered three frames per second and dropped the rest.

²We tested on more recent and principled models such as efficientnet (Tan and Le, 2019), but found the simpler Keras model to work better for our task.

representation of each object, which is a vector that represents the object. This model outputs a vector for each detected object.

We motivate this approach of using an existing object detector only for bounding box information and another model for object representation because pre-linguistic children can already detect isolated objects before they learn denotative words for those objects—our downstream *Grounded Semantic* module learns the mappings between words and objects. Moreover, this allows word learning to occur without relying on the limited vocabulary of any given object detector—those trained on Imagenet only have a vocabulary of 1000 words, and those words are generally nouns, whereas attributes such as color and shape (i.e., adjectives) should be allowed.

Automatic Speech Recognition The *Automatic Speech Recognition* (ASR) module transcribes user speech. This module then categorizes user speech according to three exclusive dialogue acts:

- positive user feedback (e.g., *yes*)
- negative user feedback (e.g., *no*)
- object denotations (all other words)

The positive and negative feedback dialogue acts are used as environment signals to our reinforcement learning regime and are identified by simple word spotting. All other utterances are regarded as object denotations for the Grounded Semantic module.

Grounded Semantic Module The Grounded Semantic Module is tasked with learning word denotations as well as determining which word to utter in certain states. As noted above, for this we leverage WAC. This module takes in transcribed speech from the ASR module and the top (i.e., most confident) object feature representations from the Feature Extraction module (i.e., one set of object features per word use). In an *explore* state, the robot records the feature representations that it receives and assigns them as positive examples to words that are heard within a 10 second window. Negative examples for words are taken from the largest rectangular area of the image from outside of the top bounding box. Anytime a word has been heard three times, the WAC classifier for that word is trained. The classifiers themselves are scikit-learn logistic regression classifiers (with L2 normalization).³ Trained clas-

sifiers can be improved each time a word is heard by re-training the classifier given the new training examples from the interaction.

Action Management For Action Management (which includes dialogue management), we use PyOpenDial (Jang et al., 2019). There are several navigational actions (the first three make up *explore* actions, the latter two *exploit* actions): `find-object`, `approach-object`, `indicate-object`, `propose-word`, `seek-face`. Several state variables are tracked to determine which of the above actions are taken, including the most recent navigation action, if the robot has found an object, and if the robot has approached an object. The robot begins in a `find-object` state where it does not yet see an object. This triggers random left and right turning, forward and backward driving until an object comes into view (determined by the Object Detection module). When an object is in view, the robot transitions to an `approach-object` state which alternates turning left and right to keep the object in the center of the robot’s camera frame while driving short distances until the object takes up a specified percentage of the camera frame. At this point the robot transitions to `indicate-object` which it accomplishes by moving its lift quickly up and down multiple times. When the Action Management module enacts a `propose-word` action, the robot utters a word that it “thinks” it learned (i.e., the robot has a trained classifier for the word in question and it fits above a certain threshold for the object). After a proposal, the robot enters performs a `seek-face` action to ground with the interlocutor that it expects them to give it positive or negative feedback.

Emotion Management This module is where the RL (i.e., reinforcement learning) takes place. The RL model (which leverages PyOpenDial Q-Learning functionality implemented as a dynamic Bayesian network with Dirichlet priors and a Gaussian posterior) tracks just a single variable: `robot-confidence` (RC), a number that represents the robot’s internal confidence that it should move into a `propose-word` state. The following modules affect the RC:

- ASR: if a positive feedback occurs anytime, the RC increases by 2; RC decreases by 4 if

³We attempted to use other classifiers, such as multi-layer perception, as well as other feature representations, such as

efficientnet, but found that this model is the most effective for fast language acquisition in this setting.

negative feedback is heard.

- Action Manager: if a *propose-word* state is reached (resulting in Cozmo uttering a word), *and* there is positive feedback from ASR, then the confidence increases by 5. If negative, the confidence decreases by 4.

The emotional displays take place before a *propose-word* action. This module uses RL to learn whether to display an *understanding* emotion or a *confusion* emotion. The above listed modules alter the RC dynamically over time (though the min/max values of RC are -10 and +10 respectively). The reward policy is as follows: if RC is positive, the policy is rewarded +5 for displaying understanding, and -5 if it displayed confusion; if RC is negative, the policy is rewarded -5 for displaying understanding and +5 for confusion. In this manner, the RL can determine, on its own, the RC threshold for producing understanding vs. confusion displays.⁴ We chose *confusion* and *understanding* for two reasons: first, because prior work has shown that confusion and understanding are opposite valences of the same affect which are very interpretable, particularly when looking at Cozmo’s movement and eyes (McNeill and Kennington, 2019); and second, because confusion and understanding are emotions that lend well to the language learning task—the robot can display confusion in states where it is unsure how to act, and understanding in states where it knows how to act. To determine which behaviors would be perceived by users as *confusion* or *understanding*, we collected Cozmo’s behaviors that were labeled with high confidence as either of those emotions by the model in McNeill and Kennington (2019). We then asked 7 people to watch recorded videos of Cozmo performing those emotions and rate them on a 5-point Likert scale. This resulted in 11 highly-rated behaviors (i.e., lasting from 3-10 seconds) for *confusion* or *understanding*. The Emotion Management model randomly selects one of the 11 for each emotion when producing a display of that emotion.

The full learning pipeline is depicted in Figure 1. Object detection occurs while users say words that refer to the objects in Comzo’s view. Object features are extracted and used for WAC to learn the

⁴More principled models of deep reinforcement learning are available, but we opted for this approach because we wanted our RL module to learn from minimal real interactions—deep learning approaches are known to require large amounts of data.

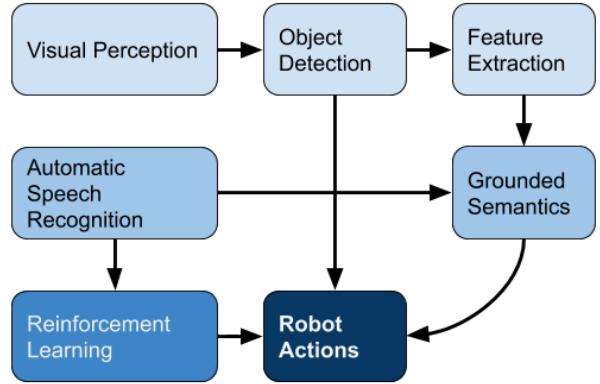


Figure 1: Schematic of our system: Visual perception passes camera frames to an Object Detector, which annotates objects with bounding boxes, then the Feature Extractor represents each of the detected objects as vectors that are passed to the Grounded Semantics module. The ASR transcribes speech, and passes those strings to the Grounded Semantics module and to the Reinforcement Learner (i.e., positive or negative feedback dialogue acts). Both Object Detection and Grounded Semantics pass their output to the Robot Actions (i.e., Action Management) which makes decisions about which actions to take, then actually performs those actions on the robot.

fitness between words and objects. If the word fits above a threshold, then Cozmo proposes that word to the user.

4 Evaluation

In this section, we explain how we evaluated our model with real human participants to determine if emotional displays increase engagement for language learning. We used two versions of our system: one which only performed the language learning task, and one which additionally included displays of emotion—the choice of which emotion was decided by a RL model. Our evaluation included objective measures logged by the system, as well as subjective measures collected using participant questionnaires.

4.1 Procedure

Study participants agreed to meet in a small room in the University’s Computer Science building. The conference room is set up for the participant interaction as follows: a table is placed to one side of the room, with one chair positioned in the middle of the longer side for the study participant. The experimenter sits at the head of the table, with a laptop positioned between himself and the participant. This laptop is running the robot’s interactive

script and the microphone that feeds the ASR module. A container of objects (specifically, pentomino blocks) is placed on the table; a handful of these have been randomly scattered on the table before the participant arrives in the room. The Cozmo robot is not introduced to the participant until the participant has signed an informed consent form and the task has been explained to them.

The experimenter was present to monitor the state of the robot and the microphone, troubleshoot any problems that might arise, and answer any questions the participant might have over the course of the interaction. The experimenter was permitted to offer a constrained set of coaching tips to the participant during the interaction, if the participant needed a reminder of the task or the initial instructions. The study participant and the robot were observed with cameras, which recorded audio and video from the interaction. Following each interaction the user moved to the experimenter’s laptop and completed a questionnaire. Following the completion of both interactions and subsequent surveys, the participant was paid eight U.S. dollars.

We recruited twenty-one study participants to interact with the Cozmo robot for two fifteen-minute periods over the course of a single session. Study participants were largely college students recruited from Boise State University’s Computer Science department. Participants’ ages range from their late teens to their forties. Eight of the participants were women; thirteen were men. Following each fifteen-minute interaction, the participant was asked to answer every question of the same questionnaire. The entire study took approximately one-hour.

We employed a within-group study design, meaning that each participant went through the same procedure twice, one time in which the independent variable (i.e., with emotional display) was present, and again when it was absent (i.e., without emotional display). To mitigate learning effects, the order in which the test condition was presented was alternated.

4.2 Task

First, the Cozmo robot was introduced to the participant, with an explanation of the following affordances and instructions: (1) Cozmo has a camera that can see them and the world; (2) Cozmo has a microphone that can hear them; (3) Cozmo doesn’t know anything, but is “curious” to learn more about the world; (4) for the next 15 minutes, it is the par-

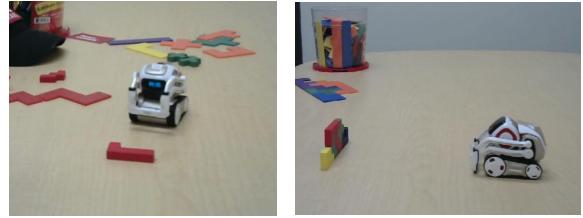


Figure 2: Cozmo looking down at an object (left) and looking up, seeking a face (right).

ticipant’s job to try to teach Cozmo as many words as they can, using the objects in the room, whatever they have on them, and their imagination; (5) if Cozmo gets off-track, they are allowed to pick Cozmo up and move it around; (6) when Cozmo is looking up, it is looking for their face; (7) when Cozmo “feels confident” enough, it will guess a word – if it gets it right, say “Yes.” If not, say, “No.” This feedback will help Cozmo learn. Figure 2 shows Cozmo in its task setting in two states: observing an object (left figure) and seeking a face (right figure).

4.3 Metrics

System Logs We track the number of utterances (termed “Heard Words”) made by the participants, including positive and negative feedbacks, and the number of proposals made by the robot which, taken together, form a proxy for engagement: higher numbers denote more engagement.

Participant Questionnaires We also evaluate the robot based on questionnaire responses written by the study participants following both sessions of the study. We used the Godspeed Questionnaire (Bartneck et al., 2009), a likert-scaled questionnaire with 24 questions ranging from negative to positive ratings of a robot’s anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety. In addition to the Godspeed questions, we also asked participants the following to ascertain their perceptions of our system and robot:

- How attached to the robot did the user feel?
- Were they engaged by the robot?
- What did they think the robot wanted?
- What did they think the robot was trying to do?
- Would they like to spend more time with the robot? Why or why not?

4.4 Results

Table 1 shows the results of the effect that emotional displays had on heard words, positive feedbacks, negative feedbacks, and proposals (note that

proposals represent trained WAC classifiers that reached the threshold for being uttered). Comparing the results of the experimental trials in which the robot displayed emotions to the control trials, it is apparent that the amount and quality of the user feedback to the robot improves in the presence of emotional displays. The sole caveat is negative feedback, which was offered the most on average by users interacting with a robot that wasn't making emotional displays.

Table 1: The effect of emotional displays on a language-acquisition task

(Mean / std. dev)	without emotions	with emotions
Heard Words	58.5 / 69.4	72.9 / 107.1
Positive Feedbacks	11.9 / 12.2	16.3 / 27.5
Negative Feedbacks	7.4 / 7.0	6.6 / 6.5
Proposals	7.8 / 7.8	9.8 / 7.5

Exploring the effect of participant learning on the language-acquisition task in Table 2 shows that users spoke more words and offered more positive feedback in the second trial than in the first, on average. Negative feedback was equivalent between the two trials, and the robot made more proposals in first trials, on average. This shows that learning effects had a minimal impact on user interaction with the robot.

Table 2: The effect of participant learning on the language-acquisition task

(Mean / std. dev)	first trial	second trial
Heard Words	60.6 / 70.3	64.1 / 103.3
Positive Feedbacks	9.8 / 11.4	16.8 / 26.5
Negative Feedbacks	6.7 / 6.9	6.7 / 6.7
Proposals	9.1 / 7.8	7.5 / 7.6

Next, we analyze the participant surveys to see if the presence of emotional displays biased the participant toward higher estimations of robot intelligence. For both the control and experimental trials, the average estimated age of the robot is two years old, which follows prior work using Cozmo (Plane et al., 2018) and is an appropriate assigned age range for this study. Additionally, the participant surveys reinforce the ambiguous role of emotion in human estimations of robot intelligence, irrespective to trial order, as seen in Figure 3.

User engagement also appeared largely uninfluenced by the presence of robot emotional displays, or the trial order, as seen in Figure 4. This is reinforced by the high p-value between user responses to the Godspeed questionnaire and the total number of emotional displays produced by the robot. As

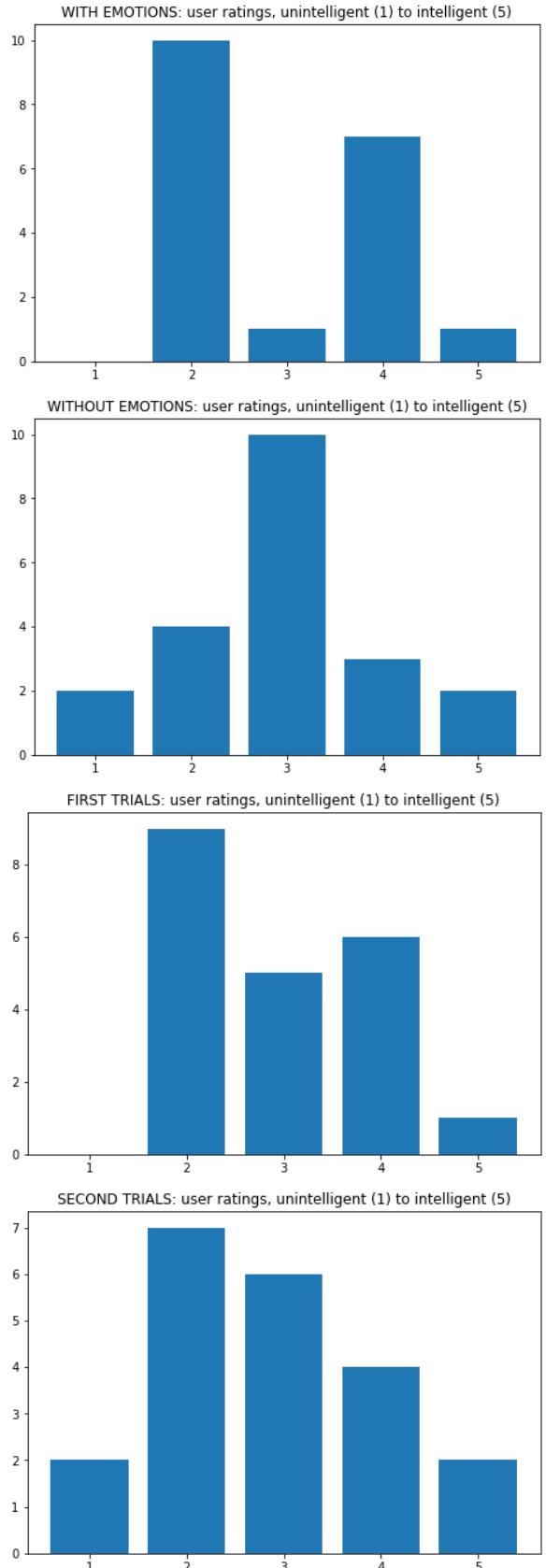


Figure 3: X-axis: Participants' ratings of robot intelligence from 1: unintelligent to 5: intelligent. Y-axis: the number of participants who selected that response.

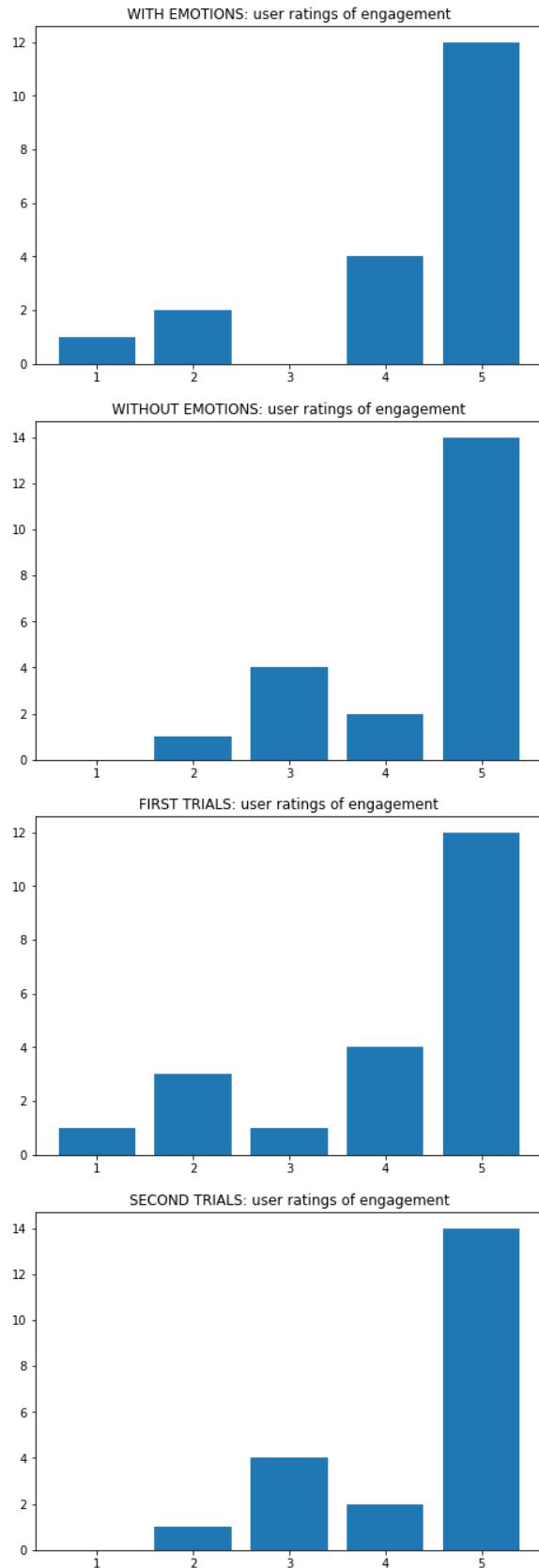


Figure 4: X-axis: Participants’ responses to the question, “Would you like to spend more time with the robot?” from 1: not at all to 5: very much. Y-axis: the number of participants who selected that response.

see in 3, there was a weak correlation and weak evidence to support a relationship between user interest and engagement with the robot, and the total number of emotional displays produced by the robot.

Table 3: Correlations between the total number of emotional displays and the following user questionnaire responses

	correlation	p-value
moves elegantly	0.48	0.03
is nice	0.40	0.09
is interesting to interact with	0.34	0.15
would like to spend more time with	0.17	0.49

In our RL module, the Q-Learning algorithm learned to put all weight onto one emotional display to the exclusion of the other for each interaction. This may have been due to the training batch size and training time for the Q-Learning algorithm (10 max samples and a 5 ms sample rate, rate to keep the interaction from slowing down). This did not have a negative effect on the choice of emotional displays produced by the robot; to the contrary, the emotional displays chosen by the RL module facilitated engagement.

5 Conclusion

We conducted an experiment with twenty-one participants who had to rely on the robot’s displays of *confusion* and *understanding* and their own performance in a language acquisition task as context. We analyzed our results by comparing the participants’ survey responses and the robots’ Grounded Semantics classifiers between the experimental and control trials. We found that a robot that displayed a combination of confused and understanding emotional displays – positive- and negatively-valenced emotion – gathered more inputs, and more useful inputs (positive feedback), than a robot that only engaged in task-specific actions (i.e., orienting to objects; seeking out the user’s face). This in turn led to the robot making more word proposals, which did not lead to greater engagement. User estimations of the robot were generally more positive estimations, supporting our choice of the Anki Cozmo robot for this task. Emotional displays did not incline participants to over-estimate the robot’s language understanding. We can conclude that emotion is an important aspect in handling the cold-start problem where a system can only use words it has heard.

In future work, we will test different policies for the reinforcement learning regime including measures for novelty rewards (i.e., hearing new words) as well as repeated words. Another aspect that demands further investigation would be the *timing* of emotional displays in the language learning interaction. Importantly, we will go beyond the two basic emotions explored here and incorporate additional emotions (e.g., the 8 valence pairs used in [McNeill and Kennington \(2019\)](#)) as the basis for additional engagement and perhaps use emotional states as features for the grounded classifiers.

Acknowledgements We thank the anonymous reviewers for their insights and feedback. This work was approved under the Boise State University IRB #126-SB20-012.

References

- Ralph Adolphs. 2002. Recognizing Emotion from Facial Expressions: Psychological and Neurological Mechanisms. *Behavioral and Cognitive Neuroscience Reviews*, 1(1):21–62.
- Christoph Bartneck, Dana Kulic, Elizabeth Croft, and Susana Zoghbi. 2009. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International journal of social robotics*, 1(1):71–81.
- Cynthia Breazeal. 2005. *Designing Socially Intelligent Robots*. National Academies Press, Washington, D.C.
- Lola Cañamero. 2005. Emotion understanding from the perspective of autonomous robots research. *Neural Networks*, 18(4):445–455.
- Josep Arnaud Claret, Gentiane Venture, and Luis Basañez. 2017. Exploiting the Robot Kinematic Redundancy for Emotion Conveyance to Humans as a Lower Priority Task. *International Journal of Social Robotics*, 9(2):277–292.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.
- Emmanuel Ferreira and Fabrice Lefèvre. 2015. Reinforcement-learning based dialogue system for human-robot interactions with socially-inspired rewards. *Computer Speech and Language*, 34(1).
- Charles J. Fillmore. 1981. Pragmatics and the description of discourse. *Radical pragmatics*, pages 143–166.
- Patrick Follmann, Tobias Böttger, Philipp Härtlinger, Rebecca König, and Markus Ulrich. 2018. MVTec D2S: Densely segmented supermarket dataset. In *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.
- He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings. *arXiv*.
- G J Hollich, K Hirsh-Pasek, and R M Golinkoff. 2000. Breaking the language barrier: An emergentist coalition model for the origins of language learning. *Monographs of the Society for Research in Child Development*, 65(3).
- Julian Hough and David Schlangen. 2017. It’s Not What You Do, It’s How You Do It: Grounding Uncertainty for a Simple Robot. In *Proceedings of the 2017 Conference on Human-Robot Interaction (HRI2017)*.
- Youngsoo Jang, Jongmin Lee, Jaeyoung Park, Kyeng-Hun Lee, Pierre Lison, and Kee-Eung Kim. 2019. PyOpenDial: A Python-based Domain-Independent Toolkit for Developing Spoken Dialogue Systems with Probabilistic Rules. In *Proceedings of EMNLP*.
- Mark Johnson. 2008. *The meaning of the body: Aesthetics of human understanding*. University of Chicago Press.
- Malte F Jung. 2017. Affective Grounding in Human-Robot Interaction. In *Proceedings of HRI’17*.
- Casey Kennington and David Schlangen. 2015. Simple Learning and Compositional Application of Perceptually Grounded Word Meanings for Incremental Reference Resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 292–301, Beijing, China. Association for Computational Linguistics.
- Sara Kiesler and Jennifer Goetz. 2002. Mental Models and Cooperation with Robotic Assistants. *CHI’02 extended abstracts on Human factors in computing systems*, pages 576–577.
- Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2013. Age-of-acquisition ratings for 30,000 English words.
- Cynthia Matuszek. 2018. Grounded Language Learning: Where Robotics and NLP Meet. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*.
- Cynthia Matuszek, Liefeng Bo, Luke S Zettlemoyer, and Dieter Fox. 2014. Learning from Unscripted Deictic Gesture and Language for Human-Robot Interactions. In *Proceedings of AAAI 2014*. AAAI Press.
- Lorraine McCune. 2008. *How Children Learn to Learn Language*. Oxford University Press.

- David McNeill and Casey Kennington. 2019. Predicting Human Interpretations of Affect and Valence in a Social Robot. In *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany.
- Joseph E. Michaelis and Bilge Mutlu. 2019. Supporting Interest in Science Learning with a Social Robot. In *Proceedings of the Interaction Design and Children on ZZZ - IDC '19*, pages 71–82, New York, New York, USA. ACM Press.
- Jekaterina Novikova, Gang Ren, and Leon Watts. 2015. It's Not the Way You Look, It's How You Move: Validating a General Scheme for Robot Affective Behaviour. In *Human-Computer Interaction – INTERACT 2015*, pages 239–258, Cham. Springer International Publishing.
- Sarah Plane, Ariel Marvasti, Tyler Egan, and Casey Kennington. 2018. Predicting Perceived Age: Both Language Ability and Appearance are Important. In *Proceedings of SigDial*.
- Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition.
- Linda Smith and Michael Gasser. 2005. The Development of Embodied Cognition: Six Lessons from Babies. *Artificial Life*, (11):13–29.
- Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv*.

Learning and Reasoning for Robot Dialog and Navigation Tasks

Keting Lu¹ Shiqi Zhang² Peter Stone³ Xiaoping Chen¹

¹ USTC ² SUNY Binghamton ³ UT Austin

ktlu@mail.ustc.edu.cn; zhangs@binghamton.edu;

pstone@cs.utexas.edu; xpchen@ustc.edu.cn

Abstract

Reinforcement learning and probabilistic reasoning algorithms aim at learning from interaction experiences and reasoning with probabilistic contextual knowledge respectively. In this research, we develop algorithms for robot task completions, while looking into the complementary strengths of reinforcement learning and probabilistic reasoning techniques. The robots learn from trial-and-error experiences to augment their declarative knowledge base, and the augmented knowledge can be used for speeding up the learning process in potentially different tasks. We have implemented and evaluated the developed algorithms using mobile robots conducting dialog and navigation tasks. From the results, we see that our robot's performance can be improved by both reasoning with human knowledge and learning from task-completion experience. More interestingly, the robot was able to learn from navigation tasks to improve its dialog strategies.

1 Introduction

Knowledge representation and reasoning (KRR) and reinforcement learning (RL) are two important research areas in artificial intelligence (AI) and have been applied to a variety of problems in robotics. On the one hand, KRR research aims to concisely represent knowledge, and robustly draw conclusions with the knowledge (or generate new knowledge). Knowledge in KRR is typically provided by human experts in the form of declarative rules. Although KRR paradigms are strong in representing and reasoning with knowledge in a variety of forms, they are not designed for (and hence not good at) learning from experiences of accomplishing the tasks. On the other hand, RL algorithms enable agents to learn by interacting with an environment, and RL agents are good at learning action policies from trial-and-error experiences toward maximizing long-term rewards un-

der uncertainty, but they are ill-equipped to utilize declarative knowledge from human experts. Motivated by the complementary features of KRR and RL, we aim at a framework that integrates both paradigms to enable agents (robots in our case) to simultaneously reason with declarative knowledge and learn by interacting with an environment.

Most KRR paradigms support the representation and reasoning of knowledge in logical form, e.g., Prolog-style. More recently, researchers have developed hybrid KRR paradigms that support both logical and probabilistic knowledge (Richardson and Domingos, 2006; Bach et al., 2017; Wang et al., 2019). Such logical-probabilistic KRR paradigms can be used for a variety of reasoning tasks. We use P-log (Baral et al., 2009) in this work to represent and reason with both human knowledge and the knowledge from RL. The reasoning results are then used by our robot to compute action policies at runtime.

Reinforcement learning (RL) algorithms can be used to help robots learn action policies from the experience of interacting with the real world (Sutton and Barto, 2018). We use model-based RL in this work, because the learned world model can be used to update the robot's declarative knowledge base and combined with human knowledge.

Theoretical Contribution: In this paper, we develop a learning and reasoning framework (called KRR-RL) that integrates logical-probabilistic KRR and model-based RL. The KRR component reasons with the qualitative knowledge from humans (e.g., it is difficult for a robot to navigate through a busy area) and the quantitative knowledge from model-based RL (e.g., a navigation action's success rate in the form of a probability). The hybrid knowledge is then used for computing action policies at runtime by planning with task-oriented partial world models. KRR-RL enables a robot to: i) represent

the probabilistic knowledge (i.e., world dynamics) learned from RL in declarative form; ii) unify and reason with both human knowledge and the knowledge from RL; and iii) compute policies at runtime by dynamically constructing task-oriented partial world models.

Application Domain: We use a robot delivery domain for demonstration and evaluation purposes, where the robot needs to **dialog with people** to figure out the delivery task’s goal location, and then physically **take navigation actions** to complete the delivery task (Thomason et al., 2020; Veloso, 2018). A delivery is deemed successful only if both the dialog and navigation subtasks are successfully conducted. We have conducted experiments using a simulated mobile robot, as well as demonstrated the system using a real mobile robot. Results show that the robot is able to learn world dynamics from navigation tasks through model-based RL, and apply the learned knowledge to both navigation tasks (with different goals) and delivery tasks (that require subtasks of navigation and dialog) through logical-probabilistic reasoning. In particular, we observed that the robot is able to adjust its dialog strategy through learning from navigation behaviors.

2 Related Work

Research areas related to this work include integrated logical KRR and RL, relational RL, and integrated KRR and probabilistic planning.

Logical KRR has previously been integrated with RL. Action knowledge (McDermott et al., 1998; Jiang et al., 2019) has been used to reason about action sequences and help an RL agent explore only the states that can potentially contribute to achieving the ultimate goal (Leonetti et al., 2016). As a result, their agents learn faster by avoiding choosing “unreasonable” actions. A similar idea has been applied to domains with non-stationary dynamics (Ferreira et al., 2017). More recently, task planning was used to interact with the high level of a hierarchical RL framework (Yang et al., 2018). The goal shared by these works is to enable RL agents to use knowledge to improve the performance in learning (e.g., to learn faster and/or avoid risky exploration). However, the KRR capabilities of these methods are limited to *logical* action knowledge. By contrast, we use a logical-probabilistic KRR paradigm that can directly reason with probabilities learned from RL.

Relational RL (RRL) combines RL with relational reasoning (Džeroski et al., 2001). Action models have been incorporated into RRL, resulting in a relational temporal difference learning method (Asgharbeigi et al., 2006). Recently, RRL has been deployed for learning affordance relations that forbid the execution of specific actions (Sridharan et al., 2017). These RRL methods, including deep RRL (Zambaldi et al., 2018), exploit structural representations over states and actions in (only) current tasks. In this research, KRR-RL supports the KRR of world factors beyond those in state and action representations, e.g., *time* in navigation tasks, as detailed in Section 4.2.

The research area of integrated KRR and probabilistic planning is related to this research. Logical-probabilistic reasoning has been used to compute informative priors and world dynamics (Zhang et al., 2017; Amiri et al., 2020) for probabilistic planning. An action language was used to compute a deterministic sequence of actions for robots, where individual actions are then implemented using probabilistic controllers (Sridharan et al., 2019). Recently, human-provided information has been incorporated into belief state representations to guide robot action selection (Chitnis et al., 2018). In comparison to our approach, learning (from reinforcement or not) was not discussed in the above-mentioned algorithms.

Finally, there are a number of robot reasoning and learning architectures (Tenorth and Beetz, 2013; Oh et al., 2015; Hanheide et al., 2017; Khandelwal et al., 2017), which are relatively complex, and support a variety of functionalities. In comparison, we aim at a concise representation for robot KRR and RL capabilities. To the best of our knowledge, this is the first work on a tightly coupled integration of logical-probabilistic KRR with model-based RL.

3 Background

We briefly describe the two most important building blocks of this research, namely model-based RL and hybrid KRR.

3.1 Model-based Reinforcement Learning

Following the Markov assumption, a Markov decision process (MDP) can be described as a four-tuple $\langle \mathcal{S}, \mathcal{A}, T, R \rangle$ (Puterman, 1994). \mathcal{S} defines the state set, where we assume a factored space in this work. \mathcal{A} is the action set. $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow$

$[0,1]$ specifies the state transition probabilities. $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ specifies the rewards. Solving an MDP produces an *action policy* $\pi : s \mapsto a$ that maps a state to an action to maximize long-term rewards.

RL methods fall into classes including model-based and model-free. Model-based RL methods learn a model of the domain by approximating $R(s,a)$ and $P(s'|s,a)$ for state-action pairs, where P represents the probabilistic transition system. An agent can then use planning methods to calculate an action policy (Sutton, 1990; Kocsis and Szepesvári, 2006). Model-based methods are particularly attractive in this work, because they output partial world models that can better accommodate the diversity of tasks we are concerned with, c.f., model-free RL that is typically goal-directed.

One of the best known examples of model-based RL is R-Max (Brafman and Tennenholtz, 2002), which is guaranteed to learn a near-optimal policy with a polynomial number of suboptimal (exploratory) actions. The algorithm classifies each state-action pair as known or unknown, according to the number of times it was visited. When planning on the model, known state-actions are modeled with the learned reward, while unknown state-actions are given the maximum one-step reward, R_{max} . This “maximum-reward” strategy automatically enables the agent to balance the exploration of unknown states and exploitation. We use R-Max in this work, though KRR-RL practitioners can use supervised machine learning methods, e.g., imitation learning (Osa et al., 2018), to build the model learning component.

3.2 Logical Probabilistic KRR

KRR paradigms are concerned with concisely representing and robustly reasoning with declarative knowledge. Answer set programming (ASP) is a non-monotonic logical KRR paradigm (Baral, 2010; Gelfond and Kahl, 2014) building on the stable model semantics (Gelfond and Lifschitz, 1988). An ASP program consists of a set of logical rules, in the form of “head :- body”, that read “head is true if body is true”. Each ASP rule is of the form:

```
a or ... or b :- c, ..., d, not e, ..., not f.
```

where $a \dots f$ are literals that correspond to true or false statements. Symbol `not` is a logical connective called *default negation*; `not l` is read as “it is not believed that l is true”, which does not imply that l is false. ASP has a variety of applications (Erdem et al., 2016).

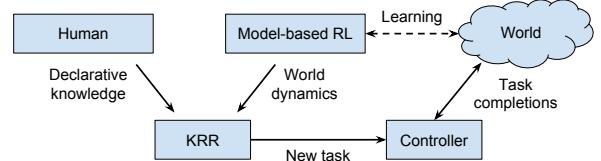


Figure 1: An overview of KRR-RL for robot learning and reasoning to complete complex tasks.

Traditionally, ASP does not explicitly quantify degrees of uncertainty: a literal is either true, false or unknown. P-log extends ASP to allow *probability atoms* (or *pr-atoms*) (Baral et al., 2009; Balai and Gelfond, 2017). The following pr-atom states that, if B holds, the probability of $a(t)=y$ is v :

$$\text{pr}(a(t)=y \mid B)=v.$$

where B is a collection of literals or their default negations; a is a random variable; t is a vector of terms (a term is a constant or a variable); y is a term; and $v \in [0, 1]$. Reasoning with an ASP program generates a set of *possible worlds*: $\{W_0, W_1, \dots\}$. The pr-atoms in P-log enable calculating a probability for each possible world. Therefore, P-log is a KRR paradigm that supports both logical and probabilistic inferences. We use P-log in this work for KRR purposes.

4 KRR-RL Framework

KRR-RL integrates logical-probabilistic KRR and model-based RL, and is illustrated in Figure 1. The KRR component includes both declarative qualitative knowledge from humans and the probabilistic knowledge from model-based RL. When the robot is free, the robot arbitrarily selects goals (different navigation goals in our case) to work on, and learns the world dynamics, e.g., success rates and costs of navigation actions. When a task becomes available, the KRR component dynamically constructs a partial world model (excluding unrelated factors), on which a task-oriented controller is computed using planning algorithms. **Human knowledge concerns environment variables and their dependencies**, i.e., what variables are related to each action. For instance, the human provides knowledge that navigation actions’ success rates depend on current time and area (say elevator areas are busy in the mornings), **while the robot must learn specific probabilities** by interacting with the environment.

Why is KRR-RL needed? Consider an indoor robot navigation domain, where a robot wants to

maximize the success rate of moving to goal positions through navigation actions. *Shall we include factors, such as time, weather, positions of human walkers, etc, into the state space?* On the one hand, to ensure model completeness, the answer should be “yes”. Human walkers and sunlight (that blinds robot’s LiDAR sensors) reduce the success rates of the robot’s navigation actions, and both can cause the robot irrecoverably lost. On the other hand, to ensure computational feasibility, the answer is “no”. Modeling whether one specific grid cell being occupied by humans or not introduces one extra dimension in the state space, and doubles the state space size. If we consider (only) ten such grid cells, the state space becomes $2^{10} \approx 1000$ times bigger. As a result, RL practitioners frequently have to make a trade-off between model completeness and computational feasibility. In this work, we aim at a framework that retains both model scalability and computational feasibility, i.e., the agent is able to learn within relatively little memory while computing action policies accounting for a large number of domain variables.

4.1 A General Procedure

In factored spaces, state variables $\mathcal{V} = \{V_0, V_1, \dots, V_{n-1}\}$ can be split into two categories, namely endogenous variables \mathcal{V}^{en} and exogenous variables \mathcal{V}^{ex} (Chermack, 2004), where $\mathcal{V}^{\text{en}} = \{V_0^{\text{en}}, V_1^{\text{en}}, \dots, V_{p-1}^{\text{en}}\}$ and $\mathcal{V}^{\text{ex}} = \{V_0^{\text{ex}}, V_1^{\text{ex}}, \dots, V_{q-1}^{\text{ex}}\}$. In our integrated KRR-RL context, \mathcal{V}^{en} is goal-oriented and includes the variables whose values the robot wants to actively change so as to achieve the goal; and \mathcal{V}^{ex} corresponds to the variables whose values affect the robot’s action outcomes, but the robot cannot (or does not want to) change their values. Therefore, \mathcal{V}^{en} and \mathcal{V}^{ex} both depend on task τ . Continuing the navigation example, robot position is an endogenous variable, and current time is an exogenous variable. For each task, $\mathcal{V} = \mathcal{V}^{\text{en}} \cup \mathcal{V}^{\text{ex}}$ and $n = p + q$, and RL agents learn in spaces specified by \mathcal{V}^{en} .

The KRR component models V , their dependencies from human knowledge, and conditional probabilities on how actions change their values, as learned through model-based RL. When a task arrives, the KRR component uses probabilistic rules to generate a task-oriented Markov decision process (MDP) (Puterman, 1994), which only contains a subset of \mathcal{V} that are relevant to the current task,

Procedure 1 Learning in KRR-RL Framework

Require: Logical rules Π^L ; probabilistic rules Π^P ; random variables $\mathcal{V} = \{V_0, V_1, \dots, V_{n-1}\}$; task selector Δ ; and guidance functions (from human knowledge) of $f^V(\mathcal{V}, \tau)$ and $f^A(\tau)$

- 1: **while** Robot has no task **do**
- 2: $\tau \leftarrow \Delta()$: a task is heuristically selected
- 3: $\mathcal{V}^{\text{en}} \leftarrow f^V(\mathcal{V}, \tau)$, and $\mathcal{V}^{\text{ex}} \leftarrow \mathcal{V} \setminus \mathcal{V}^{\text{en}}$
- 4: $A \leftarrow f^A(\tau)$
- 5: $\mathcal{M} \leftarrow \text{Procedure-2}(\Pi^L, \Pi^P, \mathcal{V}^{\text{en}}, \mathcal{V}^{\text{ex}}, A)$
- 6: Initialize agent: $\text{agent} \leftarrow R\text{-Max}(\mathcal{M})$
- 7: RL agent repeatedly works on task τ , and keeps maintaining task model \mathcal{M}' , until policy convergence
- 8: **end while**
- 9: Use \mathcal{M}' to update Π^P

i.e., \mathcal{V}^{en} , and their transition probabilities. Given this task-oriented MDP, a corresponding action policy is computed using value iteration or policy iteration.

Procedures 1 and 2 focus on how our KRR-RL agent learns by interacting with an environment when there is no task assigned.¹ Next, we present the details of these two interleaved processes.

Procedure 1 includes the steps of the learning process. When the robot is free, it interacts with the environment by heuristically selecting a task², and repeatedly using a model-based RL approach, R-Max (Brafman and Tennenholtz, 2002) in our case, to complete the task. The two guidance functions come from human knowledge. For instance, given a navigation task, it comes from human knowledge that the robot should model its own position (specified by f^V) and actions that help the robot move between positions (specified by f^A). After the policy converges or this learning process is interrupted (e.g., by task arrivals), the robot uses the learned probabilities to update the corresponding world dynamics in KRR. For instance, the robot may have learned the probability and cost of navigating through a particular area in early morning. In case this learning process is interrupted, the so-far-“known” probabilities are used for knowledge base update.

Procedure 2 includes the steps for building the probabilistic transition system of MDPs. The key point is that we consider only endogenous variables in the task-specific state space. However, when

¹As soon as the robot’s learning process is interrupted by the arrival of a real service task (identified via dialog), it will call Procedure 2 to generate a controller to complete the task. This process is not included in the procedures.

²Here curriculum learning in RL (Narvekar et al., 2017) can play a role to task selection and we leave this aspect of the problem for future work.

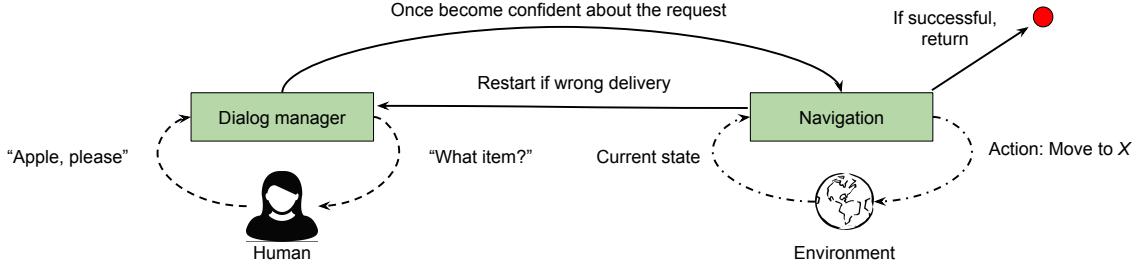


Figure 2: Transition system specified for delivery tasks, where question-asking actions are used for estimating the service request in dialog. Once the robot becomes confident about the service request, it starts to work on the navigation subtask. After the robot arrives, the robot might have to come back to the dialog subtask and redeliver, depending on whether the service request was correctly identified.

reasoning to compute the transition probabilities (Line 5), the KRR component uses both Π^P and \mathcal{V}^{ex} . The computed probabilistic transition systems are used for building task-oriented controllers, i.e., π , for task completions. In this way, the dynamically constructed controllers do not directly include exogenous variables, but their parameters already account for the values of all variables.

Next, we demonstrate how our KRR-RL framework is instantiated on a real robot.

4.2 An Instantiation on a Mobile Robot

We consider a mobile service robot domain where a robot can do navigation, dialog, and delivery tasks. A **navigation task** requires the robot to use a sequence of (unreliable) navigation actions to move from one point to another. In a **dialog task**, the robot uses spoken dialog actions to specify service requests from people under imperfect language understanding. There is the trend of integrating language and navigation in the NLP and CV communities (Chen et al., 2019; Shridhar et al., 2020). In this paper, they are integrated into **delivery tasks** that require the robot to use dialog to figure out the delivery request and conduct navigation tasks to physically fulfill the request. Specifically, a delivery task requires the robot to deliver item I to room R for person P , resulting in services in the form of $\langle I, R, P \rangle$. The challenges come from unreliable human language understanding (e.g., speech recognition) and unforeseen obstacles that probabilistically block the robot in navigation.

Human-Robot Dialog The robot needs spoken dialog to identify the request under unreliable language understanding, and navigation controllers for physically making the delivery.

The service request is not directly observable to the robot, and has to be estimated by asking questions, such as “What item do you want?” and “Is

Procedure 2 Model Construction for Task Completion

Require: $\Pi^L; \Pi^P; \mathcal{V}^{en}; \mathcal{V}^{ex}$; Action set A

- 1: **for** $V_i \in \mathcal{V}^{en}, i$ in $[0, \dots, |\mathcal{V}^{en}| - 1]$ **do**
- 2: **for** each possible value v in $range(V_i)$ **do**
- 3: **for** each $a \in A$ **do**
- 4: **for** each possible value v' in $range(V_i)$ **do**
- 5: $\mathcal{M}(v'|a, v) \leftarrow$ Reason with Π^L and Π^P w.r.t \mathcal{V}^{ex}
- 6: **end for**
- 7: **end for**
- 8: **end for**
- 9: **end for**
- 10: **return** \mathcal{M}

this delivery for Alice?” Once the robot is confident about the request, it takes a delivery action (i.e., $serve(I, R, P)$). We follow a standard way to use partially observable MDPs (POMDPs) (Kaelbling et al., 1998) to build our dialog manager, as reviewed in (Young et al., 2013). The state set S is specified using `curr_s`. The action set A is specified using `serve` and question-asking actions. Question-asking actions do not change the current state, and delivery actions lead to one of the terminal states (success or failure).³

After the robot becomes confident about the request via dialog, it will take a delivery action $serve\{I, R, P\}$. This delivery action is then implemented with a sequence of `act_move` actions. When the request identification is incorrect, the robot needs to come back to the shop, figure out the correct request, and redeliver, where we assume the robot will correctly identify the request in the second dialog. We use an MDP to model this robot navigation task, where the states and actions are specified using sorts `cell` and `move`. We use pr-atoms to represent the success rates of the unreliable movements, which are learned through model-based RL. The dialog system builds on our

³More details in the supplementary document.

previous work (Lu et al., 2017). Figure 2 shows the probabilistic transitions in delivery tasks.

Learning from Navigation We use R-Max (Brafman and Tennenholtz, 2002), a model-based RL algorithm, to help our robot learn the success rate of navigation actions in different positions. The agent first initializes an MDP, from which it uses R-Max to learn the partial world model (of navigation tasks). Specifically, it initializes the transition function with $T^N(s, a, s') = 1.0$, where $s \in \mathcal{S}$ and $a \in \mathcal{A}$, meaning that starting from any state, after any action, the next state is always s' . The reward function is initialized with $\mathcal{R}(s, a) = R_{max}$, where R_{max} is an upper bound of reward. The initialization of T^N and \mathcal{R} enables the learner to automatically balance exploration and exploitation. There is a fixed small cost for each navigation action. The robot receives a big bonus if it successfully achieves the goal (R_{max}), whereas it receives a big penalty otherwise ($-R_{max}$). A transition probability in navigation, $T^N(s, a, s')$, is not computed until there are a minimum number (M) of transition samples visiting s' . We recompute the action policy after E action steps.

Dialog-Navigation Connection The update of knowledge base is achieved through updating the success rate of delivery actions $\text{serve}(\mathcal{I}, \mathcal{R}, \mathcal{P})$ (in dialog task) using the success rate of navigation actions $\text{act_move}=\mathcal{M}$ in different positions.

$$T^D(s^r, a^d, s^t) = \begin{cases} P^N(s^{sp}, s^{gl}), \text{ if } s^r \odot a^d \\ P^N(s^{sp}, s^{mi}) \times P^N(s^{mi}, s^{sp}) \times P^N(s^{sp}, s^{gl}), \text{ if } s^r \otimes a^d \end{cases}$$

where $T^D(s^r, a^d, s^t)$ is the probability of fulfilling request s^r using delivery action a^d ; s^t is the “success” terminal state; s^{sp} , s^{mi} and s^{gl} are states of the robot being in the shop, a misidentified goal position, and real goal position respectively; and $P^N(s, s')$ is the probability of the robot successfully navigating from s to s' positions. When s^r and a^d are aligned in all three dimensions (i.e., $s^r \odot a^d$), the robot needs to navigate once from the shop (s^{sp}) to the requested navigation goal (s^{gl}). $P^N(s^{sp}, s^{gl})$ is the probability of the corresponding navigation task. When the request and delivery action are not aligned in at least one dimension (i.e., $s^r \otimes a^d$), the robot has to navigate back to the shop to figure out the correct request, and then redeliver, resulting in three navigation tasks.

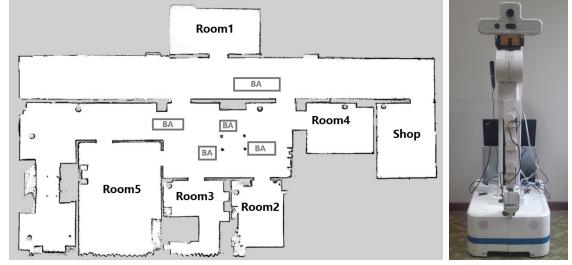


Figure 3: Occupancy-grid map used in our experiments (Left), including five rooms, one shop, and four blocking areas (indicated by ‘BA’), where all deliveries are from the shop and to one of the rooms; and (Right) mobile robot platform used in this research.

Intuitively, the penalty of failures in a dialog subtask depends on the difficulty of the wrongly identified navigation subtask. For instance, a robot supposed to deliver to a near (distant) location being wrongly directed to a distant (near) location, due to a failure in the dialog subtask, will produce a higher (lower) penalty to the dialog agent.

5 Experiments

In this section, the goal is to evaluate our hypothesis that our KRR-RL framework enables a robot to learn from model-based RL, reason with both the learned knowledge and human knowledge, and dynamically construct task-oriented controllers. Specifically, our robot learns from navigation tasks, and applied the learned knowledge (through KRR) to navigation, dialog, and delivery tasks.

We also evaluated whether the learned knowledge can be represented and applied to tasks under different world settings. In addition to simulation experiments, we have used a real robot to demonstrate how our robot learns from navigation to perform better in dialog. Figure 3 shows the map of the working environment (generated using a real robot) used in both simulation and real-robot experiments. Human walkers in the blocking areas (“BA”) can probabilistically impede the robot, resulting in different success rates in navigation tasks.

We have implemented our KRR-RL framework on a mobile robot in an office environment. As shown in Figure 3, the robot is equipped with two Lidar sensors for localization and obstacle avoidance in navigation, and a Kinect RGB-D camera for human-robot interaction. We use the Speech Application Programming Interface (SAPI) package (<http://www.iflytek.com/en>) for speech recognition. The robot software runs in the Robot Operating System (ROS) (Quigley et al., 2009).

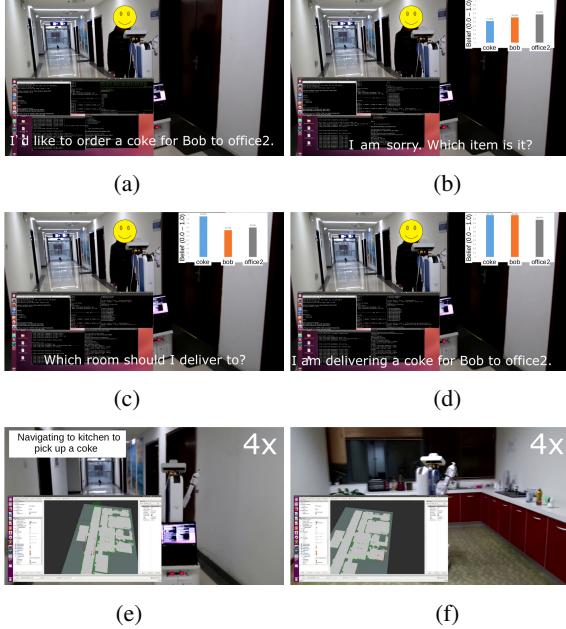


Figure 4: Screenshots of a demonstration trial on a real robot. (a) User gives the service request; (b) The robot decided to confirm about the item, considering its unreliable language understanding capability; (c) After hearing “coke”, the robot became more confident about the item, and decided to ask again about the goal room’; (d) After hearing “office2”, the robot became confident about the whole request, and started to work on the task; (e) Robot was on the way to the kitchen to pick up the object; and (f) Robot arrived at the kitchen, and was going to pick up the object for delivery.

An Illustrative Trial on a Robot: Figure 4 shows the screenshots of milestones of a demo video, which will be made available given its acceptance. After hearing “a coke for Bob to office2”, the three sub-beliefs are updated (turn1). Since the robot is aware of its unreliable speech recognition, it asked about the item, “Which item is it?” After hearing “a coke”, the belief is updated (turn2), and the robot further confirmed on the item by asking “Should I deliver a coke?” It received a positive response (turn3), and decided to move on to ask about the delivery room: “Should I deliver to office 2?” *After this question, the robot did not further confirm the delivery room, because it learned through model-based RL that navigating to office2 is relatively easy and it decided that it is more worth risking an error and having to replan than it is to ask the person another question.* The robot became confident in three dimensions of the service request (<coke, Bob, office2> in turn4) without asking about person, because of the prior knowledge (encoded in P-log) about Bob’s office.

Figure 5 shows the belief changes (in the di-

mensions of item, person, and room) as the robot interacts with a human user. The robot started with a uniform distribution in all three categories. It should be noted that, although the marginal distributions are uniform, the joint belief distribution is not, as the robot has prior knowledge such as Bob’s office is office2 and people prefer deliveries to their own offices. Demo video is not included to respect the anonymous review process.

Learning to Navigate from Navigation Tasks

In this experiment, the robot learns in the shop-room1 navigation task, and extracts the learned partial world model to the shop-room2 task. It should be noted that navigation from shop to room2 requires traveling in areas that are unnecessary in the shop-room1 task.

Figure 6 presents the results, where each data points corresponds to an average of 1000 trials. Each episode allows at most 200 (300) steps in small (large) domain. The curves are smoothed using a window of 10 episodes. The results suggest that with knowledge extraction (the dashed line) the robot learns faster than without extraction, and this performance improvement is more significant in a larger domain (the Right subfigure).

Learning to Dialog and Navigate from Navigation Tasks Robot delivering objects requires both tasks: dialog management for specifying service request (under unreliable speech recognition) and navigation for physically delivering objects (under unforeseen obstacles). Our office domain includes five rooms, two persons, and three items, resulting in 30 possible service requests. In the dialog manager, the reward function gives delivery actions a big bonus (80) if a request is fulfilled, and a big penalty (-80) otherwise.

General questions and confirming questions cost 2.0 and 1.5 respectively. In case a dialog does not end after 20 turns, the robot is forced to work on the most likely delivery. The cost/bonus/penalty values are heuristically set in this work, following guidelines based on studies from the literature on dialog agent behaviors (Zhang and Stone, 2015).

Table 1: Overall performance in delivery tasks (requiring both dialog and navigation).

	Static policy			KRR-RL		
	Reward	Fulfilled	QA Cost	Reward	Fulfilled	QA Cost
$br = 0.1$	182.07	0.851	20.86	206.21	0.932	18.73
$br = 0.5$	30.54	0.853	20.84	58.44	0.927	18.98
$br = 0.7$	-40.33	0.847	20.94	-14.50	0.905	20.56

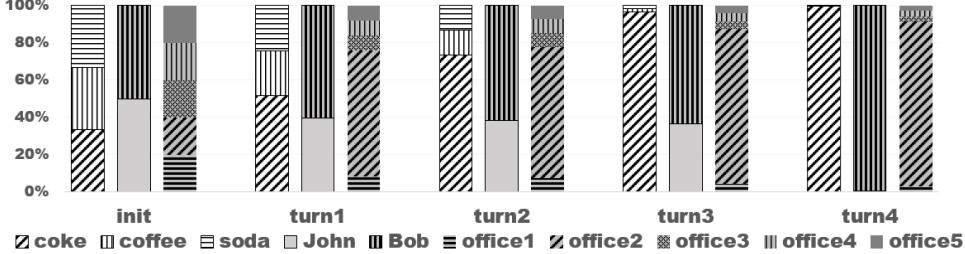


Figure 5: Belief change in three dimensions (In order from the left: Items, Persons and Offices) over five turns in a human-robot dialog . The distributions are grouped by turns (Including the initial distribution). In each turn, there are three distribution bars which means three different dimensions (In order from the left: Item, Person and Office). In order from the bottom, the values in each dimension are 1) coke, coffee and soda in Item; 2) John and Bob in Person; and 3) office1, office2, office3, office4 and office5 in Office.

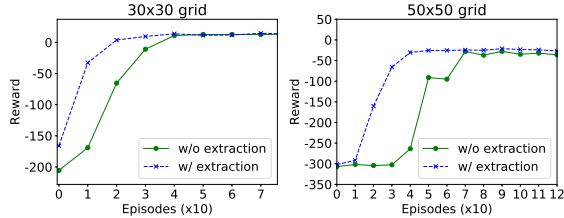


Figure 6: Navigation tasks in small (Left: 30×30 grid) and large (Right: 50×50 grid) domains. With extraction (KRR-RL in dashed line), the robot learns faster in the target navigation task.

Table 1 reports the robot’s overall performance in delivery tasks, which requires **accurate dialog** for identifying delivery tasks and **safe navigation** for object delivery. We conduct 10,000 simulation trials under each blocking rate. Without learning from RL, the robot uses a world model (outdated) that was learned under $br = 0.3$. With learning, the robot updates its world model in domains with different blocking rates. We can see, when learning is enabled, our KRR-RL framework produces higher overall reward, higher request fulfillment rate, and lower question-asking cost. *The improvement is statistically significant*, i.e., the p -values are 0.028, 0.035, and 0.049 for overall reward, when br is 0.1, 0.5, and 0.7 respectively (100 randomly selected trials with/without extraction).

Learning to Adjust Dialog Strategies from Navigation In the last experiment, we quantify the information collected in dialog in terms of entropy reduction. The hypothesis is that, using our KRR-RL framework, the dialog manager wants to collect more information before physically working on more challenging tasks. In each trial, we randomly generate a belief distribution over all possible service requests, evaluate the entropy of this belief, and record the suggested action given this belief.

We then statistically analyze the entropy values of beliefs, under which delivery actions are suggested.

Table 2: The amount of information (in terms of entropy) needed by a robot before taking delivery actions.

	Entropy (room1)		Entropy (room2)		Entropy (room5)	
	Mean (std)	Max	Mean (std)	Max	Mean (std)	Max
$br = 0.1$.274 (.090)	.419	.221 (.075)	.334	.177 (.063)	.269
$br = 0.5$.154 (.056)	.233	.111 (.044)	.176	.100 (.041)	.156
$br = 0.7$.132 (.050)	.207	.104 (.042)	.166	.100 (.041)	.156

Table 2 shows that, when br grows from 0.1 to 0.7, the means of belief entropy decreases (i.e., belief is more converged). This suggests that the robot collected more information in dialog in environments that are more challenging for navigation, which is consistent with Table 1 in the main paper. Comparing the three columns of results, we find the robot collects the most information before it delivers to room5. This is because such delivery tasks are the most difficult due to the location of room5. The results support our hypothesis that learning from navigation tasks enables the robot to adjust its information gathering strategy in dialog given tasks of different difficulties.

Adaptive Control in New Circumstances The knowledge learned through model-based RL is contributed to a knowledge base that can be used for many tasks. So our KRR-RL framework enables a robot to dynamically generate partial world models for tasks under settings that were never experienced. For example, an agent does not know the current time is morning or noon, there are two possible values for variable “time”. Consider that our agent has learned world dynamics under the times of morning and noon. Our KRR-RL framework enables the robot to reason about the two transition systems under the two settings and generate a new transition system for this “morning-or-noon” setting.

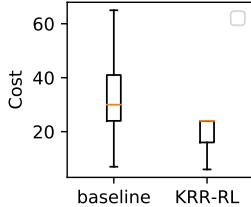


Figure 7: Adaptive behaviors under new circumstances.

Without our framework, an agent would have to randomly select one between the “morning” and “noon” policies.

To evaluate our policies dynamically constructed via KRR, we let an agent learn three controllers under three different environment settings – the navigation actions have decreasing success rates under the settings. In this experiment, the robot does not know which setting it is in (out of two that are randomly selected). The baseline does not have the KRR capability of merging knowledge learned from different settings, and can only randomly select a policy from the two (each corresponding to a setting). Experimental results show that the baseline agent achieved an average of 26.8% success rate in navigation tasks, whereas our KRR-RL agent achieved 83.8% success rate on average. Figure 7 shows the costs in a box plot (including min-max, 25%, and 75% values). Thus, KRR-RL enables a robot to effectively apply the learned knowledge to tasks under new settings.

Let us take a closer look at the “time” variable T . If \mathcal{T} is the domain of T , the RL-only baseline has to compute a total of $2^{|\mathcal{T}|}$ world models to account for all possible information about the value of T , where $2^{|\mathcal{T}|}$ is the number of subsets of \mathcal{T} . If there are N such variables, the number of world models grows exponentially to $2^{|\mathcal{T}|\cdot N}$. In comparison, the KRR-RL agent needs to compute only $|\mathcal{T}|^N$ world models, which dramatically reduces the number of parameters that must be learned through RL while retaining policy quality.

6 Conclusions and Future Work

We develop a KRR-RL framework that integrates computational paradigms of logical-probabilistic knowledge representation and reasoning (KRR), and model-based reinforcement learning (RL). Our KRR-RL agent learns world dynamics via model-based RL, and then incorporates the learned dynamics into the logical-probabilistic reasoning module, which is used for dynamic construction of efficient

run-time task-specific planning models. Experiments were conducted using a mobile robot (simulated and physical) working on delivery tasks that involve both navigation and dialog. Results suggested that the learned knowledge from RL can be represented and used for reasoning by the KRR component, enabling the robot to dynamically generate task-oriented action policies.

The integration of a KRR paradigm and model-based RL paves the way for at least the following research directions. We plan to study how to sequence source tasks to help the robot perform the best in the target task (i.e., a curriculum learning problem within the RL context (Narvekar et al., 2017)). Balancing the efficiencies between service task completion and RL is another topic for further study – currently the robot optimizes for task completions (without considering the potential knowledge learned in this process) once a task becomes available. Fundamentally, all domain variables are endogenous, because one can hardly find variables whose values are completely independent from robot actions. However, for practical reasons (such as limited computational resources), people have to limit the number of endogenous. It remains an open question of how to decide what variables should be considered as being endogenous.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under grant number U1613216. This work has taken place partly in the Autonomous Intelligent Robotics (AIR) Group at SUNY Binghamton. AIR research is supported in part by grants from the National Science Foundation (IIS-1925044), Ford Motor Company (URP Award), OPPO (Faculty Research Award), and SUNY Research Foundation. This work has taken place partly in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CPS-1739964, IIS-1724157, NRI-1925082), the Office of Naval Research (N00014-18-2243), Future of Life Institute (RFP2-000), Army Research Office (W911NF-19-2-0333), DARPA, Lockheed Martin, General Motors, and Bosch. The views and conclusions contained in this document are those of the authors alone. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

References

- Saeid Amiri, Mohammad Shokrolah Shirazi, and Shiqi Zhang. 2020. Learning and reasoning for robot sequential decision making under uncertainty. In *AAAI*.
- Nima Asgharbeygi, David Stracuzzi, and Pat Langley. 2006. Relational temporal difference learning. In *Proceedings of the 23rd international Conference on Machine learning*.
- Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2017. Hinge-loss markov random fields and probabilistic soft logic. *JMLR*, 18(1):3846–3912.
- Evgenii Balai and Michael Gelfond. 2017. Refining and generalizing p-log: Preliminary report. In *Proceedings of the 10th Workshop on ASPOCP*.
- Chitta Baral. 2010. *Knowledge Representation, Reasoning and Declarative Problem Solving*, 1st edition. Cambridge University Press, New York, NY, USA.
- Chitta Baral, Michael Gelfond, and Nelson Rushton. 2009. Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming*, 9(1):57–144.
- Ronen I Brafman and Moshe Tennenholtz. 2002. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *JMLR*.
- Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12538–12547.
- Thomas J Chermack. 2004. Improving decision-making with scenario planning. *Futures*, 36(3):295–309.
- Rohan Chitnis, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. 2018. Integrating human-provided information into belief state representation using dynamic factorization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Sašo Džeroski, Luc De Raedt, and Kurt Driessens. 2001. Relational reinforcement learning. *Machine learning*.
- Esra Erdem, Michael Gelfond, and Nicola Leone. 2016. Applications of answer set programming. *AI Magazine*, 37(3):53–68.
- Leonardo A Ferreira, Reinaldo AC Bianchi, Paulo E Santos, and Ramon Lopez de Mantaras. 2017. Answer set programming for non-stationary markov decision processes. *Applied Intelligence*, 47(4):993–1007.
- Michael Gelfond and Yulia Kahl. 2014. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach*. Cambridge University Press.
- Michael Gelfond and Vladimir Lifschitz. 1988. The Stable Model Semantics for Logic Programming. In *International Conference on Logic Programming*, pages 1070–1080.
- Marc Hanheide, Moritz Göbelbecker, Graham S Horn, et al. 2017. Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence*, 247:119–150.
- Yuqian Jiang, Shiqi Zhang, Piyush Khandelwal, and Peter Stone. 2019. Task planning in robotics: an empirical comparison of pddl- and asp-based systems. *Frontiers of Information Technology & Electronic Engineering*, 20(3):363–373.
- Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134.
- Piyush Khandelwal, Shiqi Zhang, Jivko Sinapov, et al. 2017. Bwibots: A platform for bridging the gap between ai and human-robot interaction research. *The International Journal of Robotics Research*, 36(5–7):635–659.
- Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, pages 282–293. Springer.
- Matteo Leonetti, Luca Iocchi, and Peter Stone. 2016. A synthesis of automated planning and reinforcement learning for efficient, robust decision-making. *Artificial Intelligence*.
- Dongcai Lu, Shiqi Zhang, Peter Stone, and Xiaoping Chen. 2017. Leveraging commonsense reasoning and multimodal perception for robot spoken dialog systems. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6582–6588. IEEE.
- Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. 1998. Pddl-the planning domain definition language.
- Sanmit Narvekar, Jivko Sinapov, and Peter Stone. 2017. Autonomous task sequencing for customized curriculum design in reinforcement learning. In *Proceedings of IJCAI*.
- Jean Oh, Arne Suppé, Felix Duvallet, Abdeslam Boularias, Luis Navarro-Serment, Martial Hebert, Anthony Stentz, Jerry Vinokurov, Oscar Romero, Christian Lebriere, et al. 2015. Toward mobile robots reasoning like humans. In *AAAI*.

- Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. 2018. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):1–179.
- Martin L Puterman. 1994. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
- Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. Ros: an open-source robot operating system. In *ICRA workshop on open source software*.
- Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine learning*, 62(1):107–136.
- Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Computer Vision and Pattern Recognition (CVPR)*.
- Mohan Sridharan, Michael Gelfond, Shiqi Zhang, and Jeremy Wyatt. 2019. Reba: A refinement-based architecture for knowledge representation and reasoning in robotics. *Journal of Artificial Intelligence Research*, 65:87–180.
- Mohan Sridharan, Ben Meadows, and Rocio Gomez. 2017. What can i not do? towards an architecture for reasoning about and learning affordances. In *ICAPS*.
- Richard S Sutton. 1990. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *ICML*.
- Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- Moritz Tenorth and Michael Beetz. 2013. Knowrob: A knowledge processing infrastructure for cognition-enabled robots. *The International Journal of Robotics Research*, 32(5):566–590.
- Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Nick Walker, Yuqian Jiang, Harel Yedid-sion, Justin Hart, Peter Stone, and Raymond Mooney. 2020. Jointly improving parsing and perception for natural language commands through human-robot dialog. *Journal of Artificial Intelligence Research*, 67:327–374.
- Manuela M Veloso. 2018. The increasingly fascinating opportunity for human-robot-ai interaction: The cobot mobile service robots.
- Yi Wang, Shiqi Zhang, and Joohyung Lee. 2019. Bridging commonsense reasoning and probabilistic planning via a probabilistic action language. In *the 35th International Conference on Logic Programming (ICLP)*.
- Fangkai Yang, Daoming Lyu, Bo Liu, and Steven Gustafson. 2018. PEORL: integrating symbolic planning and hierarchical reinforcement learning for robust decision-making. In *IJCAI*.
- Steve Young, Milica Gai, Blaise Thomson, and Jason D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.
- Vinicio Zambaldi, David Raposo, Adam Santoro, et al. 2018. Relational deep reinforcement learning. *arXiv preprint arXiv:1806.01830*.
- Shiqi Zhang, Piyush Khandelwal, and Peter Stone. 2017. Dynamically constructed (po) mdps for adaptive robot planning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3855–3862.
- Shiqi Zhang and Peter Stone. 2015. CORPP: Commonsense reasoning and probabilistic planning, as applied to dialog with a mobile robot. In *AAAI*.

An Attentive Listening System with Android ERICA: Comparison of Autonomous and WOZ Interactions

Koji Inoue, Divesh Lala, Kenta Yamamoto, Shizuka Nakamura,
Katsuya Takanashi, and Tatsuya Kawahara

Graduate School of Informatics, Kyoto University, Japan

[inoue, lala, yamamoto, shizuka, takanashi, kawahara]
@sap.ist.i.kyoto-u.ac.jp

Abstract

We describe an attentive listening system for the autonomous android robot ERICA. The proposed system generates several types of listener responses: backchannels, repeats, elaborating questions, assessments, generic sentimental responses, and generic responses. In this paper, we report a subjective experiment with 20 elderly people. First, we evaluated each system utterance excluding backchannels and generic responses, in an offline manner. It was found that most of the system utterances were linguistically appropriate, and they elicited positive reactions from the subjects. Furthermore, 58.2% of the responses were acknowledged as being appropriate listener responses. We also compared the proposed system with a WOZ system where a human operator was operating the robot. From the subjective evaluation, the proposed system achieved comparable scores in basic skills of attentive listening such as *encouragement to talk*, *focused on the talk*, and *actively listening*. It was also found that there is still a gap between the system and the WOZ for more sophisticated skills such as *dialogue understanding*, *showing interest*, and *empathy towards the user*.

1 Introduction

In recent years, android robots have drawn much attention from researchers and the public. Their realistic appearance is their main feature, though this requires that their behaviors are also human-like. In particular, a conversational android should not only respond correctly in terms of their dialogue content, but also exhibit phenomena such as backchanneling and correct turn taking which are present in human-human conversation. Their use as an interface for natural conversation makes them an attractive prospect for research.

Since an android which can engage in free, unstructured conversation on any topic is still a long

way off, we investigate a more limited task domain. In this paper we investigate attentive listening, and propose such a system for the android ERICA (Glas et al., 2016), who has been used for tasks such as job interviews (Inoue et al., 2019) and to investigate various conversational phenomena (Lala et al., 2017a, 2019). The extension of ERICA’s abilities to attentive listening draws from our previous research (Inoue et al., 2016; Lala et al., 2017b; Milhorat et al., 2019; Kawahara, 2019).

In attentive listening, much of the talk is from the user. The system may interject to stimulate further conversation, but does not engage in deep discussions. The advantage of this task is that the user can theoretically talk about any topic without the system needing any deep background knowledge. Such robots are useful in areas such as elderly care, where users often desire social contact but may be isolated from family (Okubo et al., 2018; Sorbello et al., 2016; Yamazaki et al., 2012). In this case, an android which provides companionship can improve the mental and emotional well-being of the elderly.

This domain provides several technical challenges. The main requirement for attentive listening is that ERICA be seen as actively listening to the conversation. The system must be able to extract the correct topic or keyword and then generate a coherent response which can stimulate further conversation, by using a variety of responses. Furthermore, while the user speaks, ERICA should exhibit human-like listening behavior which may not necessarily be verbal. Synchronizing all these features into an autonomous system is a non-trivial task, as we wish to avoid breakdowns in the conversation.

This system draws together speech recognition, natural language processing and conversational behavior. Our goal is for ERICA to be as human-like as possible in her interactions with users. We com-

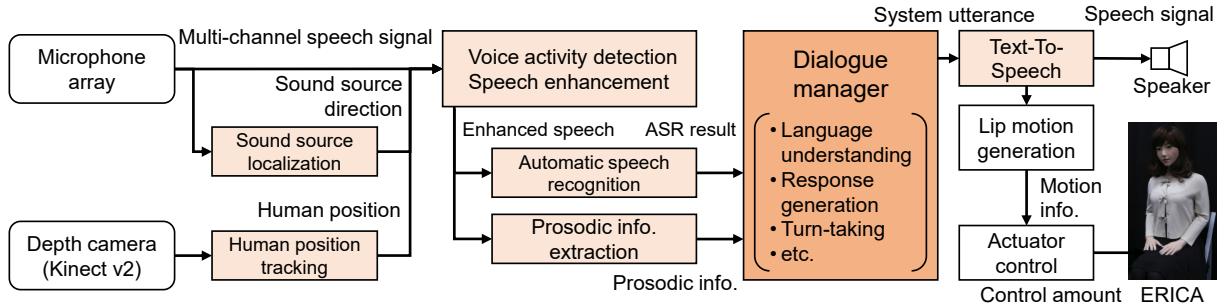


Figure 1: Architecture of a spoken dialogue system for android ERICA

pare an autonomous system to one which is controlled by a Wizard of Oz (WOZ) operator, and see how close we are to achieving human-like attentive listening.

The main contribution of this paper is a fully autonomous android attentive listener. We also report our user study which compares it to a WOZ system. The outcomes of this study will be used to guide further work in the domain of conversational androids.

2 Attentive listening system

We now describe the attentive listening system for the android robot ERICA. The whole architecture of the system is illustrated in Figure 1. First, we explain the speech processing module as the input. We then explain how to generate listener responses, followed by other necessary conversational components such as turn-taking and speech synthesis. A dialogue example can be found in Appendix A. Note that although the following processing is implemented in the Japanese language, the fundamental ideas are language-independent.

2.1 Speech processing

We use a 16-channel microphone array for automatic speech recognition (ASR) and extraction of prosodic features. Based on the multi-channel audio signals, sound source localization is conducted by multiple signal classification (MUSIC) (Ishi et al., 2016) and the direction of the audio is compared with human positions tracked by a Kinect v2 depth camera. If the sound source direction overlaps with the position of a person, enhancement of the audio is conducted and the enhanced speech is fed into an ASR system. The ASR system is implemented by an end-to-end deep neural network model (subword unit). Prosodic information including fundamental frequency (F0) and power is also extracted from the enhanced speech at 100Hz.

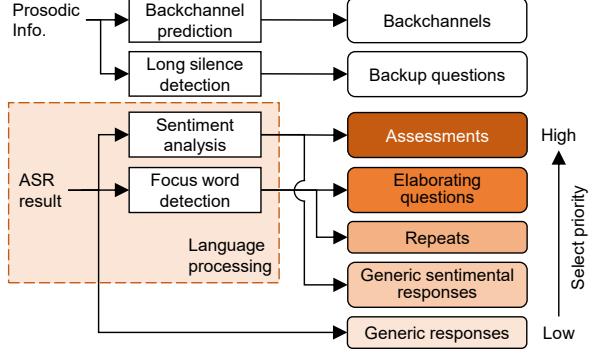


Figure 2: Architecture of listener response generation

2.2 Listener response generation

It is important for attentive listening to generate a variety of listener responses and then select an appropriate one to elicit more utterances from the user. In attentive listening, it is desirable for listener responses to express both understandings of user utterances and empathy towards users. Several attempts to implement artificial attentive listeners have been made so far (Schröder et al., 2012; DeVault et al., 2014; Han et al., 2015; Johansson et al., 2016). Our proposed attentive listening system generates backchannels, repeats, elaborating questions, assessments, generic sentimental responses, generic responses, and backup questions. Excluding backup questions, these responses do not depend on specific dialogue domains, meaning response generation is domain-independent. We now explain how the system generates each response and the selection of the final response.

Backchannels

The system generates backchannels such as “yeah” in English and “un” in Japanese. Backchannels play an important role in attentive listening in order to make users continue to talk and also to express listener attention and interest in the conversation.

There have been many works on automatic

backchannel generation, with most using prosodic features (Ward and Tsukahara, 2000; Morency et al., 2008; Ozkan et al., 2010; Truong et al., 2010; Kawahara et al., 2016). In our system, we use a logistic regression model that predicts if the system should utter a backchannel within the next 500 milliseconds. This prediction is continuously made every 100 milliseconds during the user’s turn. Input features are prosodic information consisting of the statistics (e.g., mean, maximum, minimum, and range) of the F0 and power of the user’s speech signal. This continuous prediction makes it possible to generate and utter backchannels during the utterances of the user, making the dialogue more smooth and natural. The backchannel form is determined based on a distribution observed in our attentive listening dialogue corpus, since continuous prediction of backchannel forms is much more difficult. In our system, the backchannels forms are “*un*”, “*un un*”, and “*un un un*”. In Japanese, the use of many repeating backchannels represents the stronger reaction of listeners.

Repeats

For this response, the system extracts a focus word from a user utterance and repeats it. This is expected to express understanding of the dialogue. We use a simple rule to extract a focus word, defining it as the latest noun or adjective in a user utterance. For example, if a user says “*I went to Paris to visit a museum*”, the system response would be “*A museum*”. If there are several continuous nouns, they are regarded as a compound word and are considered as the focus word. If the ASR confidence score of the focus word is lower than a threshold, the system ignores this to avoid errors caused by ASR.

Elaborating questions

If the extracted focus word can be extended to elicit more dialogue about a topic, an elaborating question is generated. Generating the proper elaborating question not only extends the dialogue but also expresses deeper understanding of the dialogue. The system generates a question by concatenating the focus word with interrogatives such as *which*, *when*, and *what*. In total, we use 11 types of interrogatives as candidates. For example, if a user says “*I went to Paris to visit a museum*”, the focus word would be “*a museum*” and the elaborating question would be “*Which museum?*”. To select the proper interrogative, the system refers to bigram probabilities

of all possible pairs and selects the interrogative that has the highest probability with the focus word. The probability must also be higher than a fixed threshold. If all bigram probabilities are lower than the threshold, no elaborating question is generated. Bigram probabilities are calculated in advance by using large-scale language corpora. In our case, we use the balanced corpus of contemporary written Japanese (BCCWJ)¹.

Assessments

If a user utterance contains a positive or negative sentiment, the system utterance reflects this by using an assessment response. This emotional response is expected to express empathy towards users. We first conduct sentiment analysis of a user utterance by using two kinds of Japanese sentiment dictionaries²³ where positive and negative words (phrases) are defined. Since sentiment responses strongly depend on the dialogue context, the dictionaries should focus on precision rather than the coverage. Therefore, we ensure that words in the dictionary are robust in terms of correctly determining the sentiment, even though the number of words is comparatively small. The system determines the sentiment of the current user utterance as positive, negative, or neutral by referring to a sentiment score of each word. If the utterance contains both positive and negative scores, the majority sentiment is used. Similar to focus word detection, if the ASR score of a word is lower than a threshold, then the corresponding sentiment score is ignored. The assessment response is selected according to the estimated sentiment of the user utterance. A positive sentiment leads to system responses such as “*That is good* (いいですね)” or “*That is nice* (素敵ですね)”, and negative sentiment leads to responses such as “*That is bad* (残念でしたね)” or “*That is hard* (大変ですね)”. If no sentimental words were found, this module does not output any responses.

Generic responses

The system prepares generic responses because the above-mentioned responses are not always generated. Generic responses are “*I see* (そうですか)” or “*I got it* (なるほど)”. These responses can be used for any dialogue context. If the user utterance

¹https://pj.ninjal.ac.jp/corpus_center/bccwj/

²<https://www.gsk.or.jp/catalog/gsk2011-c/>

³<http://www.jnlp.org/SNOW/D18>

is short, the system also uses a short generic response such “Yes (はい)” to avoid system barge-in.

Generic sentimental responses

The system also generates another type of generic response according to the sentiment of user utterances. For this response type, we use a different sentiment dictionary (Kobayashi et al., 2004) that covers a wider range of words but also expressions that might have opposing sentiments depending on the dialogue context. We designed generic sentimental responses where the surface form is the same as those of the generic responses but the prosodic pattern changes according to the estimated sentiment. By generating these responses, the system can reduce the risk of a linguistic breakdown (since they don’t explicitly use an emotional linguistic response) but also express empathy towards users through prosody.

Backup questions

If a user stays silent longer than a specific amount of time (four seconds in the current system), the system generates one of several backup questions. The questions are defined in advance according to the theme of the user’s talk. For example, if the theme is *traveling*, a backup question is “*Where did you go after that?*”.

Response selection

Since the above-mentioned modules generate several response candidates, it is important for this attentive listening system to select the proper one among them. Backchannels are uttered during the user’s turn, so this module works independently from the others. Backup questions are triggered by a longer pause so that this module is also independent. For the other response types, we designed a priority system as depicted in Figure 2. The system will respond using the highest priority response type which can be generated given the user’s utterance. The priority order is based on how likely it is to generate the response type. For example, assessments use a limited dictionary so it is less likely that a user utterance will generate these kinds of responses than the other response types. On the other hand, generic responses can be used without any modeling so will inevitably be required if no other valid response type can be generated..

2.3 Turn taking

Turn-taking is an important feature of attentive listening, since we want to strike a balance between

reducing barge-in from the system and allowing the system to interject during the dialogue. A simple approach in a basic spoken dialogue system is to wait until the user has been silent for a set period of time before the system can take the turn. However, this requires fine tuning and is usually inflexible.

We implement a machine learning turn-taking model that uses the ASR result as an input and supplement this with an finite-state turn-taking machine (FSTTM) as used in previous works (Raux and Eskenazi, 2009; Lala et al., 2018) to determine how much silence from the user should elapse before the turn switches to the system. Utterances with a high probability of being end-of-turn are responded to quickly, while the system will wait longer if the user says utterances such as fillers or hesitations.

2.4 Speech synthesis

The speech synthesis in the system has been designed for android ERICA ⁴. Since the vocabulary of backchannels, assessments, generic responses are fixed, we recorded natural speech voices and directly play them instead of using real-time synthesis. This is also because it is still difficult to synthesize these kinds of dialogue-specific utterances with a variety of prosodic patterns using current synthesis techniques. For other responses such as repeats and elaborating questions, we can use real-time synthesis because the focus word depends on user utterances.

3 Dialogue experiment

We conducted a dialogue experiment to evaluate how the proposed system works with elderly people as subjects. We also investigated how the system compared when compared to attentive listening with a WOZ operator.

3.1 Conditions

We recruited 20 Japanese elderly people (between 70-90 years old). A snapshot of this dialogue experiment is shown in Figure 3. Subjects were asked to talk to the android robot about two topics: “*Most memorable travel experience*” and “*Delicious food you recently ate*”.

We prepared two types of systems: autonomous and WOZ. The autonomous system corresponds to the proposed attentive listening system. The WOZ

⁴<https://voicetext.jp/news/product/151023/>

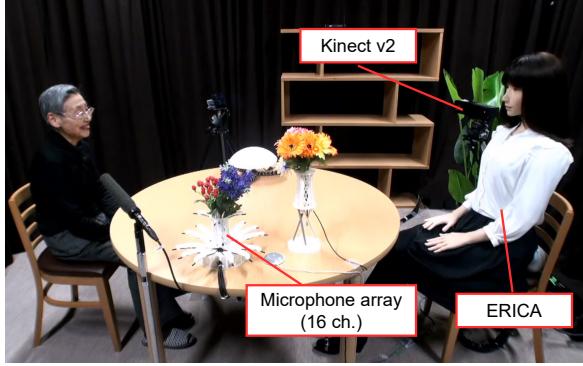


Figure 3: Snapshot of dialogue experiment

system is the case where the robot was operated by a human operator. Each subject talked with one of the systems about one dialogue topic in one condition and then did the same with the other condition. The order of the systems and topics were randomized among the subjects. After they had talked in one of the conditions, we asked them to evaluate the system individually. Note that the average word error rate (WER) of the ASR in the autonomous system was 33.8%, which suggests that the ASR with elderly people is more difficult than those with younger people. The current dialogue experiment explores what level of dialogue can be realized in this challenging situation.

The WOZ operators were two amateur actresses and each of them attended to each dialogue. The operator was asked to use the same set of listener responses as our autonomous system but also asked to properly select the timing and type of the proper response by herself. The operator spoke directly into a microphone and the voice was played via a speaker nearby ERICA, so this dialogue seemed to be natural spoken dialogue. Although the operators' voices were different from those of the speech synthesis, we asked the operators to imitate ERICA's synthesized voice as much as possible.

The dialogue time was set at seven minutes for each conversation. Our experimental trials determined this time as the longest where the autonomous system can continue with the dialogue before it becomes too repetitive. In the autonomous system, when the dialogue time passes seven minutes and the system takes the turn, the system says a fixed phrase to end the dialogue. The same rule was imposed on the WOZ operators.

Table 1: Total frequencies (per session) of each response type in the proposed system

Response type	Frequency
Backchannels	1,601 (80.1)
Repeats	90 (4.5)
Elaborating questions	16 (0.8)
Assessments	45 (2.3)
Generic sentimental responses	62 (3.1)
Generic responses	325 (16.3)
Backup questions	12 (0.6)

3.2 Evaluation on utterances of autonomous system

At first, we analyzed the distribution of response types uttered by the autonomous system. The distribution is reported in Table 1. It can be seen that all the response types could be generated in the autonomous system. As we expected, many system utterances consisted of backchannels and generic responses. On average, repeats were uttered about 4-5 times per dialogue, and elaborating questions were uttered just once, assessments were uttered about twice and generic sentimental responses were uttered about three times. This distribution will also be compared with those of the WOZ system in the later analysis.

We also evaluated each system response manually in an offline manner. In this evaluation, three criteria were considered: (1) *no error*, (2) *reaction*, and (3) *appropriate*.

The first criterion, *no error*, validates the extracted focus word. If the uttered focus word is contained in the context of user utterances and is not strange (e.g., unused words in the human dialogue), the system response was marked as accepted, otherwise rejected. The target types of responses were repeats and elaborating questions. This criterion was used to detect linguistic errors of system utterances caused by ASR or language processing errors.

The second criterion, *reaction*, focuses on the subjects' reactions after the system utterances. The reaction of the subjects to system utterances is also important for evaluation. The target types of responses were repeats, elaborating questions, and assessments. For repeats and assessments, if a subject said a positive reaction such as “Yeah” after a system response, the response was accepted. For elaborating questions, if a subject answered the system question the question was accepted.

Table 2: Offline evaluation on each system utterance (*No error* means the correctness of the language processing on the surface level. *Reaction* means the positive reaction of the user after the system utterance. *Appropriate* represents the appropriateness of the system utterance as effective listener responses.)

Response type	(1) No error		(2) Reaction		(3) Appropriate	
	YES	NO	YES	NO	YES	NO
Repeats	83	7	79	11	57	33
Elaborating questions	16	0	13	3	11	5
Assessments	-	-	32	13	31	14
Generic sentimental responses	-	-	-	-	25	37
Total	99 (93.4%)	7 (6.6%)	124 (82.1%)	27 (17.9%)	124 (58.2%)	89 (41.8%)

The third criterion, *appropriate*, validates appropriateness as listener responses. The target types of responses were repeats, elaborating questions, assessments, and generic sentimental responses. Since this criterion seems to be the most important but subjective, we defined the detailed criterion for each type of response as follows. For repeats, if there is another focus word that is clearly more adequate than the used one or there is no focus word in the dialogue context, the repeat response is rejected. For elaborating questions, the same criterion of repeats is firstly applied. Additionally, if the combination of the interrogative and focus word is strange or the elaborating question itself is strange, the question is rejected. For assessments, if the estimated sentiment is the opposite or the sentiment of a user utterance is neutral rather than positive or negative, the assessment response is rejected. For generic sentimental responses, if the estimated sentiment is the opposite or the sentiment of a user utterance is clearly neutral, the response is rejected. Although this criterion was expected to be the most strict, most of the utterances of the WOZ system were accepted as we observed the dialogue. Therefore, this criterion is needed to reveal the future work.

We conducted a manual evaluation of the above three criteria by two evaluators who checked their results with each other. The evaluation result is reported in Table 2. For the first criterion, *no error*, most of the responses were accepted. This means that uttered responses did not elicit linguistic errors. For the second criterion, *reaction*, about 80 percent of the system responses elicited positive reactions from the subjects. This result is good because it shows that many utterances were understood by the subjects. For the third criterion, *appropriate*, more than half of the responses were accepted. Excluding generic sentimental responses, two thirds

of responses were accepted. In summary, most of the uttered responses were acceptable on the surface level and were also reacted to by the subjects. On the other hand, nearly half of them left room for improvement by considering a strict level of appropriateness for attentive listening.

We introduce some real examples on the evaluation of the third criterion. The dialogues were done in Japanese so the following is the English translation together with the original Japanese utterances (not the ASR results). E and S represent ERICA and subject utterances respectively.

The following two examples are accepted repeats.

S: This talk is about something 35 years ago.
(今から35年くらい前の話ですね。)

E: 35 years.
(35年ですか。)

S: I got on the plane, but we had to wait for one hour.
(飛行機に乗ったんですけど、その飛行機の中で一時間ほど待たせられて。)

E: One hour. (一時間ですか。)

The following example was rejected.

S: The travel to there still remains in my heart.
(そこへ行った旅行がとても心に残っています。)

E: Heart. (心ですか。)

In this case, there is no focus word in the user utterance so assessments or generic responses would be more appropriate.

The following examples are accepted elaborating questions.

S: Considering side menus, she makes delicious cakes for me.

Table 3: Average scores (standard deviations) on subjective evaluation and t-test results ($n = 20$)

	Question item	Autonomous	WOZ	<i>p</i> -value
(Robot behaviors)				
Q1	The words uttered by the robot were natural	5.0 (1.6)	5.9 (0.9)	.003 **
Q2	The robot responded with good timing	4.8 (1.4)	5.6 (1.3)	.022 *
Q3	The robot responded diligently	5.5 (0.7)	5.8 (1.0)	.005 **
Q4	The robot's reaction was like a human's	4.4 (1.3)	5.2 (1.3)	.008 **
Q5	The robot's reaction adequately encouraged my talk	5.0 (1.4)	5.2 (0.9)	.359
Q6	The frequency of the robot's reaction was adequate	5.1 (1.1)	5.4 (1.1)	.232
(Impression on the robot)				
Q7	I want to talk with the robot again	4.6 (1.3)	5.4 (1.5)	.005 **
Q8	The robot was easy to talk with	4.9 (1.3)	5.4 (1.2)	.116
Q9	I felt the robot is kind	4.7 (1.4)	5.6 (1.2)	.004 **
Q10	The robot listened to the talk seriously	5.6 (1.1)	6.0 (1.1)	.072 +
Q11	The robot listened to the talk with focus	5.6 (1.1)	5.7 (1.1)	.681
Q12	The robot listened to the talk actively	5.4 (1.3)	5.6 (1.1)	.385
Q13	The robot understood the talk	5.0 (1.1)	5.9 (1.4)	.002 **
Q14	The robot showed interest for the talk	5.2 (1.3)	5.8 (1.2)	.028 *
Q15	The robot showed empathy towards me	5.1 (1.4)	5.8 (1.0)	.015 *
Q16	I think the robot was being operated by a human	3.3 (1.3)	2.9 (1.1)	.286
Q17	The robot was good at taking turns	4.5 (1.1)	4.8 (1.3)	.209
(Impression on the dialogue)				
Q18	I was satisfied with the dialogue	4.6 (1.5)	5.3 (1.5)	.012 *
Q19	The exchange in the dialogue was smooth	4.6 (1.4)	5.3 (1.4)	.002 **

(+ $p < .1$, * $p < .05$, ** $p < .01$)

(副食物を考えて、おいしいケーキを作ってくれます。)

E: What cakes?
(どんなケーキですか?)

S: I stayed in Shilla hotel, a big hotel in Seoul.
(ソウルの新羅ホテルってね、大きなホテルでそこで泊まりましたけどね。)

E: What hotel?
(どんなホテルですか?)

The following is a rejected sample.

S: I put radish inside a pot.
(大根の中だけお鍋に入れて。)
E: Which pot?
(どのお鍋ですか?)

In this case, the pot seems not to be the focus word in the dialogue context and should not be elaborated upon as the next dialogue topic.

The following example is an accepted assessment.

S: I ate many dishes and visited many places, so I really enjoyed the time.
(いろんなものを食べたり見たりしてそれは楽しい時間が過ごせました。)

E: That is good. (いいですね。)

The following is a rejected sample.

S: She gets angry if I do not start to eat while it is still hot.
(温かいうちに食べないと怒ります。)
E: That is bad. (残念でしたね。)

In this case, although there is actually negative word *angry*, the user was just introducing his family and the sentiment of this utterance seems to be neutral.

3.3 Comparison with WOZ system

We compared the autonomous system with the WOZ system in order to investigate how much it could match that of a human. Table 4 reports the average scores on the subjective evaluation. The question items consist of three categories: robot behavior, impression of the robot, and impression of the dialogue. The subjects evaluated each question item in the 7-point scale from 1 to 7. Overall the evaluated scores were higher than the middle point (4), meaning the autonomous system was given a positive evaluation.

Table 4: Average values (standard deviation) on the analysis of the subjects' utterances and results of t-test

Analyzed behavior	Autonomous	WOZ	p-value
Subject's utterance time / min.	38.3 (5.5)	37.5 (5.9)	.287
Number of uttered words / min.	107.5 (19.1)	112.0 (23.1)	.177
Unique number of uttered words / min.	29.0 (4.4)	32.6 (5.1)	.003 **
Number of uttered content words / min.	53.2 (9.8)	55.6 (12.3)	.220
Unique number of uttered content words / min.	23.3 (4.1)	26.3 (4.4)	.008 **

(** $p < .01$)

We conducted a paired t-test on each question item between the autonomous and WOZ systems ($n=20$). In the first category, significant differences were observed from Q1 to Q4, but no significant differences were observed in Q5 and Q6. This means that the subjects could perceive the difference in ERICA's utterances between the autonomous and WOZ systems. However, from Q5, there was no clear difference in encouraging the subjects' talk. From Q6, the frequency of listener responses was natural even in the autonomous system.

In the second category, significant differences were observed in questions Q7, Q9, Q13, Q14, and Q15. Interestingly, although there is no significant difference in the listening attitude (Q10, Q11, Q12), significant differences were observed in the items of dialogue understanding (Q13), showing interest (Q14), and empathy towards the user (Q15). This means that the proposed system achieved basic listening skills as well as a human operator, but there is room for improvement on sophisticated skills.

In the third category, impression on the dialogue, scores of both items had significant differences. It is expected that improving the above-mentioned items (e.g., Q13, Q14, Q15) leads to improvement on the impression of this dialogue.

We also analyzed the subjects' utterances as reported in Table 4. These measurements provide objective scores on how much the systems encouraged the subjects' talk. To count the number of words, word segmentation is required in the Japanese language so we used a public tool⁵. Content words were defined as nouns, verbs, adjectives, adverbs, and conjunctions. From our result, the numbers of uttered words and content words were not different between the autonomous and WOZ systems. Interestingly, the unique numbers of uttered words and content words were significantly different, meaning the human operators could elicit a wider variety of lexical content than the autonomous system.

⁵<https://taku910.github.io/mecab>

Table 5: Total frequencies (per session) of each response type uttered by the WOZ operators

Response type	Frequency
Backchannels	1,573 (78.7)
Repeats	48 (2.4)
Elaborating questions	13 (0.7)
Assessments	126 (6.3)
Generic responses	259 (13.0)
Backup questions	3 (0.2)
Others	28 (1.4)

Finally, we analyzed the distribution of listener responses in the WOZ system, as reported in Table 5. Note that generic sentimental responses are included in generic responses because it is hard to distinguish them when they are said by the WOZ operator. Compared with the case of the autonomous system reported in Table 1, assessments were used more by the human operators. Furthermore, the number of repeats was smaller in the WOZ system. This difference can be reflected in the design of the priority order of response types shown in Figure 2.

4 Conclusion

In this work, we described the implementation of an attentive listening system for the android ERICA. We discussed details of the system including how it generates various response types based on the user's utterance. Furthermore, we conducted a user study to investigate the performance of the system compared to one operated by a WOZ operator. We found that the proposed system could match the WOZ system in terms of perceived basic listening skills, but was outperformed by the human for more sophisticated skills such as displaying empathy.

Acknowledgments

This work was supported by JST ERATO Grant number JPMJER1401 and JSPS KAKENHI Grant number JP19H05691.

References

- David DeVault, Ron Artstein, Grace Benn, Teresa Dey, Ed Fast, Alesia Gainer, Kallirroi Georgila, Jon Gratch, Arno Hartholt, Margaux Lhommet, Gale Lucas, Stacy Marsella, Fabrizio Morbini, Angela Nazarian, Stefan Scherer, Giota Stratou, Apar Suri, David Traum, Rachel Wood, Yuyu Xu, Albert Rizzo, and Louis P. Morency. 2014. SimSensei Kiosk: A virtual human interviewer for healthcare decision support. In *International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 1061–1068.
- Dylan F. Glas, Takashi Minato, Carlos T. Ishi, Tatsuya Kawahara, and Hiroshi Ishiguro. 2016. ERICA: The ERATO intelligent conversational android. In *International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 22–29.
- Sangdo Han, Jeesoo Bang, Seonghan Ryu, and Gary Geunbae Lee. 2015. Exploiting knowledge base to generate responses for natural language dialog listening agents. In *Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 129–133.
- Koji Inoue, Kohei Hara, Divesh Lala, Shizuka Nakamura, Katsuya Takanashi, and Tatsuya Kawahara. 2019. A job interview dialogue system with autonomous android ERICA. In *International Workshop on Spoken Dialog System Technology (IWSDS)*.
- Koji Inoue, Pierrick Milhorat, Divesh Lala, Tianyu Zhao, and Tatsuya Kawahara. 2016. Talking with ERICA, an autonomous android. In *Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 212–215.
- Crlos T. Ishi, Chaoran Liu, Jani Even, and Norihiro Hagita. 2016. Hearing support system using environment sensor network. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1275–1280.
- Martin Johansson, Tatsuro Hori, Gabriel Skantze, Anja Höthker, and Joakim Gustafson. 2016. Making turn-taking decisions for an active listening robot for memory training. In *International Conference on Social Robotics (ICSR)*, pages 940–949.
- Tatsuya Kawahara. 2019. Spoken dialogue system for a human-like conversational robot ERICA. In *International Workshop on Spoken Dialog System Technology (IWSDS)*.
- Tatsuya Kawahara, Takashi Yamaguchi, Koji Inoue, Katsuya Takanashi, and Nigel G. Ward. 2016. Prediction and generation of backchannel form for attentive listening systems. In *INTERSPEECH*, pages 2890–2894.
- Nozomi Kobayashi, Kentaro Inui, Yuji Matsumoto, Kenji Tateishi, and Toshikazu Fukushima. 2004. Collecting evaluative expressions for opinion extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 596–605.
- Divesh Lala, Koji Inoue, and Tatsuya Kawahara. 2018. Evaluation of real-time deep learning turn-taking models for multiple dialogue scenarios. In *International Conference on Multimodal Interaction (ICMI)*, pages 78–86.
- Divesh Lala, Koji Inoue, and Tatsuya Kawahara. 2019. Smooth turn-taking by a robot using an online continuous model to generate turn-taking cues. In *International Conference on Multimodal Interaction (ICMI)*, pages 226–234.
- Divesh Lala, Koji Inoue, Pierrick Milhorat, and Tatsuya Kawahara. 2017a. Detection of social signals for recognizing engagement in human-robot interaction. In *AAAI Fall Symposium on Natural Communication for Human-Robot Collaboration*.
- Divesh Lala, Pierrick Milhorat, Koji Inoue, Masanari Ishida, Katsuya Takanashi, and Tatsuya Kawahara. 2017b. Attentive listening system with backchanneling, response generation and flexible turn-taking. In *Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, pages 127–136.
- Pierrick Milhorat, Divesh Lala, Koji Inoue, Tianyu Zhao, Masanari Ishida, Katsuya Takanashi, Shizuka Nakamura, and Tatsuya Kawahara. 2019. A conversational dialogue manager for the humanoid robot ERICA. In *Advanced Social Interaction with Agents*, pages 119–131. Springer.
- Louis P. Morency, Iwan De Kok, and Jonathan Gratch. 2008. Predicting listener backchannels: A probabilistic multimodal approach. In *International Conference on Intelligent Virtual Agents (IVA)*, pages 176–190.
- Masataka Okubo, Hidenobu Sumioka, Soheil Keshmiri, and Hiroshi Ishiguro. 2018. Intimate touch conversation through teleoperated android: Toward enhancement of interpersonal closeness in elderly people. In *International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 23–28.
- Derya Ozkan, Kenji Sagae, and Louis P. Morency. 2010. Latent mixture of discriminative experts for multimodal prediction modeling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 860–868.
- Antoine Raux and Maxine Eskenazi. 2009. A finite-state turn-taking model for spoken dialog systems. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 629–637.
- Marc Schröder, Elisabetta Bevacqua, Roddy Cowie, Florian Eyben, Hatice Gunes, Dirk Heylen, Mark ter Maat, Gary McKeown, Sathish Pammi, Maja Pantic, Catherine Pelachaud, Björn Schuller, Etienne de Sevin, Michel Valstar, and Martin Wöllmer. 2012. Building autonomous sensitive artificial listeners. *IEEE Transaction on Affective Computing*, 3(2):165–183.

Rosario Sorbello, Antonio Chella, Marcello Giardina, Shuichi Nishio, and Hiroshi Ishiguro. 2016. An architecture for telenoid robot as empathic conversational android companion for elderly people. In *Intelligent Autonomous Systems 13*, pages 939–953. Springer.

Khiет P Truong, Ronald Poppe, and Dirk Heylen. 2010. A rule-based backchannel prediction model using pitch and pause information. In *INTERSPEECH*, pages 3058–3061.

Nigel G. Ward and Wataru Tsukahara. 2000. Prosodic features which cue back-channel responses in English and Japanese. *Journal of pragmatics*, 32(8):1177–1207.

Ryuji Yamazaki, Shuichi Nishio, Hiroshi Ishiguro, Marco Nørskov, Nobu Ishiguro, and Giuseppe Balistreri. 2012. Social acceptance of a teleoperated android: field study on elderly’s engagement with an embodied communication medium in Denmark. In *International Conference on Social Robotics (ICSR)*, pages 428–437.

A Appendix: Dialogue example

A dialogue example of the proposed system is as follows, together with original Japanese sentences. E and U represent ERICA (system) and user utterances respectively. Note that this is a written example for illustrative purposes, and backchannels are omitted in this example.

E1: Please tell me your most memorable travel.
(今までで最も印象に残っている旅行についてお話を聞かせてください。)

U1: Last year, I went to Kyoto to visit a park.
(去年、京都にある公園へ行きました。)

E2: Which park? (elaborating question)
(どの公園ですか？)

U2: It is Maruyama park.
(円山公園です。)

E3: Maruyama park (repeating)
(円山公園ですか。)

U3: There is a famous cherry blossom.
(そこには、有名な桜があります。)

E4: A famous cherry blossom (repeating)
(有名な桜ですか。)

U4: Then I watched the cherry blossom.
(それで、その桜を見ました。)

E5: I see. (generic response)
(そうなんですね。)

U5: It was really beautiful and I was moved.
(本当に綺麗で感動しました。)

E6: That is nice (assessment)
(いいですね。)

– long silence –

E7: Where did you go after that? (backup question)

(そのあとはどこかへ行きましたか？)

U7: Yes. I went shopping in a mall.

(はい。ショッピングモールへ買い物に行きました。)

E8: Where is the mall? (elaborating question)

(どこのショッピングモールですか？)

U8: It is in the middle of the city and it was very crowded.

(街の中心地にあるもので、とても混雑していました。)

E9: That’s hard. (assessment)

(大変ですね。)

A Spoken Dialogue System for Spatial Question Answering in a Physical Blocks World

Georgiy Platonov

Benjamin Kane

Aaron Gindi

Lenhart K. Schubert

Department of Computer Science

University of Rochester

{gplatono, bkane2, agindi, schubert}@cs.rochester.edu

Abstract

A physical blocks world, despite its relative simplicity, requires (in fully interactive form) a rich set of functional capabilities, ranging from vision to natural language understanding. In this work we tackle spatial question answering in a holistic way, using a vision system, speech input and output mediated by an animated avatar, a dialogue system that robustly interprets spatial queries, and a constraint solver that derives answers based on 3-D spatial modeling. The contributions of this work include a semantic parser that maps spatial questions into logical forms consistent with a general approach to meaning representation, a dialogue manager based on a schema representation, and a constraint solver for spatial questions that provides answers in agreement with human perception. These and other components are integrated into a multi-modal human-computer interaction pipeline.

1 Introduction

Despite impressive recent advances of AI in specific, narrow tasks, such as object recognition, natural language parsing and machine translation, game playing, etc., there is still a shortage of multimodal interactive systems capable of performing high-level tasks requiring understanding and reasoning. The blocks world domain, despite its relative simplicity, motivates implementation of a diverse range of capabilities in a virtual interactive agent aware of physical blocks on a table, including visual scene analysis, spatial reasoning, planning, learning of new concepts, dialogue management and voice interaction, and more. In this work, we describe an end-to-end system that integrates several such components in order to perform a simple task of spatial question answering about block configurations. Our goal is dialogue-based question answering about spatial configurations of blocks on a table,

in a way that reflects people’s intuitive understanding of prepositional spatial relations. The system is able to answer questions such as “*Which blocks are touching some red block?*”, “*Is the X block clear?*”, “*Where is the Y block?*”, etc. (where X and Y are unique block labels). Distinctive features of our work: (1) it is an end-to-end system using computer vision and spoken dialogue with an on-screen virtual human; (2) it did not require a large training corpus, only a modest development corpus using naturally posed spatial questions by a few participants; (3) it derives and relies on a 3D representation of the scene; (4) it models spatial relations realistically in terms of meaningful geometric and contextual constraints.

2 Related Work

Early studies featuring the blocks world include (Winograd, 1972) and (Fahlman, 1974), both of which maintained symbolic memory of blocks-world states. They demonstrated impressive planning capabilities, but their worlds were simulated, interaction was text-based, and they lacked a realistic understanding of spatial relations. Modern efforts in blocks worlds include work by Perera et al. (Perera et al., 2018), which is focused on learning spatial concepts (such as staircases, towers, etc.) based on verbally-conveyed structural constraints, e.g., “*The height is at most 3*”, as well as explicit examples and counterexamples, given by the user. Bisk et al. (Bisk et al., 2018) use deep learning to transduce verbal instructions into block displacements in a simulated environment. Some deep learning based studies achieve near-perfect scores on the CLEVR question answering dataset (Kottur et al., 2019; Mao et al., 2019). Common limitation of these approaches is reliance on unrealistically simple spatial models and domain-specific language formalisms, and in relation to our

work, there is no question answering functionality or episodic memory. Our work is inspired by the psychologically and linguistically oriented studies (Garrod et al., 1999; Herskovits, 1985; Tyler and Evans, 2003). Studies of human judgements of spatial relations show that no crisp, qualitative models can do justice to those judgments. The study (Platonov and Schubert, 2018) explored computational models for prepositions using imagistic modeling, akin to the current work. Another study (Bigelow et al., 2015) applied imagistic approach to a story understanding task and employed Blender to create 3D scenes and reason about the relative configuration and visibility of objects in the scene.

3 Blocks World System Overview

Fig. 1, 2 depict our physical blocks world (consisting of a square table with several cubical blocks, two Kinect sensors and a display) and the system’s software architecture¹. The blocks are color-coded as green, red, or blue, and marked with corporate logos, serving as unique identifiers. The system uses audio-visual I/O: the block tracking module periodically updates the block positioning information by reading from the Kinect cameras and an interactive avatar, David, is used for human-machine communication. The block arrangement is modeled as a 3D scene in Blender, which acts as system’s “mental image” of the state of the world. Google’s Cloud Speech-To-Text API is used for the automatic speech recognition. Its output is processed to fix some common mistakes in the transcripts. The avatar is capable of vocalizing the text and displaying facial expressions, making the flow of conversation more natural than with textual I/O. The spatial component module together with the constraint solver is responsible for analyzing the block configuration with respect to the conditions implicit in the user’s utterance. The Eta dialogue manager is responsible for *unscoped logical form* (ULF) generation (see subsection below) and controlling the dialogue flow and transition between phases, such as greeting, ending the session, etc.

3.1 Eta Dialogue Manager and Semantic Parser

Eta is a dialogue manager (DM) designed to follow a modifiable dialogue schema, specified using

¹The code for Eta and the rest of the system can be found at <https://github.com/bkane2/eta> and <https://github.com/gplatono/BlocksWorld>



Figure 1: The blocks world apparatus setup.

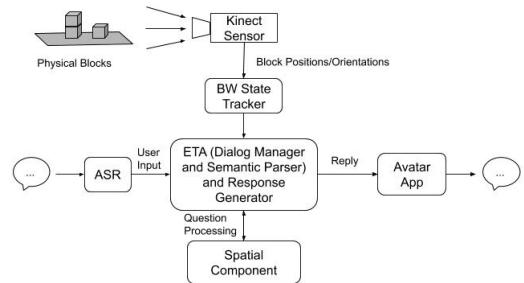


Figure 2: The blocks world dialogue pipeline. The arrows indicate the direction of interaction between the modules.

a flexible and expressive schema language. The main contents of a dialogue schema are logical formulas with open variables describing successive steps (events) expected in the course of the interaction, typically speech acts by the system or the user. These are either realized directly as actions (with variables instantiated to particular entities), or, in the case of abstract actions, expanded into subschemas for further processing as the interaction proceeds.² A key mechanism used in the course of instantiating schema steps, including interpretation of user inputs, is *hierarchical pattern transduction*. Transduction hierarchies specify patterns at their nodes, with branches from a node providing alternative continuations as a hierarchical match proceeds. Terminal nodes provide result templates, or specify a subschema, a subordinate transduction tree, or some other result. The patterns are simple template-like ones that look for particular words or word features, and allow for “match-anything”, length-bounded word spans.

Eta extends the approach implemented in the

²Intended actions obviated by earlier events may be deleted.

LISSA system (Razavi et al., 2016, 2017). Like the latter, Eta derives English *gist clauses* by pre-processing the input. However, it is only used for handling casual aspects of dialogue such as greetings, and for “tidying up” some inputs in preparation for further processing. Additional regularization is done with a limited coreference module, which can resolve anaphora and referring expressions such as “it”, “that block”, etc., by detecting and storing discourse entities in context and employing recency and syntactic salience heuristics. This allows Eta to answer some simple follow-up questions like “Where is it now?” From the tidied-up inputs, Eta derives an *unscoped logical form* (ULF) (Kim and Schubert, 2019). ULF is closely related to the logical syntax used in schemas – it is a preliminary form of that syntax, when mapping English to logic. ULF differs from analogs, e.g., AMR, in that it is close to the surface form of English, covers a richer set of semantic phenomena, and does so in a type-consistent way. For example, ULF for the sentence “Which blocks are on two other blocks?” will be (((Which.d (plur block.n)) ((pres be.v) (on.p (two.d (other.a (plur block.n))))))) ?. Resulting ULF retains much of the surface structure, but uses semantic typing and adds operators to indicate plurality, tense, aspect, and other linguistic phenomena. We introduced recursion into hierarchical transduction trees to enable ULF derivation.

3.2 Spatial Relations

We model spatial relations as probabilistic predicates, using 3-D imagistic scene representations. Each predicate is composed of several factors, which represent basic relations that correlate with higher level spatial relation, e.g., if A is on top of B, then (usually) A is above B, and A is in contact with B. Thus, “above-ness” and contact serve as (some of the) factors used in determining “on-ness”. After all the contributing factors are computed, their values are combined, e.g., by taking a linear combination, maximal value, etc., depending on the relation. Examples of factors are the scaled distance between centroids, frame size (size of the scene in context, important for judging relative distances), contact, support, certain shapes or types, proportion of the overlap of objects’ projections onto the visual plane (for deictic sense of certain relations), etc. Not all factors potentially influencing a relation are relevant in a given situation, so we check various combinations of them

that correspond to different usage patterns.

Some factors involve scene statistics, e.g., when determining nearness of A and B , the distribution of other objects is important. First, raw context-independent value is computed, which is then scaled up or down, depending on the raw scores for other objects, e.g., let $\text{near_raw}(A, B) = 0.55$. If B is the closest object to A , i.e., $\text{near_raw}(C, A) < 0.55, \forall C(C \neq B)$, we perceive B as the best near-object of A . Thus, the final score $\text{near}(A, B)$ will be boosted by a small (variable) amount.

4 Evaluation

We enlisted 5 volunteers, including native and non-native English speakers. The participants were instructed to ask spatial questions of the general type supported by the system, but without restriction on wording; before their first session they were shown a short demonstration of the expected kind of interaction with the system, including question-answer exchanges. Each session started with the blocks positioned in a row at the front of the table. The participants were instructed to move the blocks arbitrarily to test the robustness and consistency of the spatial models. During each session they were requested to ask 40-50 questions and mark system’s answers as correct, partially correct or incorrect. They were asked to indicate separately if no answer could be given due to ASR errors or when the answer (regardless of correctness) seemed to be improperly or oddly phrased. The data are presented in Table 1.

Table 1: Evaluation data.

Total number of questions	388
Bad transcripts due to ASR errors	59
Well-formed transcripts (no ASR errors, or fixed)	329
Correct answers	219 (66.6% of 329)
Partially correct answers	45 (13.7%)
Incorrect answers	65 (18.8%)
The answer was given but sounded unnatural/ungrammatical	25

We found that the system returns correct answer in 67% of the cases. Including partially correct ones, the accuracy rises to 80%. Given that inter-annotator agreement of around 0.72 was observed in (Platonov and Schubert, 2018) for human judgements of prepositional relations on a 5-point Likert scale, our results are reasonable. Such variability is due to the fact that spatial relations are quite vague and people’s intuitions differ significantly. Correctness was tracked for both the ULFs produced and the generated spoken answers. The spatial com-

ponent displays satisfactory sensitivity in terms of the certainty cut-off threshold, i.e., the threshold determining which objects are included seems in accord with human intuitions. Below we present separate evaluation data for the ULF parser.

Table 2: Evaluation data on ULF parsing.

Total number of spatial questions	635
Number of correctly interpreted questions	470
Number of incorrectly interpreted questions	165
Number of incorrect parses due to ASR errors	87
Accuracy	74.02%
Percentage of incorrect parses due to ASR errors	52.73%

Most errors in the ULF parsing are due to either ASR errors, unsupported sentence constructions (e.g., passive voice expressions, some prepositions, etc.), or indexical questions (e.g., “What block did I just move?”).

5 Conclusion and Future Work

We have built a spatial QA system for a physical blocks world, already able to handle a majority of questions in dialogue mode. We are not aware of any other end-to-end system with comparable abilities in QA about spatial relations. Our spatial language model relies on intuitive computational models of spatial prepositions that come close to mirroring human judgments by combining geometrical information with context-specific information about the objects and the scene. This enables natural user-machine interaction. The ongoing work is targeting world history-tracking to enable answering question like “Where was the Toyota block initially?”

Acknowledgments

This work was supported by DARPA grant W911NF-15-1-0542. We thank our team of volunteers for their suggestions and contributions to system evaluation.

References

- Eric Bigelow, Daniel Scarafoni, Lenhart Schubert, and Alex Wilson. 2015. On the need for imagistic modeling in story understanding. *Biologically Inspired Cognitive Architectures*, 11:22–28.
- Yonatan Bisk, Kevin J Shih, Yejin Choi, and Daniel Marcu. 2018. Learning interpretable spatial operations in a rich 3d blocks world. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Scott Elliott Fahlman. 1974. A planning system for robot construction tasks. *Artificial intelligence*, 5(1):1–49.
- Simon Garrod, Gillian Ferrier, and Siobhan Campbell. 1999. In and on: investigating the functional geometry of spatial prepositions. *Cognition*, 72(2):167–189.
- Annette Herskovits. 1985. Semantics and pragmatics of locative expressions. *Cognitive Science*, 9(3):341–378.
- Gene Louis Kim and Lenhart Schubert. 2019. A type-coherent, expressive representation as an initial step to language understanding. In *Proceedings of the 13th International Conference on Computational Semantics-Long Papers*, pages 13–30.
- Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. 2019. Clevr-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. *arXiv preprint arXiv:1903.03166*.
- Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*.
- Ian Perera, James Allen, Choh Man Teng, and Lucian Galescu. 2018. Building and learning structures in a situated blocks world through deep language understanding. In *Proceedings of the First International Workshop on Spatial Language Understanding*, pages 12–20.
- Georgiy Platonov and Lenhart Schubert. 2018. Computational models for spatial prepositions. In *Proceedings of the First International Workshop on Spatial Language Understanding*, pages 21–30.
- S.Z. Razavi, M.R. Ali, T.H. Smith, L.K. Schubert, and M.E. Hoque. 2016. The LISSA virtual human and ASD teens: An overview of initial experiments. In *Proc. of the 16th Int. Conf. on Intelligent Virtual Agents (IVA 2016)*, pages 460–463, Los Angeles, CA.
- S.Z. Razavi, L.K. Schubert, M.R. Ali, and H.E. Hoque. 2017. Managing casual spoken dialogue using flexible schemas, pattern transduction trees, and gist clauses. In *5th Ann. Conf. on Advances in Cognitive Systems (ACS 2017)*, Rensselaer Polytechnic Institute, Troy, NY.
- Andrea Tyler and Vyvyan Evans. 2003. *The semantics of English prepositions: Spatial scenes, embodied meaning, and cognition*. Cambridge University Press.
- Terry Winograd. 1972. Understanding natural language. *Cognitive psychology*, 3(1):1–191.

rrSDS: Towards a Robot-ready Spoken Dialogue System

Casey Kennington

Daniele Moro

Lucas Marchand

Jake Carns

David McNeill

Department of Computer Science

Boise State University

1910 W University Dr

Boise, ID 83725

`firstname.lastname@boisestate.edu`

Abstract

Spoken interaction with a physical robot requires a dialogue system that is modular, multimodal, distributive, incremental and temporally aligned. In this demo paper, we make significant contributions towards fulfilling these requirements by expanding upon the ReTiCo incremental framework. We outline the incremental and multimodal modules and how their computation can be distributed. We demonstrate the power and flexibility of our robot-ready spoken dialogue system to be integrated with almost any robot.

1 Introduction

Spoken Dialogue Systems (SDSS) are well-suited to handle complex artifacts of dialogue such as hesitations and clarification requests in many domains. However, to further extend SDSS to work effectively on physical robots, we offer the following additional requirements towards a *robot-ready* SDS: **modular**: robot components are modular and individual modules must be able to integrate with SDS modules, **multimodal**: robots are *situated* dialogue partners whose many sensors must be integrated with the SDS speech input, **distributive**: robot and SDS modules should easily communicate with each other in a distributed environment, **incremental**: modules must be able to process input quickly and immediately, **aligned**: sensors must be temporally aligned, i.e., synchronized in time.

Existing systems offer solutions to some of the requirements. The OpenDial toolkit gives researchers the ability to model dialogue states using probabilistic rules (Lison and Kennington, 2016), but any incrementality has not been systematically evaluated. InproTK (Baumann and Schlangen, 2012), is incremental and InproTK_s (Kennington et al., 2014) added distributiveness and multimodality, and Kennington et al. (2017) offered an ap-

proach to temporal alignment, albeit with offline evaluation.

The PSI framework is inherently modular, multimodal, temporally aligned, has been evaluated on robot platforms, and has several options for distributing computation (Bohus et al., 2017). However, the PSI framework does not yet build on any incremental framework. Also similar to our work is the platform MultiBot presented in Marge et al. (2019), but that model does not work incrementally nor does it consider temporal alignment.

In this paper, we design and evaluate a modular, incremental, multimodal, and distributive *robot-ready* SDS, called *rrSDS* which is primarily written in the Python programming language.¹ To address the requirements of *modularity* and *incrementality*, we adopt the Incremental Unit Framework (Schlangen and Skantze, 2011) where *incremental units* (IUs) are passed between modules (IUs can be *added* to reflect new information, or *revoked* if a previously added IU needs to be updated) by building on the *ReTiCo* (Michael and Möller, 2019) platform. To handle *distributiveness*, *rrSDS* has modules (i.e., *ZeroMQ*, *ROS*) that afford interoperability with processes outside of the system. To address the requirement of *multimodality*, we build on top of the modularity requirement and incorporate additional sensors (e.g., cameras and internal robot states).

2 The *rrSDS* Spoken Dialogue System

ReTiCo has existing modules for built-in microphones and Google Speech API for speech recognition. We extend it to be multimodal by adding additional sensor modules such as cameras and internal robot states, depicted in Figure 1. We add distributive modules that handle interoperability with

¹Available at <https://github.com/bsu-slim/rr-sds>

outside modules. The rest of this section explains the modules for *rr* SDS.

Dialogue Management OpenDial is a toolkit for developing SDSS with probabilistic rules (Lison and Kennington, 2016) that can be used as a rule-based dialogue manager (DM), but can be extended to include stochastic processes when data is available. We incorporate a recent Python implementation called pyOpenDial (Jang et al., 2019) into our SDS as a DM. Our pyOpenDial module takes any IU payload, expecting a list of attributes (i.e., variables) and values that it adds to pyOpenDial’s dialogue state as attribute/value pairs.

Natural Language Understanding *Words-as-Classifiers* (WAC) is a model of grounded semantics that learns a ‘fitness’ score between physical entities and words (Kennington and Schlangen, 2015), where each word in a known vocabulary is represented as a classifier. WAC is inherently incremental and can learn word groundings with minimal training data. This module takes words from ASR and features from detected objects. It outputs the best fit word for the detected object as well as confidence scores for all the words in its vocabulary and their fitness to all detected objects.

Object Detection & Feature Extraction This module uses Huang et al. (2017), which builds on several other advances in fast object detection. The output of this module is a list of bounding boxes for each object, along with corresponding labels and confidence scores. The feature extractor takes those object bounding boxes, isolates the bounded object from the rest of the image, and passes that single object image through a pre-trained Imagenet model, for example, EfficientNet (Tan and Le, 2019) or InceptionV3,² but designers can specify any Keras network and target layer. This module outputs a list of vectors that represent each object that was found in the input image.

Seeed Respeaker The respeaker is an array microphone with 6 individual microphones on a disc-like board.³ Respeaker has built-in functionality for direction-of-arrival, noise suppression, keyword wake up, and network connectivity.

²This needs to match the vector representations that any grounded NLU module (e.g., WAC) was trained with.

³http://wiki.seeedstudio.com/ReSpeaker_Core_v2.0/

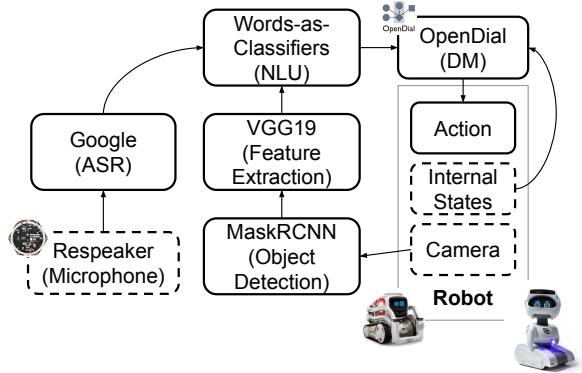


Figure 1: Overview of our *rr* SDS integrated with the robot modules. Dashed lines represent sensory input modules.

Distributive Interop ZeroMQ is a universal message passing library that builds and maintains sockets that carry atomic messages across various transports.⁴ ZeroMQ supports most programming languages and operating systems. The amount of code required to use ZeroMQ to pass messages between separate processes is very minimal. For our SDS, we have two types of ZeroMQ modules: Readers and Writers.

A key interoperability module in *rr* SDS is the Robotics Operating System (ROS), which is widely adopted in the robotics community.⁵ ROS has a built-in communication layer across any robotic system’s architecture that provides data pipelines on different scopes (Quigley et al., 2009). Our *rr* SDS interfaces with ROS using Publish and Subscribe modules (similar to ZeroMQ’s Writer and Reader modules). We evaluated our implementation using Turtlesim, a common test bed simulation for ROS.⁶

Additional Modules *rr* SDS has additional modules that we do not use in our evaluation, but we do mention them for completeness: Azure Cognitive Services Speech Recognition (ASR), Azure Emotion Recognition API (takes in an image and returns a distribution over 8 emotional states), Azure Object Detector (similar to the MaskRCNN module above, this takes an image as input and returns a list of bounding boxes and corresponding labels), RASA (NLU) (Bocklisch et al., 2017) which has been evaluated to be competitive with commercial NLU platforms (Braun et al., 2017; Liu et al., 2019)

⁴<https://zeromq.org/>

⁵We note that our chosen interoperability platforms are also available on PSI, which motivated our choices.

⁶<http://wiki.ros.org/turtlesim>



Figure 2: Misty in its task setting: Misty could move its head left and right, and had to look down at the objects on the table.

and has recently been incrementalized [Rafla and Kennington \(2019\)](#).

3 Evaluation

We used the Mistyrobotics Misty II and Anki Cozmo robot platforms to evaluate rr SDS, depicted in Figures 2 and 3. We briefly explain the two platforms and the modules we built to integrate them into our rr SDS, then we describe the evaluation.

Robot Modules Integration of Cozmo with rr SDS is done using its Python SDK and Misty using its REST API, each broken into three ReTiCo modules: (1) camera, (2) internal state, and (3) action control. The output of each camera module is an IU with a still image as its payload. Both robots have internal state variables (e.g., left-wheel-speed, head-height, light-height). As the state of the robot changes, this module produces an IU containing a full attribute-value matrix of the internal state representation (e.g., wheel speed, lift height) at the state update. The action modules use the decisions made by the DM to produce the following actions: `explore`, `align`, `approach`, `confirm`, and `speak`.

A human user utters a short description and the robot attempts to explore its surroundings until it finds an object that matches the description. After a user description, the robot enters an `explore` state to seek out an object, then an `align` state to move the object into center view. Then the robot confirms if the description matches the object it is looking at. The robot speaks, uttering either *That looks X* or *Uhh that's not X that's Y* where *X* is the description and *Y* is the robot's best guess at a description (i.e., a better color word).

An overview of our rr SDS is depicted in Figure 1. We use the Respeaker microphone, Google ASR,

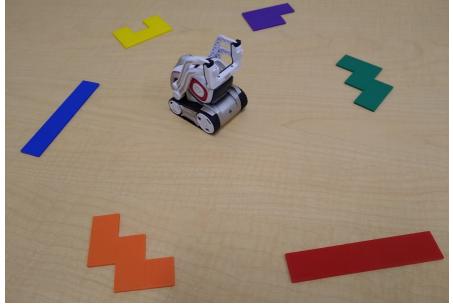


Figure 3: Cozmo in its task setting: in order for Cozmo to observe objects with its camera, its head had to be pointed slightly down, and its lift had to be raised.

and WAC modules for spoken input, recognition, and understanding, respectively. Each robot's camera passed image frames to the MaskRCNN Object Detection module, then we used the VGG19 fc1 layer (4096 features; pre-trained on imangenet data) to represent objects for the WAC module. For dialogue and action management, we used the pyOpenDial module. For the WAC module, we used logistic regression classifiers pretrained on words that only focused on colors. We obtained the training data for WAC by capturing objects using Cozmo's camera; 5-10 training instances per color (trained using l2 normalization).

We recruited 15 participants from Boise State University (4 female, 11 male) to interact with each robot and fill out the Godspeed Questionnaire ([Bartneck et al., 2009](#)) after each interaction.

Our rr SDS can run completely on a single machine;⁷ output from all system processing modules were logged using PSI on a separate machine. We used the ZeroMQ modules to send information from rr SDS to PSI.

We found in our evaluation that participants were able to accomplish the same number of tasks with both robots, but generally found Cozmo interesting, likeable and pleasant whereas Misty was judged as more mechanistic, rigid, stagnant and machine like.

4 Conclusions & Future Work

Our rr SDS is flexible, being evaluated on multiple robot platforms to create engaging human-robot interactions, and fulfills the modular, incremental, multimoal, and distributive requirements for a robot-ready SDS. Our evaluation of rr SDS allowed users to successfully interact with two different

⁷Our Machine had 32GB of RAM and an NVidia Tesla M40 with 12GB of Video RAM.

robots to accomplish a simple task with comparable performance. *rrSDS* is agnostic to the robot platform used, enabling future research to experiment with robot platforms using our flexible system. For future work, we plan to add natural language generation modules and integrate *rrSDS* more directly with PSI to make use of its architecture, thereby allowing developers and researchers to make use of PSI temporal alignment functionality, but spend most of their development time with Python.

Acknowledgements We thank the anonymous reviewers for their feedback and useful insights. We thank Microsoft Research for answering questions related to PSI. This work was approved under the Boise State University IRB #126-SB20-012.

References

- Christoph Bartneck, Dana Kulic, Elizabeth Croft, and Susana Zoghbi. 2009. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International journal of social robotics*, 1(1):71–81.
- Timo Baumann and David Schlangen. 2012. The InproTK 2012 release. In *NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012)*, pages 29–32.
- Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. *Rasa: Open Source Language Understanding and Dialogue Management*. *Proceedings of the 31st Conference on Neural Information Processing Systems*.
- Dan Bohus, Sean Andrist, and Mihai Jalobeanu. 2017. *Rapid Development of Multimodal Interactive Systems: A Demonstration of Platform for Situated Intelligence*. In *Proceedings of ICMI*, Glasgow, UK. ACM.
- Daniel Braun, Adrian Hernandez Mendez, Florian Matthes, and Manfred Langen. 2017. *Evaluating Natural Language Understanding Services for Conversational Question Answering Systems*. In *Proceedings of the SIGDIAL 2017 Conference*, pages 174–185.
- Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. 2017. *Speed/accuracy trade-offs for modern convolutional object detectors*. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 3296–3305.
- Youngsoo Jang, Jongmin Lee, Jaeyoung Park, Kyeng-Hun Lee, Pierre Lison, and Kee-Eung Kim. 2019. PyOpenDial: A Python-based Domain-Independent Toolkit for Developing Spoken Dialogue Systems with Probabilistic Rules. In *Proceedings of EMNLP*.
- Casey Kennington, Ting Han, and David Schlangen. 2017. *Temporal Alignment Using the Incremental Unit Framework*. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction, ICMI 2017*, pages 297–301, New York, NY, USA. ACM.
- Casey Kennington, Spyros Kousidis, and David Schlangen. 2014. *InproTKs: A Toolkit for Incremental Situated Processing*. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 84–88, Philadelphia, PA, U.S.A. Association for Computational Linguistics.
- Casey Kennington and David Schlangen. 2015. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. In *Proceedings of ACL-IJCNLP 2015*, volume 1.
- Pierre Lison and Casey Kennington. 2016. OpenDial: A toolkit for developing spoken dialogue systems with probabilistic rules. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - System Demonstrations*.
- Xingkun Liu, Arash Eshghi, Paweł Swietojanski, and Verena Rieser. 2019. *Benchmarking Natural Language Understanding Services for building Conversational Agents*.
- Matthew Marge, Stephen Nogar, Cory J Hayes, Stephanie M Lukin, Jesse Bloecker, Eric Holder, and Clare Voss. 2019. *A Research Platform for Multi-Robot Dialogue with Humans*. Technical report.
- Thilo Michael and Sebastian Möller. 2019. ReTiCo: An open-source framework for modeling real-time conversations in spoken dialogue systems. In *Tagungsband der 30. Konferenz Elektronische Sprachsignalverarbeitung 2019*, ESSV, pages 134–140, Dresden. TUDpress, Dresden.
- Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*.
- Andrew Rafla and Casey Kennington. 2019. *Incrementalizing RASA’s Open-Source Natural Language Understanding Pipeline*. *arXiv*.
- David Schlangen and Gabriel Skantze. 2011. *A General, Abstract Model of Incremental Dialogue Processing*. In *Dialogue & Discourse*, volume 2, pages 83–111.
- Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv*.

Discovering Knowledge Graph Schema from Short Natural Language Text via Dialog

Subhasish Ghosh

TCS Research, India

g.subhasish@tcs.com

Aniket Pramanick

TCS Research, India

aniket.pramanick@tcs.com

Arpita Kundu

TCS Research, India

arpita.kundu1@tcs.com

Indrajit Bhattacharya

TCS Research, India

b.indrajit@tcs.com

Abstract

We study the problem of schema discovery for knowledge graphs. We propose a solution where an agent engages in multi-turn dialog with an expert for this purpose. Each mini-dialog focuses on a short natural language statement, and looks to elicit the expert’s desired schema-based interpretation of that statement, taking into account possible augmentations to the schema. The overall schema evolves by performing dialog over a collection of such statements. We take into account the probability that the expert does not respond to a query, and model this probability as a function of the complexity of the query. For such mini-dialogs with response uncertainty, we propose a dialog strategy that looks to elicit the schema over as short a dialog as possible. By combining the notion of uncertainty sampling from active learning with generalized binary search, the strategy asks the query with the highest expected reduction of entropy. We show that this significantly reduces dialog complexity while engaging the expert in meaningful dialog.

1 Introduction

Increasingly, within and outside of enterprises, knowledge is represented and stored in knowledge graphs (Pujara and Singh, 2018). Many such knowledge graphs have a schema (Mitchell et al., 2015), so that entities and relationships are labeled with discrete categories. There has been a lot of research on annotating natural language text using the *provided* schema of such knowledge graphs (Chakrabarti and Talukdar, 2017).

A task that is often overlooked in this context is that of designing and updating the knowledge graph schema itself. It is common for such knowledge graphs schemas to be incomplete. The schema is typically designed by a human expert ahead of time based assumptions about the data to be accommodated in the future. Thus schema design is not

a one-shot affair, since the expert becomes better aware of the need once the system is deployed. Updating the schema based on new needs forms an integral part of maintaining a knowledge graph.

In this paper, we propose dialog with an expert as a means to construct and update such a schema using as evidence a collection of short natural language statements from the domain. The role of dialog has been explored in the context of updating schema-free knowledge (Hixon et al., 2015; Pappu and Rudnicky, 2014a,b). Our purpose is to recover the expert’s desired schema for such knowledge. The proposed overall dialog consists of a sequence of many mini dialogs. Each mini dialog focuses on a single short natural language statement, and recovers the expert’s desired schema-based annotation of the statement, taking into account possible enhancements to the current schema. The overall schema is augmented iteratively by performing dialog over the collection of statements. The dialog ends when the agent is able to confidently annotate all statements with its current schema.

As an illustration, consider the following mini dialog between a human (H) and an agent (A).

A: *I need help with the statement Tom Mitchell works on never-ending learning. Is Tom Mitchell a Researcher?*

H: Yes.

A: *Is never ending learning any-of Paper or Topic?*

H: It’s a topic.

A: *Is this about Researcher writing Paper about Topic?*

H: No.

A: *Is this about a new entity type?*

H: Yes.

A: *What is the name of the new entity?*

H: This is about a project

A: *Is this about Researcher related to Project related to Topic?*

H: Yes, that is right.

A: *How is Project related to Researcher?*

H: Researchers lead projects.

A: *How is Project related to Topic?*

H: Projects focus on topics.

A: I understand the statement as researcher Tom Mitchell leading project focusing on topic never ending learning.

The agent had existing knowledge of certain entity and relationship types, but needed to introduce the new entity type *Project* and the relation types *leads(Researcher, Project)* and *focusesOn(Project Topic)* to the schema. It was then able to associate a short path from the schema (Research to Project to Topic) with the statement. It elicited this knowledge from the expert using a short dialog.

Queries are sequentially chosen based on their ‘utility’ - the reduction in uncertainty over path probabilities. This has high-level similarities with active learning for structured prediction (Culotta and McCallum, 2005; Settles and Craven, 2008). This considers eliciting the entire structured label for an instance from the expert *in one shot*. In reality, experts do not always answer queries. If the query is too ‘complex’ (e.g. “Which of the following 100 paths is the right schema path for this statement?”), the expert is very likely to not answer at all. On the other hand, a simpler query (e.g. “Is this statement about a project?”) is more likely to get a response. Therefore, we consider a *mini dialog* for each instance consisting of a sequence of simple queries, based on their utility.

Each query has an associated probability of response from the user, thereby affecting its utility. The response probability depends on the complexity of the query. Active learning literature has studied no-response probability for complex queries(Yan et al., 2016). However, the notion of complexity in this work relates to closeness to the decision boundary. In the structured prediction setting, we hypothesize that the complexity relates more to the ‘size’ of the query.

In the presence of such response uncertainty, given a collection of statements and an initial knowledge graph, our goal is to design a dialog strategy that picks the statements in some sequence and elicits their desired schema paths from the expert with the shortest overall dialog length.

We propose a dialog strategy that combines the notion of uncertainty sampling from the active learning literature with that of generalized binary search. We iteratively pick the best statement to query, and then the query for the statement by considering expected entropy reduction, accounting for no-response probability. We propose two different types of categorical queries, *any-of* queries, where the expert confirms one part of a bi-partition

of the current candidate set, and *which-of* queries, where the expert is asked to pick a specific candidate from a set. We model response probability as a parametric function of query complexity, which we represent as the sum of the lengths of the paths included in the query. We show that the standard active learning strategy falls out as a special case of our strategy when there is no response uncertainty. The no-response parameters are learnt by the agent as the dialog progresses.

We evaluate our strategy using a collection of short statements from the web-page of a large organization. We show that our proposed strategy results in meaningful and dialog with an expert that yields the expert’s desired schema for the organization via significantly lower dialog complexity compared to multiple baseline strategies.

2 Related Work

Hixon et al. (2015) investigate the problem of augmenting knowledge graphs using an open dialog with the user to improve factual multiple-choice science question answering. Earlier work(Pappu and Rudnicky, 2014a,b) looks at designing dialog for the related task of information retrieval for academic events. In all of these cases, the back-end knowledge graph is type-free and does not involve a schema. Mazumder et al. (2019) address lifelong learning via dialog, where they query the expert back for supporting facts when confronted with queries for entities and relationships with insufficient evidence in the knowledge graph. In contrast, we focus on dialog for augmenting the schema for typed knowledge graphs.

There is existing work on active learning for structured output spaces(Culotta and McCallum, 2005; Settles and Craven, 2008; Tong and Koller, 2001). Of these, sequence annotation (Culotta and McCallum, 2005; Settles and Craven, 2008) also considers paths, but not on schema graphs. More importantly, these pose a single query to the expert for the structured label of an instance. In contrast, we propose to break this into a series of simple queries, in view of answer uncertainty, which is not considered in this line of work.

Active learning with imperfect labelers (Yan et al., 2016) considers no-response and query complexity. However, since this is not for structured labels, query complexity does not account for structural complexity.

3 KG Schema Learning Problem

Schema and statement schema paths: A Knowledge Graph schema is a graph $G = \{E, R\}$ where E is the set of entity types and R is the set of binary relation types. For example, a Research domain may have entity types *Researcher*, *Paper*, *Project* and *Topic*, and relation types *authorOf(Researcher, Paper)*, *affiliatedWith(Researcher, Project)* and *focusesOn(Project, Topic)*.

A typed knowledge graph (or KG in short) $K = (G, I)$ has a schema G and instances I of the entity and relation types in G . For example, our Research KG may contain entity instances *isResearcher(T. Mitchell)*, *isProject(NELL)*, *isPaper(Coupled semi-supervised learning)*, and relation instance *affiliatedWith(T. Mitchell, NELL)*, etc. We assume that every entity type has a single *Name* attribute.

Consider an example statement s_i : “A scientist wrote the paper ‘coupled semi-supervised learning’”. We restrict ourselves to short statements that refer to at most two entities and two entity types, which we call e_{i1}, e_{i2} and t_{i1}, t_{i2} respectively. In s_i , $t_{i1} = \text{Researcher}$, and ‘scientist’ is a *mention* of t_{i1} , which we denote as $m(t_{i1})$. $t_{i2} = \text{Paper}$, and its mention $m(t_{i2})$ is ‘paper’. e_{i2} is a specific paper instance and its mention $m(e_{i2})$ is ‘Coupled semi-supervised learning’. The first entity e_{i1} is not explicitly mentioned in this statement. Beyond mentioning the entities and their types, s_i also refers to a connection between them in the schema via the defined binary relations. We call this the *statement schema path* p_i , which is a subgraph of the schema. Here, p_i is Researcher-authorOf-Paper. Typically, given a statement, the above variables other than mentions are latent or unobserved. We will assume the availability of a probability model for the posterior distribution of these variables $P(e_{i1}, e_{i2}, t_{i1}, t_{i2}, p_i | m(e_{i1}), m(e_{i2}), m(t_{i1}), m(t_{i2}))$ given the mentions in the statement. In Sec.4, we define such a model and its corresponding inference and learning algorithms.

An important consideration for such a model in the context of schema discovery is its ability to consider entity types (also entities) not contained in the current KG schema (also instances). Consider the statement s_1 from the introduction: “Tom Mitchell works on never ending learning”. The true schema path p_1 in this case looks as follows: Researcher-leads-Project-focusesOn-Topic. This includes the entity type *Project* and relation types *focusesOn(Project, Key-*

word) and *leads(Researcher, Project)* which are not contained in the observed schema G ; i.e., $p_1 \not\subset G$. In order to correctly interpret s_1 , its schema path p_1 , and therefore the schema, needs to be enhanced to include an additional entity type and two new relation types. In other words, the posterior distribution should be capable of assigning non-zero probability to previously unseen yet likely entity types and schema paths.

Given a set of statements S and an initial schema G^0 , our goal is an enhancement G^* of G^0 such that the schema paths p_i for all statements $s_i \in S$ are contained in G^* . Along with G^* , the schema paths for the individual statements are unknown as well, and need to be identified.

Dialog for schema path discovery: The problem above is difficult to solve in practice, even without considering schema enhancements, and requires very large volumes of training data. To aid this generally intractable search, we propose a dialog with an expert. Our task is to design a dialog strategy that reduces the uncertainty (entropy) about the schema path p_i for each $s_i \in S$ — and as a result about the complete schema graph G^* — as much as possible given a dialog length as a budget. The length of the dialog is the aggregated complexity of all the queries posed to the expert during the dialog.

Our strategy uses one mini dialog for each question statement in S . The j^{th} mini dialog considers some statement $s_i \in S$, and poses a sequence of queries to identify the true schema path p_i for s_i . Each mini dialog consists of a sequence of simple mini-queries, denoted as $q(s_i)$. Specifically, the agent poses two kinds of close-ended mini-queries to the expert. The first is a binary (yes/no) query of the form “Is the statement about *any of* p_1 OR p_2 OR ... p_k ?” , where each p_1 is a possible path in G^* . The second is an n-ary query of the form “This statement is about *which of* p_1 OR p_2 OR ... p_k ?” Note that the response for this query can be ‘none’. The paths in the queries can include new entity and relation types that are not in the current schema graph. Note that such categorical queries can only recover the *structure* of p_i . An additional type of query therefore asks for the names of any new entity or relation types in p_i .

Any such query $q(s_i)$ has a *utility*. Intuitively, utility measures potential reduction in entropy of the posterior distribution over possible responses to the query. There reduction is 0 if the query gets no

response from the expert. The response probability depends on the complexity of the query. Our goal is design a strategy that can evaluate the utilities for the different query types taking into account no-response probability, and then select the mini-query with the best utility at each step.

4 Candidate Schema Paths

In this section, we discuss a probabilistic model for schema paths for a statement, and then an algorithm for finding the best schema path given a statement along with their probabilities. This is not the focus of our work, and ideally we would prefer to use a state of the art method for this task. There has been recent progress on joint linking of mentions in short statements to entities and relationships in a knowledge graph (Sakor et al., 2019). This is similar to our task, but does not consider the possibility of extending the current schema with new entity and relationship types. There is also work on inferring paths in schema-based knowledge graph given a query node (Lao et al., 2011) using random walks. We extend this line of work for statements with mentions while also considering new schema nodes in the random walk. In this section, we first explain our probabilistic model, and then the candidate path sampling algorithm.

4.1 Model for Schema Paths

For a statement s_i such as ‘‘Researcher Tom Mitchell works on never ending learning’’, and a current KG $K(G, I)$, our first goal is to define a posterior distribution for the types and path for s_i . This statement has entity mentions $m(e_{i1})$ =‘Tom Mitchell’ and $m(e_{i2})$ =‘never ending learning’, and one type mention $m(t_{i1})$ =‘Researcher’. The second type mention is absent.

Mention identification is not the focus of our work, and we use simple unsupervised NLP techniques which were sufficient for our purposes. These may be substituted with more sophisticated supervised techniques when needed without affecting the remaining components of our solution. We assume that a verb phrase separates the first ($m(e_{i1}), m(t_{i1})$) and second ($m(e_{i2}), m(t_{i2})$) set of entity and type mentions, such as ‘works on’ in this statement. We use a combination of noun phrase detection and named entity detection from *nltk*¹ to identify m_{e1} and m_{e2} .

¹<https://www.nltk.org/>

We factorize the posterior distribution as

$$\begin{aligned} & P(e_{i1}, e_{i2}, t_{i1}, t_{i2}, p_i | m_{i1}, m_{i2}) \\ & = P(e_{i1}, t_{i1} | m_{i1}) P(e_{i2}, t_{i2} | m_{i2}) P(p_i | t_{i1}, t_{i2}) \end{aligned} \quad (1)$$

We have used m_{i1} and m_{i2} as shorthand for $m(e_{i1}), m(t_{i1})$ and $m(e_{i2}), m(t_{i2})$ respectively. Here, the first term is the posterior distribution over the entity and type for the first mention pair, the second that for the second mention pair, and the third is the posterior for the schema path given the two entity types.

We assume the first two distributions to be identical. We associate two distributions with each entity type t_i in the current KG. The first θ_i^t is a distribution over entity instances e currently associated with t_i . For example, the *Researcher* type may have a non-zero probability over entity instances *Tom Mitchell*, *Will Cohen*, etc. The second ϕ_i^t is over possible mentions of this type. For example, the type *Researcher* may have non-zero probability over mentions *researcher*, *scientist*, *professor*, etc. An individual entity instance e_i also has a distribution ϕ_i^e over possible mentions of it. For example, entity *Tom Mitchell* has non-zero probability over different ways of mentioning the name, e.g. *Tom Mitchell*, *T. Mitchell*, etc. Accordingly, the posterior distribution over type and entity is further factorized as

$$\begin{aligned} & P(e_{i1}, t_{i1} | m(e_{i1}), m(t_{i1})) \propto \\ & P(m(e_{i1}) | e_{i1}; \phi^e) P(m(t_{i1}) | t_{i1}; \phi^t) P(e_{i1} | t_{i1}; \theta^t) \end{aligned} \quad (2)$$

For previously encountered mentions, we use these two distributions to identify the most likely type. For new mentions, the type could be one of the existing types in E or a new type. For this, all three sets of distributions are smoothed to allow for previously unseen mentions and entities.

We now come to the posterior distribution $P(p_i | t_{i1}, t_{i2})$ of the connecting schema path given the two entity types. We model the statement path as a random path in a partially-observed schema graph, with start probabilities over entity types and transition probabilities over relations between entity types. The path needs to start at t_{i1} and end at t_{i2} . This is similar to random walks used in (Lao et al., 2011) for link prediction in knowledge graphs. The difference is that we admit the possibility of previously unseen entity and relation types.

The probability of a statement path is defined as:

$$P(p_i|t_{i1}, t_{i2}; \pi, Q) = P_s(t_{i1}) \prod_{(t_j, t_k) \in p_i} P_t(t_k|t_j) \quad (3)$$

where $P_s(\cdot)$ is the distribution over start entities of the random walk, and $P_t(\cdot|t_j)$ is the transition distribution from entity type t_j to other entity types.

The definitions for π and Q need to account for new entity and relationship types. Accordingly, we define the probability $\pi(k)$ of the random path starting at entity type k as follows:

$$\begin{aligned} P_s(k) &\propto n_k + \alpha_e \text{ for existing } k \\ &\propto \alpha_n \text{ for new } k \end{aligned}$$

Here, n_k is the number of previously seen edges from entity type k , and $\alpha_e > 0$ allows new start entities. Similarly, the probability $P_t(k|j)$ of entity type k following entity type j in the path is defined as follows:

$$\begin{aligned} P_t(k|j) &\propto n_{jk} + \beta_{ee} \text{ for existing } j \text{ and } k \\ &\propto \beta_{en} \text{ for existing } j, \text{ new } k \\ &\propto \beta_{ne} \text{ for new } j, \text{ existing } k \\ &\propto \beta_{nn} \text{ for new } j \text{ and } k \end{aligned}$$

Here, n_{jk} is the number previously seen transitions from entity type j to entity type k , and $\beta_{ee} > \beta_{ne}, \beta_{en} > \beta_{nn} > 0$ allows transitions from and to previously unrelated and unseen entity types. The intuition is that (a) more frequently seen transitions are more likely, and (b) while encountering new entities and relationships is possible, that probability progressively reduces with increasing training count.

4.2 Sampling Algorithm for Schema Paths

Since all the distributions involved are multinomials, their parameters can be estimated in a closed form, given an initial KG and training statements labeled with schema paths. Given estimates of these parameters, we now address the problem of identifying possible candidate paths for a statement along with their probabilities.

Having defined the distribution over schema paths p for a statement s , we need to identify the top-k most likely schema paths. For this, we use a MCMC technique based on Metropolis Hasting sampling (Neal, 1993; Andrieu et al., 2003) that performs a random walk over the space of schema

paths for a statement. This requires a proposal distribution $q(p'|p)$ over possible next paths p' given the current path p . We define the neighbors p' of any statement path p using two operations. (a) Vertex insertion: This introduces a vertex between currently adjacent entity types in p . For example, Researcher-rel-Topic has Researcher-rel-Paper-rel'-Topic as a vertex-insertion neighbor. Note that this inserted entity type can be an existing type or a new one. Vertex insertion can operate on any currently adjacent pair of types in the statement schema path. (b) Vertex collapse: This is the inverse of vertex insertion. This collapses an intermediate vertex of p by introducing a direct edge between its neighbors. Vertex collapse can operate any current intermediate vertex of the schema path. At each step, we sample a neighbor using a uniform proposal distribution $q(p'|p)$ over all the neighbors p' of the current schema path p defined by these two operations, and accept the sample with probability $A(p', p) = \min\{1, \frac{p(p')q(p|p')}{p(p)q(p'|p)}\}$, where the path probabilities follow Eqn.1.

5 Dialog for Schema Path Discovery

At this point, for each statement s_i , we have n candidate schema paths p_{i1}, p_{i2}, \dots , and their probabilities. Our task now is to minimize the entropy of schema path predictions over all statements with a dialog of length L , knowing that the expert may not answer all queries.

Let us recall the standard active learning paradigm: (a) select the next statement for expert labeling, (b) acquire expert's preferred label for selected statement, (c) retrain model with newly labeled statement in training data, and continue until budget is exhausted. For selecting the next statement, the principle of uncertainty sampling is followed, with entropy as the notion of uncertainty (Hwa, 2004). In step (b), the expert provides the preferred label in one shot, even for structured output spaces (Culotta and McCallum, 2005). We call this interaction format the *active learning dialog format* and the overall strategy the entropy-based active learning strategy (**E-AL**).

Our major departure from this strategy is in step (b). We may present the expert with candidate schema paths and ask the expert to pick one. We call this a *which-of* query. When the list is long, this is unlikely to receive a response.

In addition, we can elicit the schema path with a mini-dialog, which is a sequence of simple mini-

queries. The basic idea is to iteratively prune the set of remaining candidates by splitting into two parts at each step, showing any one part to the expert and querying if it contains his preference. We call this an *any-of* query. Within a mini dialog (step (b)), our strategy repeatedly chooses the best (which-of or any-of) mini-query until the complete schema path is obtained from the expert. We call this the *mixed any-which mini dialog format*.

We can see that this is more general than the active learning dialog format.

Lemma 1. *The mixed any-which mini dialog format reduces to the active learning dialog format when each mini dialog has a single which-of query.*

For completion, we point out that if the elicited schema path in a mini dialog includes new entities and relationships, their names are also elicited from the expert via name queries. An example name query may be “What is the name of the new entity related to both researchers and topics?”.

Generalized binary search and Entropy Reduction: At any step within a mini dialog, the strategy needs to choose between the which-of query and many possible any-of queries, one for each bi-partition of the remaining candidate set. To evaluate different queries, we define the utility $u(q)$ of a query q borrowing from entropy-based uncertainty sampling and generalized binary search (Pelc, 2002).

Let P_i^k be the set of remaining candidates at step k of the mini dialog for statement s_i , and e_i^k denote the entropy of the candidate distribution. A bi-partition π_i^k splits P_i^k into P_{i1}^k and P_{i2}^k such that the entropies of the two splits are e_{i1}^k and e_{i2}^k . Depending on the expert’s response to the any-of query with this bi-partition, either P_{i1}^k or P_{i2}^k becomes the next candidate set. So the residual entropy after this query is either $(e_i^k - e_{i1}^k)$ or $(e_i^k - e_{i2}^k)$. We define the utility $u(q)$ of this bi-partition (any-of) query q as the *average reduction of entropy* after the query, which is $p_{i1}^k(e_i^k - e_{i1}^k) + p_{i2}^k(e_i^k - e_{i2}^k)$, where p_{i1}^k (p_{i2}^k) is the sum of candidate probabilities in the first (second) split.

Instead of all partitions, we order the candidate schema paths by probability, and consider each position in the order as a possible splitting point.

The alternative to partitioning is to present the entire set of candidates P_i^k to the expert and pose a which-of query. If the expert responds to this query, this mini dialog concludes and the entropy becomes 0, so that the utility (reduction in entropy) of the

which-of query is e_i^k . But this query is ‘complex’ and may not be answered by the expert.

Response uncertainty and Expected utility: Utility as defined above is not sufficient when queries are not answered with certainty. We define the complexity $c(q)$ of a query q as the sum of lengths of the paths specified in the query. For example, the query “Is this about Researchers-relatedTo-Projects-relatedTo-Topics?” has complexity 3, while the query “Is this about Researchers-relatedTo-Projects-relatedTo-Topics OR Researchers-relatedTo-Topics?” has complexity 5. We model the no-response probability $r(q)$ for a query as a function of its complexity. We have the option of various squashing functions which map positive integers to $[0, 1]$. We use the generalized logistic function:

$$r(q) = \frac{1}{1 + \exp(-w \times (c(q) - t))} \quad (4)$$

Using this definition of no-response probability, we define the *expected utility* of a query as $\bar{u}(q) = u(q)(1 - r(Q))$. Our strategy picks that mini-query at step k of a mini dialog that maximizes this expected utility. If the expert does not respond, then the next best mini-query is posed.

Having introduced the notion of expected utility to deal with no-response probability, we also modify step (a) of the framework by using expected utility instead of entropy to select the next statement for mini dialog. The expected utility of a statement is taken to be the maximum of the expected utilities of the which-of query and the possible any-of queries for its candidate set. This completes the description of our overall expected utility based dialog strategy (**EU**).

Theorem 2. *When the expert’s response probability is 1, and the dialog strategy is aware of this, the **EU** dialog strategy recovers the entropy-based active learning strategy (**E-AL**).*

The proof follows from the observation that when response probability is 1, the query with the highest expected utility is the which-of query on the entire set of candidates, and a mini dialog reduces to a single which-of query. So the dialog format becomes identical to that of active learning. Further, expected utility and entropy lead to the same statement being selected for querying.

Thus **E-AL** is a special case of the **EU** dialog strategy which is meaningful when the expert al-

ways responds. Our experiments show that **EU** far outperforms **E-AL** under response uncertainty.

Query for a partition: Having selected a bi-partition for an any-of query, the query for it needs to be formulated. Recall that *any-of* queries have the form “Is the statement about any of p_1 OR p_2 OR ...?”. The naive query for a partition enumerates all the paths in the smaller split. Instead, we identify the smallest *discriminating path feature* for the two parts of the partition — such as nodes, edges, length-2 paths, etc. For example, if all the paths in one of the splits contain the entity *Paper*, and no path in the other split contains it, then a possible query for the partition is the following: “Is the statement about papers?”. Being a less complex query, this is much more likely to get a response.

Estimating response probability: At the start of a dialog, the agent has an initial guess about the expert’s response probability model. After receiving responses or non-responses for specific queries, the parameters of this model can be estimated. On the conclusion of a mini dialog, we update the parameters w and t in the standard way for logistic regression using gradient descent.

6 Experiments

For empirical evaluation, we experimented with statements collected from the website of a large multi-national company, to see how accurately an expert’s desired schema behind the website data can be recovered. From 64 web-pages on the company’s website, we picked short statements representative about the company’s business, each containing at most 2 pairs of entity / type mentions. This resulted in a dataset of 850 short statements. In addition, we obtained from an expert a schema for the company’s business, which covers the statements that were picked. The resultant schema contains 15 entity types and 17 relationships between these. Using this schema, we manually annotated the statements with entities, entity types and type paths. About 40% statements had schema paths of length 1 and remaining of length 2. We used 200 the statements for train and the remaining 650 for test, ensuring that all entity and relation types used in the test schema paths are covered in the train.

To simulate the expert for large-scale experimentation, we used an ‘expert bot’ that had knowledge of the gold standard schema paths. The no-response parameters of the expert bot were manually specified. For each query, the expert bot sam-

pled from a Bernoulli distribution for the query to decide whether or not to respond.

To evaluate a dialog, we use the learning curve, where we plot the length of the dialog (terms of number of mini-queries) against the correctness of the inferred schema paths for the statements in the test set as compared with the gold-standard schema paths, evaluated using the F1 measure.

Our proposed dialog strategy **EU - L**, picks the next statement for mini-dialog and also the next mini-query within a mini-dialog using expected utility (EU), and further learns (L) the no-response parameter based on the expert’s action (response / no-response). We compare our strategy against a few baselines. **EU** does not learn the expert’s no-response probability. **Random + EU** picks the next statement for dialog uniformly at random (R), instead of expected utility, but uses expected utility (EU) to select the query within a mini dialog. **E-AL++** uses entropy (E) reduction assuming certain response for selecting the next statement for dialog as well as the next query within a mini dialog. Note that this is still more powerful than the *E-AL* strategy (Culotta and McCallum, 2005), which only poses *which-of* queries assuming the expert will respond. This makes no progress for non-zero no-response probability. In contrast, **E-AL++** uses the mixed any-which mini-dialog format and has the flexibility to pose *any-of* queries by partitioning. In addition, we also evaluate as a skyline **EU-O**, where the agent is an oracle (O) with perfect knowledge about the expert’s no-response parameters.

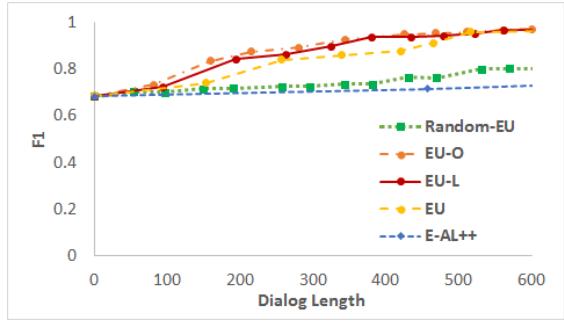


Figure 1: Learning curve for various dialog strategies

In the first experiment, the agent has knowledge of the complete schema, and does not need to consider new entity or relation types. In Fig.1, we plot the performances of **EU-L** along with the baselines. For each strategy, we plot the number of mini-queries on the x-axis and cumulative F1 on the test set after re-estimating parameters and re-

inferring paths at the end of a mini-dialog on the y-axis. **E-AL++** performs quite poorly and gets very little improvement in accuracy. Accounting for no-response via expected utility for selecting the query within a mini dialog (**Random + EU**) makes some improvement over this, but a **EU** makes a major improvement by using expected utility when selecting the next statement for dialog. Finally, **EU-L** makes a steady improvement by estimating the no-response parameters. Note that the skyline **EU-O** with perfect knowledge of the no-response parameters performs the best, but **EU-L** catches up with it as the dialogs progress, showing the effectiveness of our learning strategy. We also note that **E-AL++** can be seen as a special case of **EU** where the agent assumes that the expert always answers and does not update this model.

Average number of mini-queries per statement is very different for different strategies. This is 39 for **E-AL++**, 7.0 for **Random EU**, 7.0 for **EU** and finally to 6.0 for **EU-L**. Average query complexity (number of entity type nodes in a query) also varies significantly across the strategies. For **E-AL** (only *which-of* queries), average query complexity is 73.5, which explains why it makes no progress for non-zero no-response probability. This drops to 46 for **E-AL++**, 5.4 for **Random EU**, 4.8 for **EU** and finally to 2.7 for **EU-L**. Beyond dialog length, this also helps to highlight the benefit of expected utility and learning.

In the first experiment, the agent had knowledge of the complete schema, and only needed to detect occurrences of known entity and relation types in schema paths. We call this agent **EU-L Detect**. In the second experiment we compare this with an agent **EU-L Discover**, which only has partial knowledge of the schema at the start, and discovers new entity and relation types via dialog. In the training data for **EU-L Discover**, we randomly removed 5 entity types from the schema, which appear either as intermediate nodes as well as end nodes in the statement paths. We also removed the 9 relationships involving these entity types. This resulted in a pruned schema with 10 entities and 8 relationships. Statements associated with prune relationships were removed from the training data. In contrast, **EU-L Detect** was given the complete training data and the complete schema.

Fig.2 shows the performance of the two agents. Unsurprisingly, performance of *EU-L Discover* trails that of *EU-L Detect*, but it makes steady im-

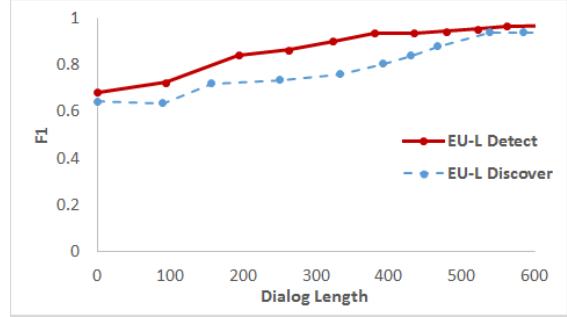


Figure 2: Learning curves for discovery and detection

provement as the dialog progresses.

In summary, we have shown that with response uncertainty, the **EU** strategy works significantly better than **E-AL** for detecting and discovering schema paths for short natural language statements, thereby enabling the enhancement of the underlying schema for the domain. On top of this, estimating the no-response parameters of the expert leads to further improvements in the learning curve.

7 Discussion and Future Work

We have introduced the problem of discovering the schema for the knowledge graph for a domain by engaging in dialog with an expert about interpreting short natural language statements in terms of the desired schema. We have proposed dialogue strategy that is aware of no-response probability for the expert, and accounts for it in its strategy by splitting the interaction for a statement into a sequence of short and simple queries, which are chosen using expected utility. The agent also estimates the no-response parameters for an expert. We have demonstrated that the proposed strategy is able to discover a schema starting from an initial partial observation. This goes well beyond the state of the art in active learning for structured spaces.

However, much remains to be done. One shortcoming of our expert model is that we have considered no-response to be the only form of ‘noise’. In reality, the expert, when presented with a complex question, may provide an incorrect response, and the algorithm should be resilient to a small probability of such erroneous responses. Next, we have only considered simple statements for which the statement graphs are paths. In general, such statement graphs can be trees or directed acyclic graphs. We will investigate these aspects in future work.

References

- Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael Jordan. 2003. An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43.
- Soumen Chakrabarti and Partha Talukdar. 2017. Tutorial on knowledge extraction and inference from text: Shallow, deep, and everything in between. In *International Conference on Information and Knowledge Management (CIKM)*.
- Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *Annual Conference of the American Association for Artificial Intelligence (AAAI)*.
- Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning knowledge graphs for question answering through conversational dialog. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.
- Rebecca Hwa. 2004. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):73–77.
- Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Sahisnu Mazumder, Bing Liu, Shuai Wang, and Nianzu Ma. 2019. Lifelong and interactive learning of factual knowledge in dialogues. In *Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.
- Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapa Nakashole, Emmanouil Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2015. Never-ending learning. In *Conference on Artificial Intelligence (AAAI)*.
- Radford Neal. 1993. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto.
- Aasish Pappu and Alexander Rudnicky. 2014a. Knowledge acquisition strategies for goal-oriented dialog systems. In *Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.
- Aasish Pappu and Alexander Rudnicky. 2014b. Learning situated knowledge bases through dialog. In *Annual Conference of the International Speech Communication Association (INTERSPEECH)*.
- Andrzej Pelc. 2002. Searching games with errors fifty years of coping with liars. *Theoretical Computer Science*, 270(1):71–109.
- Jay Pujara and Sameer Singh. 2018. Mining knowledge graphs from text. In *Tutorial at ACM International Conference on Web Search and Data Mining (WSDM)*.
- Ahmad Sakor, Isaiah Onando Mulang, Kuldeep Singh, Saeedeh Shekarpour, Maria Esther Vidal, Jens Lehmann, and Sren Auer. 2019. Old is gold: Linguistic driven approach for entity and relation linking of short text. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.
- Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Simon Tong and Daphne Koller. 2001. Active learning for structure in bayesian networks. In *International Joint Conference on Artificial Intelligence (IJCAI)*.
- Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. 2016. Active learning from imperfect labelers. In *International Conference on Neural Information Processing Systems (NIPS)*.

A Appendices

As appendix, we provide more details on the statement data preparation and include some example mini-dialogs generated on test statements by the **EU-L** strategy.

A.1 Statement Data Preparation

We prepare our statement data from the website of Tata Consultancy Services (TCS), which covers different Industries, Services, Products and Platforms and has web-pages under these covering Solutions, Insights, Customer Testimonials and Success Stories, White papers, etc. Each web-page is partially structured mentioning key people and their designations, names of clients, services and industries, along with descriptions of challenges, innovations and results. We model each web-page as a rooted directed acyclic graph, with each of the above as individual nodes. Then we consider pairs of nodes from within each web-page and create a statement with these by connecting them with templatized relation strings. The following are examples: *Client Woolworths Supermarkets appears in the context of Technology Operations, Malaysia Airlines is mentioned in the context of Industries Airline Industry*, etc. Note that the original web-pages mention entity types in some cases, such as in *Client Woolworths Supermarkets*, but not in others, such as in *Malaysia Airlines*. The statements are created to

match this data. For each statement, the unique connecting path between the two end nodes is initially set as the gold-standard path and then verified manually.

A.2 Example Mini Dialogs

We present three example mini-dialogs between the agent (A) and the Expert (E) bot. Each agent query is annotated with type of the query and its subject. The query type can be any-of, which-of or name query. Further, an any-of query can be enumerated or posed using a discriminating feature, such as node or short sub-path. The subject of the query can be the entity, its type or the path connecting two types. In addition to queries, the agent also provides context to the expert by stating facts that about which it is confident. These illustrate that the agent has basic NLU capabilities so that the expert does not have to stick to categorical answers.

The first example mini-dialog is a short one and includes examples of any-of enumeration queries on path and type and a which-of path query.

A: I need help with the statement *Client Woolworths Supermarkets appears in the context of Technology Operations*.

A: In *Client Woolworths Supermarkets*, I know *Client* means type *CLIENT*.

A: I know *Woolworths Supermarkets* means *CLIENT* Woolworths Supermarkets.

A: Is the type of *Technology Operations* any-of the following: *SERVICE*? [any-of-enumeration, type]

E: yes

A: Is the statement about any-of the following? *CLIENT* related-to *SERVICE*? [any-of-enumeration, path]

E: no

A: Is the statement about any-of the following? [any-of-enumeration, path]

1.*CLIENT* provides *CUSTOMER TESTIMONIAL* for *SERVICE*

2.*CLIENT* has *SUCCESS STORY* for *SERVICE*

E: yes

A: Which of the above? [which-of, path]

E: first one

A: Ok. I understand the statement as Client Woolworth provides Customer Testimonial for Service Technology Operations.

The second example mini-dialog additionally includes examples of which-of query on type, any-of discriminating node query on path, and discovery of a new entity.

A: I need help with the statement *Malaysia Airlines appears in the context of Industries Airline Industry*.

A: Is the entity for *Malaysia Airlines* any-of the following: new entity? [any-of-enumeration, entity]

E: yes

A: The type for *Malaysia Airlines* is which of the following? [which-of, type]

1.CUSTOMER TESTIMONIAL

2.RESULTS

3.PARTNERSHIP

4.LEADERSHIP

5.CLIENT

6.SERVICE

7.INDUSTRIES

E: It is a Client

A: In *Industries Airline Industry*, I know *Industries* means type *INDUSTRIES*.

A: Is the entity for *Airline Industry* any-of the following: new entity of type *INDUSTRIES*? [any-of-enumeration, entity]

E: yes

A: Is the statement about any-of the following: *SUCCESS STORY*? [any-of-discriminating-node, path]

E: yes

A: Ok. I understand the statement Client Malaysia Airlines has Success Story for Industries Airline Industry.

The third example mini-query additionally includes examples of a no-response from the expert, discovery of new types and relations and name queries for these.

A: I need help with the statement *Client Ceridian appears in the context of Industries Communications, Media & Technology*.

A: In *Client Ceridian*, I know *Client* means type *CLIENT*.

A: Is the entity for *Ceridian* any-of the following: new entity of type *CLIENT*? [any-of-enumeration, entity]

E: Yes

A: In *Industries Communications, Media & Technology*, I know *Industries* means type *INDUSTRIES*.

A: Is the entity for *Communications, Media & Technology* any-of the following: new entity of type *INDUSTRIES*? [any-of-enumeration, entity]

E: yes

A: Is the statement about any-of the following: *CLIENT* has *SUCCESS STORY* for *INDUSTRIES*? [any-of-enumeration, path]

E: no

A: Is the statement about any-of the following: *CLIENT* related to *INDUSTRIES*? [any-of-enumeration, path]

E: no

A: Is the statement about any-of the following? [any-of-enumeration, path]

1.*CLIENT* related to *SERVICE* mentioned in *SUCCESS STORY* for *INDUSTRIES*

2.CLIENT has SUCCESS STORY for SERVICE related to INDUSTRIES

3.CLIENT related to SERVICE related to INDUSTRIES

4.CLIENT related to BENEFITS related to INDUSTRIES

E: [no answer]

A: Is the statement about any of the following: NEW TYPE related to INDUSTRIES? [any-of-discriminating-edge, path]

E: yes

A: What is the name of the new type? [name, entity]

E: customer testimonial

A: Could you tell me how CLIENT is related to CUSTOMER TESTIMONIAL? [name, relation]

E: client provides customer testimonial

A: Could you tell me how CUSTOMER TESTIMONIAL is related to INDUSTRIES? [name, relation]

E: customer testimonial belongs to industries

A: Ok. I understand Client Ceridian provides Customer Testimonial belongs to Industries Communication, Media & Technology.

User Impressions of Questions to Acquire Lexical Knowledge

Kazunori Komatani

Osaka University

8-1 Mihogaoka, Ibaraki

Osaka 567-0047, Japan

komatani@sanken.osaka-u.ac.jp nakano@jp.honda-ri.com

Mikio Nakano

Honda Research Institute Japan Co., Ltd.

8-1 Honcho, Wako

Saitama 351-0188, Japan

Abstract

For the acquisition of knowledge through dialogues, it is crucial for systems to ask questions that do not diminish the user's willingness to talk, i.e., that do not degrade the user's impression. This paper reports the results of our analysis on how user impression changes depending on the types of questions to acquire lexical knowledge, that is, explicit and implicit questions, and the correctness of the content of the questions. We also analyzed how sequences of the same type of questions affect user impression. User impression scores were collected from 104 participants recruited via crowdsourcing and then regression analysis was conducted. The results demonstrate that implicit questions give a good impression when their content is correct, but a bad impression otherwise. We also found that consecutive explicit questions are more annoying than implicit ones when the content of the questions is correct. Our findings reveal helpful insights for creating a strategy to avoid user impression deterioration during knowledge acquisition.

1 Introduction

Structured knowledge bases are not only crucial for providing various services such as information search and recommendations but also effective for non-task-oriented dialogue systems to avoid generic or dull responses (Xing et al., 2017; Young et al., 2018; Zhou et al., 2018; Liu et al., 2019). However, it is impractical to presuppose a perfect knowledge base (West et al., 2014) in an early stage of system development.

Therefore, being able to acquire knowledge from users and thereby enhance knowledge bases through dialogues is one of the most important abilities that dialogue systems should possess. Although knowledge acquisition can be done by asking people to input information on GUIs or spreadsheets, knowledge acquisition through dialogues

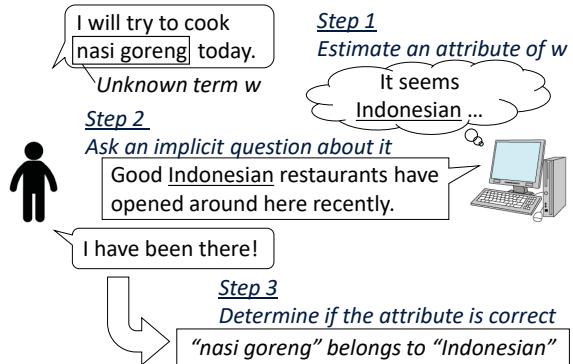


Figure 1: An example of implicit confirmation.

has an advantage in that people can enjoy conversations with the system, especially when the system can engage in non-task-oriented dialogues (Kobori et al., 2016).

One of the targets of knowledge acquisition through dialogues is knowledge about unknown terms and unknown relations between terms by asking the appropriate questions. This would enable the systems to keep learning even when unknown terms appear during dialogues (Meena et al., 2012; Sun et al., 2015).

To enable non-task-oriented systems to acquire a variety of knowledge, the dialogue needs to continue, but this can be a difficult task, as revealed in the Amazon Alexa Prize challenges (Fang et al., 2018; Chen et al., 2018). Users tend to stop interacting with a dialogue system if it repeatedly asks annoying questions, as they do not wish to use the system like an “oracle” who must repeatedly tell it whether a target is correct or wrong (Amershi et al., 2014). Therefore, asking questions for acquiring knowledge should be designed so that they do not irritate the user too much.

For acquiring domain knowledge without asking abrupt questions, the process of implicit confirmation was proposed for non-task-oriented dia-

logue systems (Ono et al., 2016, 2017). Figure 1 shows an example of this process. First, when an unknown term appears in a user utterance, the system estimates its attribute (Otsuka et al., 2013) (Step 1). Second, the system asks an implicit question¹ about the estimated result, instead of asking an explicit question (Step 2). The implicit question is not a superficially interrogative sentence, but it functions as a question by interpreting it along with the subsequent user utterance. Third, the system determines whether or not the estimated result included in the implicit question was correct by also taking the subsequent user response into consideration, and then it adds the estimated result to the system knowledge if it is correct (Step 3). Although these studies assume that implicit questions are less irritating than explicit questions, this has not been empirically verified. Moreover, since the estimated results used in the questions are not always correct, any effect on user impression when the results in the questions are wrong should be considered.

We therefore investigate how system questions for acquiring knowledge affect user impression, including the user’s irritation by asking the extent to which the system utterances were annoying. Here, two research questions, RQ1 and RQ2, are addressed. RQ1 is how the system’s question types affect user impression. The questions consist of five types comprising both explicit and implicit questions, and the correctness of the content of the questions. RQ2 is whether or not consecutive explicit questions for acquiring knowledge are felt as more annoying than consecutive implicit ones. A strategy based upon the results will be discussed in Section 5.

We gathered user impression data after users engaged in a session consisting of several interactions with the system and then analyzed the impression in relation to the question types used in the session. The most naive approach to obtain user impressions is to ask after every system turn, but this would be very annoying and disturb the dialogue flow. Instead, we estimated the effect of each question type in the session by means of a regression model. This model also enables us to analyze user impression when the same question type is repeated.

¹This system utterance was called an implicit *confirmation request* in (Ono et al., 2016, 2017), but in this paper we call it an implicit *question* to clarify their difference in purpose, which will be explained in Section 2.2.

2 Related Work

2.1 Knowledge acquisition in dialogue systems

It has been of great interest that computers continue to learn knowledge autonomously. A famous example is the Never-Ending Language Learner (NELL) (Carlson et al., 2010; Mitchell et al., 2015), which continuously extracts information from the Web. Several methods have been developed for machine learning tasks (such as information extraction) to continuously improve the performance of classifiers in a semi-supervised manner, which is known as life-long learning (Chen and Liu, 2018). We aim to develop systems that can perform such knowledge acquisition through dialogues.

Several studies have investigated how dialogue systems acquire knowledge. Otsuka et al. (2013) proposed a method to estimate the cuisine of an unknown restaurant name from its character sequence and to accordingly change question forms to acquire knowledge. Pappu and Rudnick (2014) designed strategies for asking users questions in a goal-oriented dialogue system and analyzed the acquired knowledge through a user study. Hixon et al. (2015) proposed a method for asking questions to obtain relations between concepts in a question-answering system. Weston (2016) designed ten tasks and demonstrated that supervision given as feedback from simulated interlocutors enables an end-to-end memory network to predict the next utterances better; Li et al. (2017) implemented Weston’s method with reinforcement learning and showed that the system performance improved by asking questions. Mazumder et al. (2019) proposed a system that asks questions about a triple by using knowledge graph completion where a triple (s, r, t) denotes a source entity, a relation, and a target entity, respectively, and lacks either a source s or target t . In these problem settings, it is important to consider how users feel about the system’s questions in order to continue dialogues to acquire a variety of knowledge. As mentioned in Section 1, Ono et al. (2017) proposed implicit questions to avoid decreasing the user’s willingness to talk, but its effect has not been verified through a user study.

2.2 Implicit questions

Implicit questions for non-task-oriented dialogues (Ono et al., 2017) differ from implicit confirma-

	Correct C	Wrong W
Explicit E	EC “Is puttanesca Italian?”	EW “Is puttanesca Japanese?”
Implicit I	IC “Italian is perfect for a date.”	IW “Japanese foods are healthy.”
Whq	Whq “What is puttanesca?”	

Table 1: Examples of five types of system questions for *puttanesca* whose correct cuisine is *Italian*. E and I denote explicit and implicit questions. C and W denote whether the content is correct or wrong. Whq denotes Wh-questions.

tion requests for task-oriented dialogues from the viewpoint of purpose. Implicit confirmation is a well-known technique for task-oriented spoken dialogue systems as a way of handling errors (Bohus and Rudnicky, 2005; Skantze, 2005). A number of studies have focused on changing the form of confirmation requests, including explicit and implicit ones (Bouwman et al., 1999; Komatani and Kawahara, 2000). Consider an example in a flight reservation task where the system tries to determine the destination (going to Boise). The system can ask something like “Are you going to Boise?” as an explicit confirmation request, and it can also continue the dialogue by asking its next question, e.g., “To get to Boise, where will you depart from?”, as an implicit confirmation request. Prior research in task-oriented dialogues has shown that an implicit confirmation request can reduce the number of turns when the content is correct and that it is difficult to correct the system’s misunderstanding when the content is incorrect (Sturm et al., 1999).

The advantage of implicit questions in non-task-oriented dialogues is not the reduction in the number of turns, which is well-known in task-oriented dialogues, but rather that they do not disturb the dialogue flow, which hopefully will decrease the likelihood of the user becoming irritated and stopping the dialogue. User impression, particularly how annoying a question type is, should be investigated in order to enable non-task-oriented systems to continue dialogues, especially when they are utilized by real users. In this paper, we address this issue from the viewpoint of user impression through a user study.

2.3 User impression of dialogues

Several studies have tried to predict user impression of dialogues. Walker et al. (1997) proposed a

framework to predict user satisfaction by means of a regression model using various objective factors during task-oriented dialogues. Higashinaka et al. (2010) developed a method to model user satisfaction transitions using a hidden Markov model even when only user impression scores for entire dialogues were given. Ultes and Minker (2014) and Ultes (2019) improved the prediction accuracy of the interaction quality with various machine learning methods. In contrast, the aim of this paper is *not* to predict user impressions, but rather to analyze the effects of question types on them in dialogues by means of a regression model inspired by (Walker et al., 1997).

3 User Study Design

We assume a system that obtains an attribute value for an unknown term. That is, when an unknown term appears in a dialogue, we try to make the system acquire its attribute from the user through the dialogue. A pair consisting of the term and its attribute can then be stored as new system knowledge.

More specifically, we assume the pair of an unknown food name and its cuisine. First, the cuisine of a food name is estimated from its character sequence (Otsuka et al., 2013), and next, the estimated cuisine is verified by asking either form of question. We focus here on the types of questions for verifying the estimated cuisine.

3.1 Five question types for knowledge acquisition

Table 1 lists the five question types along with examples. The examples correspond to a case where the unknown term is *puttanesca*, its estimated correct cuisine is *Italian*, and its estimated wrong cuisine is *Japanese*.

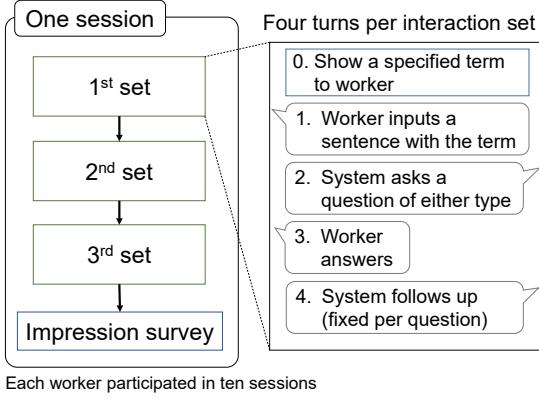


Figure 2: Flow of data collection.

The question types have two components: its question form and the correctness of its content. The first one can be explicit ('E'), implicit ('I'), or a Wh-question ("Whq"). An explicit question explicitly asks whether its content is correct or not through a Yes/No question (e.g., "Is puttanesca Italian?"). An implicit question continues the dialogue with a system utterance containing the estimated cuisine name (e.g., "Italian is perfect for a date.") and then implicitly determines whether the cuisine is correct or not by also considering the subsequent user utterance (Ono et al., 2017). A Wh-question simply asks without using an estimated cuisine (e.g., "What is puttanesca?").

The other component is whether the estimated cuisine is correct or not. We utilize it to investigate any effects on user impression caused by correct or wrong content, which is derived from the automatic estimation about the unknown food name (Otsuka et al., 2013), before the system asks a question. This is applied only to the explicit and implicit questions, as Wh-questions have no concrete content. Thus, C and W, which respectively denote correct and wrong content, are added to E and I explained above, except for Whq. For simplicity, we only consider explicit questions with one choice and do not consider those with multiple choices (Komatani et al., 2016).

3.2 Data collection

We investigated user impression of dialogues including questions of the five types via crowdsourcing.² Crowdworkers were Japanese speakers and thus all the dialogues were in Japanese. We explained that they would talk with an "AI chatbot"

²We used the platform of CrowdWorks, Inc. <https://crowdworks.co.jp/>

1. Were the system utterances annoying?						
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Not annoying				Annoying		
2. Was the system intelligent?						
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Not intelligent				Intelligent		
Submit and continue						

Figure 3: Screenshot of impression survey.

and asked them to talk as if they were meeting for the first time.

The workers gave their impression scores once per session. The flow is depicted in Figure 2. One session consists of three sets of interactions, followed by an impression survey.

Each interaction set consists of four turns: two system turns and two user turns. Before the first turn, a term is displayed as an instruction, e.g., "Please input your thought as if you ate *puttanesca* recently". The four turns flow as follows.

Turn 1: A worker inputs a sentence containing the term specified in the instruction. The term is prepared before the experiment.

Turn 2: The system asks a question about the term as one of the five question types. The question type is randomly selected from the five. Wrong cuisine estimation results and expressions of implicit questions are manually prepared before the experiment.

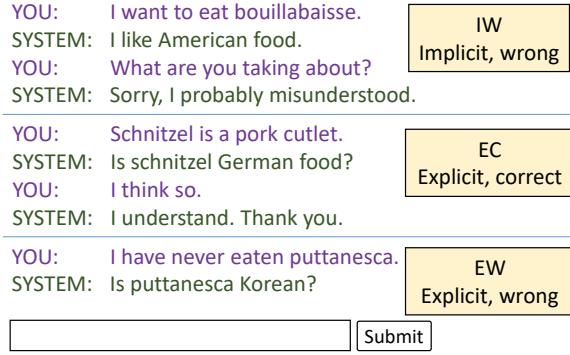
Turn 3: The worker inputs a response to the system question. There is no restriction on the response.

Turn 4: The system's follow-up response is displayed. It depends on the question type³ used in Turn 2. For example, it is "Sorry, I probably misunderstood." for type IW (implicit, wrong).

One interaction set ends after the four turns have finished, and then the next specified term is displayed for the next interaction set.

After engaging in the interaction sets three times, the workers fill in a questionnaire (Figure 3)

³The type selection is random. It is not based on automatic estimation, e.g., determining whether the content is correct or wrong, in this data collection.



Click below to read about puttanesca:
<https://en.wikipedia.org/wiki/Puttanesca>

Figure 4: A screenshot during a third set. The boxes on the right with the question types are only for explanation and were not displayed.

about their impression scores for the session. The questionnaire features 7-point Likert scales for “Were the system utterances annoying?” and “Was the system intelligent?”.⁴ Hereafter, these impression scores are denoted as *annoying* and *intelligent*, respectively.

Each worker was asked to engage in ten sessions. The number of specified terms, which are regarded as unknown terms, was 30, that is, three per session.

Figure 4 shows an example screenshot (translated from Japanese). The lines starting with “YOU” and “SYSTEM” denote a worker’s and the system’s utterances, respectively. The initial part of each interaction set, in which the specified term was shown to workers, is not displayed in the figure, as it disappears when workers input their first sentence. If a worker did not know the term, he or she could check Wikipedia via a link at the bottom of the screen. This was to prevent dialogues in which workers were unaware of the term’s meaning. The dialogues are not very natural, but we used them as the first step for this kind of study, since currently there is no system that can acquire knowledge many times in a natural way.

In total, we obtained 1,183 sessions by 104 workers after removing unusable data (e.g., that of workers who did not finish all ten sessions) from the original 1,319 sessions by 120 workers.⁵

⁴These questionnaire items are unvalidated; they are not captured using redundancy (i.e., different ways of asking the same content) in order to minimize misinterpretations, as argued in (Davis, 1989). We used simple items because they were easiest to explain to the crowdworkers.

⁵Due to a system error, some workers engaged in more than ten sessions.

That is, we obtained 1,183 *annoying* and *intelligent* impression scores corresponding to every session, each of which contains three system question types to be analyzed. There was little agreement among the workers because the impression scores are subjective; some workers gave higher scores overall and others did the opposite. However, there is a certain tendency within each worker’s impression scores for different question types.

4 Analysis with Linear Regression

We analyzed the effect of each question type by using the coefficients of a linear regression model that predicts the collected impression scores. First, we describe the basic regression model and its refinement to make the multiple correlation coefficients (R) higher. After that, we discuss the effect of each question type on user impression and analyze results when the same question types were repeated.

4.1 Linear regression model

A linear regression model was used to predict user impression scores (*annoying* or *intelligent*) from the number of question types used in each session. The basic regression model for the score of the i -th session is given as

$$\text{score}_i = w_0 + \sum_{c \in \{EC, EW, IC, IW, Whq\}} w_c n_i(c), \quad (1)$$

where $n_i()$ denotes the number of each question type c used in the session. The value was 0, 1, 2, or 3 in the basic model.

We applied two refinements to improve the multiple correlation coefficients. First, we normalized impression scores to make their mean 0 and variance 1 per worker. This is effective because each worker gave impression scores in a different range; that is, some gave higher scores on average on the 7-point scale, while others gave lower. As we wanted to know the effect of each question type that had been randomly selected, we used the relative scores given by each worker.

Second, we considered the temporal position of the questions out of the three interaction sets in a session. That is, we used 15 independent variables: the five question types having the three positions each (representing the first, second, and third interaction sets in one session). The refined

	<i>intelligent</i>	<i>annoying</i>
Basic regression model	0.368	0.207
+Normalized per worker	0.493	0.308
+Considering positions	0.540	0.354

Table 2: Multiple correlation coefficients (R) of the models.

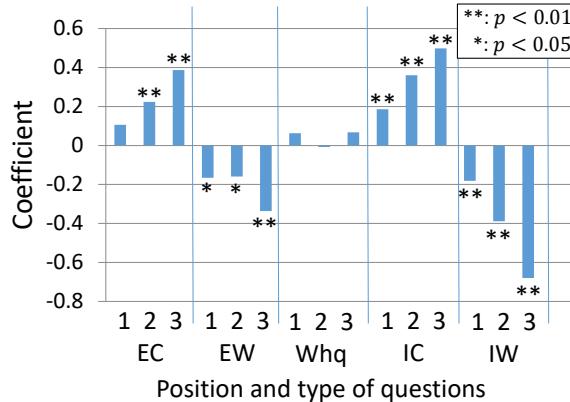


Figure 5: Coefficients of the regression model for *intelligent* when types and positions were considered. The symbols ** and * denote the coefficient is not zero with statistical significance at $p < 0.01$ and $p < 0.05$, respectively.

regression model is given as

$$\text{score}_i = w_0 + \sum_d w_d n_i(d), \quad (2)$$

where $d \in \{EC, EW, IC, IW, Whq\} \times \{1, 2, 3\}$, and $n_i()$ denotes the number of each question type with the position d . It is thus binary in this refined model.

The multiple correlation coefficients for the two impression scores are listed in Table 2. The coefficients became higher by the normalization, and became even higher by considering the temporal positions. Thus, in the following analysis, we use the model with these 15 coefficients considering the positions after the normalization per worker.

The table also shows that the *intelligent* scores had a better fit to the collected data. Since the two impression scores had almost the reverse tendency, either will be used in the following sections for brevity.

4.2 Analysis of obtained coefficients

RQ1 is addressed here: “how the system’s question types affect user impression”. Figure 5 shows the values of the 15 coefficients obtained for *intelligent*, which fitted the data better. We also tested the statistical significance of individual regression

	EC	EW	Whq	IC	IW
<i>intelligent</i>	0.24	-0.22	0.04	0.35	-0.42
<i>annoying</i>	-0.13	0.08	-0.02	-0.21	0.28

Table 3: Averages over the three positions of the coefficients in the regression models.

coefficients that verifies whether or not the coefficient is zero; these results are shown as well. Larger positive values indicate that the question type in that position tends to give a better impression to workers, that is, they felt the system was more intelligent. Larger negative values indicate the opposite.

The averages over the three positions for *intelligent* are summarized in Table 3, along with those for *annoying*. The coefficients of the types are ordered as

$$IC > EC > Whq > EW > IW$$

for *intelligent*, and

$$IC < EC < Whq < EW < IW$$

for *annoying*. The two impression scores showed the reverse order.

Details follow using the case of *intelligent*. The coefficients of IC and EC, both of which had correct content, were positive, and those of EW and IW, both of which had wrong content, were negative. This result corresponds to our intuition that workers feel the system is not intelligent when it asks questions with the wrong content. The coefficient of Whq was in-between, as it had no concrete content.

Next, we focus on the relationship between the explicit and implicit questions. When they had correct content, the coefficients of the implicit questions (IC) were larger than those of the explicit questions (EC). This result indicates that the implicit questions give a better impression than the explicit ones. This is because the workers felt the system knew rare and difficult terms; the impression scores were higher when the target food names seemed more uncommon. In contrast, when they had wrong content, the coefficients of the explicit questions (EW) were less negative than those of the implicit questions (IW). In other words, if the estimated cuisine was wrong, the explicit questions caused less damage to user impression than the implicit ones. This is probably because the workers felt the system ignored their previous utterances and selfishly started a new topic

when an implicit question was asked with a wrong cuisine.

Figure 5 also shows the tendency among three temporal positions of each question type. In the cases of both negative and positive coefficients, they were the largest at the third positions for all five types. This suggests that the question type just before the impression survey might have the largest effect on the impression scores.

4.3 Impression when the same question type is repeated

This section addresses RQ2: “whether or not consecutive explicit questions are considered more annoying than implicit ones”. Here, the impression scores for *annoying* are used, as the purpose of RQ2 is to investigate whether the consecutive questions are annoying or not.

We compare the following two impression scores for the case where the same question type is repeated.

- Actual scores when same question type was repeated three times
- Predicted scores by regression model

By comparing the two scores, we can analyze the difference between impression when the same question type was actually repeated and that when the question type was used with various contexts.

Specifically, the former scores were calculated by averaging the scores of the sessions where the same question types were actually repeated as a result of random selection. Such cases occurred 10.4 times on average per question type in the collected data. On the other hand, the latter scores were calculated with the model of Eq. (2) for the cases when a question type was used three times. Its coefficients were obtained using data where each question type was randomly selected, that is, without considering whether the same question types were repeated or not. We can thus regard them as averages over the cases when the five question types appeared in various contexts.

Figure 6 shows the results and Table 4 lists their concrete values. For all question types, the impression scores for the actual cases were larger, i.e., more annoying, than those for the predicted cases. Furthermore, the differences in the scores for types with wrong content (EW and IW) were larger than those with correct content (EC and IC), as shown in the “Difference” column in Table 4.

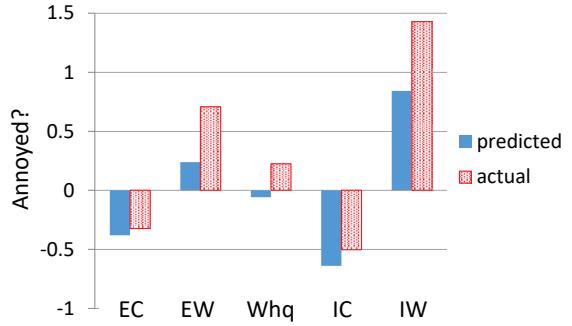


Figure 6: Scores for *annoying* (predicted vs. actual).

	Predicted	Actual	Difference
EC	-0.380	-0.324	+0.056
EW	0.239	0.707	+0.468
Whq	-0.058	0.224	+0.282
IC	-0.639	-0.502	+0.137
IW	0.842	1.429	+0.587

Table 4: Predicted and actual impression scores for *annoying* when the same question types are repeated three times.

This corresponds to our intuition that consecutive questions with wrong content would be more annoying than consecutive ones with correct content.

The results also show that the scores of IC were more negative, i.e., less annoying, than those of EC in both the predicted and actual cases. Although the degrees of “not annoying” diminished in the actual cases, the order between them did not change. This means that, when the content is correct, consecutive explicit questions (EC) are more annoying than consecutive implicit ones (IC). This result answers RQ2; it is true when the content is correct.

One reason for these results is that all explicit questions had almost the same simple form (e.g., “Is puttanesca Italian?”), and this made users feel as if the system repeated such questions without thinking. In contrast, implicit questions disturbed the dialogue flow less and thus were less annoying, as they contained the content of the previous user utterance when the cuisine in it was correct.

Example dialogues of sessions when explicit and implicit questions with correct content (EC and IC) were asked three times are respectively shown in Figures 7 and 8.

5 Conclusion

We investigated how five types of question affect user impression and clarified the order of the ques-

User	The <i>panna cotta</i> was very sweet and good.
System	Is panna cotta Italian?
User	Yes. It's Italian.
System	I understand. Thank you.
User	The <i>pad thai</i> was refreshing and delicious.
System	Is pad thai a Thai food?
User	That's right. It's a Thai food.
System	I understand. Thank you.
User	I ate a <i>bagna cauda</i> . It was similar to shabu-shabu.
System	Is bagna cauda Italian?
User	Yes. It is an Italian pot cuisine.
System	I understand. Thank you.

Figure 7: Example session of consecutive explicit questions with correct content (EC).

tion types in terms of two impression scores: *intelligent* and *annoying*. We also found that repeating the same question type annoys the user and degrades user impression even when the content is correct. The implicit questions tend to give a better user impression than the explicit ones as long as the content is correct. On the other hand, the impression by the implicit questions with wrong content is worse.

These results justify an adaptive strategy using a confidence measure of the cuisine estimation in order to lower the risk of user impression deterioration. Specifically, it is reasonable that the system should select an implicit question if the confidence is high, and select an explicit one (or a Wh-question) if the confidence score is low. This is consistent with results discussed for task-oriented dialogues using confidence measures of automatic speech recognition, such as (Sturm et al., 1999).

A new finding here, based on the results of our analysis in Section 4.3, is that the designer of the dialogue system also needs to avoid repeating the same type of questions in non-task-oriented dialogues. The system should have multiple choices of question types in order to prevent users from becoming irritated. That is, it is necessary to change question types appropriately by considering not only the confidence of the estimation but also the history of the dialogue. This will help the dialogue to continue with less degradation of the user's impression and enable the system to acquire knowledge through dialogues.

Several issues remain as future work. Our experiment was limited in terms of the number of turns and the domain where it was tested. The results need to be verified with non-task-oriented systems that can engage in longer dialogues in various domains. We are planning to implement

User	<i>Gelato</i> looks gorgeous.
System	Ice confections are perfect for eating on a hot day.
User	That's right. It's refreshing.
System	I think so too.
User	I ate <i>paella</i> for dinner.
System	Many Japanese people like Spanish food.
User	Japanese people like such taste.
System	I think so too.
User	<i>Ushiojiru</i> is a nice, tasty soup.
System	Many Japanese dishes are lightly seasoned.
User	That's right. They're so good.
System	I think so too.

Figure 8: Example session of consecutive implicit questions with correct content (IC).

a non-task-oriented dialogue system that has the function to acquire knowledge. The subdialogue shown in this paper can be embedded within a longer dialogue. The implicit confirmation can be implemented by preparing the expressions of implicit questions for each category (cuisine type, in this paper) to be estimated. A further user study will be conducted with the implemented system. Another issue is that answers from users may be different; e.g., some users may say that “mapo doufu” is Sichuan, but others may say it is Chinese. This is caused by the different concept granularity of individual users, which appears in the answers. A knowledge graph that can have different nodes representing the both concepts may be a possible solution for this issue. Incorporating the utility of each question type for acquiring knowledge (Komatani et al., 2016) would be another interesting extension of the strategy.

Acknowledgments

This work was partly supported by JSPS KAKENHI Grant Numbers JP16H02869 and JP19H04171.

References

- Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4).
- Dan Bohus and Alexander Rudnicky. 2005. Error handling in the RavenClaw dialog management architecture. In *Proc. Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP)*, pages 225–232.
- Gies Bouwman, Janienke Sturm, and Lou Boves. 1999. Incorporating confidence measures in the Dutch

- train timetable information system developed in the ARISE project. In *Proc. IEEE International Conference on Acoustics, Speech & Signal Processing (ICASSP)*.
- Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proc. Conference on Artificial Intelligence (AAAI)*.
- Chun-Yen Chen, Dian Yu, Weiming Wen, Yi Mang Yang, JiaPing Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhownick, Shreenath Iyer, Giritheja Sreenivasulu, Runxiang Cheng, Ashwin Bhandare, and Zhou Yu. 2018. Gunrock: Building a human-like social bot by leveraging large scale real user data. In *2nd Proceedings of Alexa Prize (Alexa Prize 2018)*.
- Zhiyuan Chen and Bing Liu. 2018. *Lifelong Machine Learning, Second Edition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.
- Fred D. Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340.
- Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark, Ari Holtzman, Yejin Choi, Noah A. Smith, and Mari Ostendorf. 2018. Sounding board: A user-centric and content-driven social chatbot. In *Proc. North American Chapter of Association for Computational Linguistics (NAACL)*, pages 96–100.
- Ryuichiro Higashinaka, Yasuhiro Minami, Kohji Dohsaka, and Toyomi Meguro. 2010. Modeling user satisfaction transitions in dialogues from overall ratings. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, page 18–27.
- Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning knowledge graphs for question answering through conversational dialog. In *Proc. North American Chapter of Association for Computational Linguistics (NAACL)*, pages 851–861.
- Takahiro Kobori, Mikio Nakano, and Tomoaki Nakamura. 2016. Small talk improves user impressions of interview dialogue systems. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 370–380.
- Kazunori Komatani and Tatsuya Kawahara. 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. International Conference on Computational Linguistics (COLING)*, pages 467–473.
- Kazunori Komatani, Tsugumi Otsuka, Satoshi Sato, and Mikio Nakano. 2016. Question selection based on expected utility to acquire information through dialogue. In *Proc. International Workshop on Spoken Dialogue System Technology (IWSDS)*, pages 27–38.
- Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc’Aurelio Ranzato, and Jason Weston. 2017. Learning through dialogue interactions by asking questions. In *Proc. International Conference on Learning Representations (ICLR)*.
- Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. 2019. Knowledge aware conversation generation with explainable reasoning over augmented graphs. In *Proc. Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1782–1792.
- Sahisnu Mazumder, Bing Liu, Shuai Wang, and Nianzu Ma. 2019. Lifelong and interactive learning of factual knowledge in dialogues. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 21–31.
- Raveesh Meena, Gabriel Skantze, and Joakim Gustafson. 2012. A data-driven approach to understanding spoken route directions in human-robot dialogue. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 226–229.
- T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proc. Conference on Artificial Intelligence (AAAI)*.
- Kohei Ono, Ryu Takeda, Eric Nichols, Mikio Nakano, and Kazunori Komatani. 2016. Toward lexical acquisition during dialogues through implicit confirmation for closed-domain chatbots. In *Proc. of Second Workshop on Chatbots and Conversational Agent Technologies (WOCHAT)*.
- Kohei Ono, Ryu Takeda, Eric Nichols, Mikio Nakano, and Kazunori Komatani. 2017. Lexical acquisition through implicit confirmations over multiple dialogues. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 50–59.
- Tsugumi Otsuka, Kazunori Komatani, Satoshi Sato, and Mikio Nakano. 2013. Generating more specific questions for acquiring attributes of unknown concepts from users. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 70–77.
- Aasish Pappu and Alexander Rudnicky. 2014. Knowledge acquisition strategies for goal-oriented dialog systems. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 194–198.

Gabriel Skantze. 2005. Galatea: a discourse modeller supporting concept-level error handling in spoken dialogue systems. In *Proc. SIGdial Workshop on Discourse and Dialogue*, pages 178–189.

Janienke Sturm, Els den Os, and Lou Boves. 1999. Issues in spoken dialogue systems: Experiences with the Dutch ARISE system. In *Proc. ESCA Workshop on Interactive Dialogue in Multi-Modal Systems*, pages 1–4, Kloster Irsee, Germany.

Ming Sun, Yun-Nung Chen, and Alexander I. Rudnicky. 2015. Learning OOV through semantic relatedness in spoken dialog systems. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1453–1457.

Stefan Ultes. 2019. Improving interaction quality estimation with BiLSTMs and the impact on dialogue policy learning. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 11–20.

Stefan Ultes and Wolfgang Minker. 2014. Interaction quality estimation in spoken dialogue systems using hybrid-hmms. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 208–217.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In *Proc. Annual Meeting of the Association for Computational Linguistics and Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL)*, pages 271–280.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *Proc. International Conference on World Wide Web (WWW)*, pages 515–526.

Jason Weston. 2016. Dialog-based language learning. In *Proc. International Conference on Neural Information Processing Systems (NIPS)*, pages 829–837.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proc. Conference on Artificial Intelligence (AAAI)*, page 3351–3357.

Tom Young, Erik Cambria, Iti Chaturvedi, Minlie Huang, Hao Zhou, and Subham Biswas. 2018. Augmenting end-to-end dialog systems with common-sense knowledge. In *Proc. Conference on Artificial Intelligence (AAAI)*, pages 4970–4977.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4623–4629.

Simulating Turn-Taking in Conversations with Delayed Transmission

Thilo Michael

Quality and Usability Lab

Technische Universität Berlin

thilo.michael@tu-berlin.de

Sebastian Möller

Quality and Usability Lab

Technische Universität Berlin

German Research Center for AI (DFKI)

sebastian.moeller@tu-berlin.de

Abstract

Conversations over the telephone require timely turn-taking cues that signal the participants when to speak and when to listen. When a two-way transmission delay is introduced into such conversations, the immediate feedback is delayed, and the interactivity of the conversation is impaired. With delayed speech on each side of the transmission, different *conversation realities* emerge on both ends, which alters the way the participants interact with each other. Simulating conversations can give insights on turn-taking and spoken interactions between humans but can also be used for analyzing and even predicting human behavior in conversations. In this paper, we simulate two types of conversations with distinct levels of interactivity. We then introduce three levels of two-way transmission delay between the agents and compare the resulting interaction-patterns with human-to-human dialog from an empirical study. We show how the turn-taking mechanisms modeled for conversations without delay perform in scenarios with delay and identify to which extent the simulation is able to model the delayed turn-taking observed in human conversation.

1 Introduction

Turn-taking in human conversations has proven to be influenced by many auditory, visual, and contextual cues. Especially in telephone conversations, where no visual cues are present, people rely on the immediacy of signals in prosody and content to perform smooth and uninterrupted turn-taking. Investigating the influence of delay on conversations has been a focus in telephone quality research for a long time, where the goal is to study how degradations of packet-switched VoIP-transmissions influence the conversation structure and thus, the perceived quality ([ITU-T Recommendation P.805, 2007](#); [ITU-T Recommandation G.107, 2011](#)). But

also in the field of human-computer interaction, where Spoken Dialogue Systems (SDS) with realistic turn-taking have become feasible, it is of interest to study how humans interact and react to delayed voice transmission.

It has been shown that the perception of changes in transmission time not only depends on the duration of the delay but that the effects on the conversations also vary with the type of conversation itself ([Hammer et al., 2005](#)). Concretely, conversations with lower interactivity, i.e., slower speaker alternation rate and less turn-taking, are not as prone to be affected by transmission delay than conversations with higher interactivity. Simulating a conversation does not only give insights into the interactivity patterns that arise during a conversation but can also be used to predict events and behaviors. In such a simulation, two dialog systems exchange information through a speech channel. Information is processed in increments to allow for a turn-taking mechanism and structured dialog ([Michael and Möller, 2020](#)).

In this paper, we present a simulation with different levels of interactivity and evaluate how a probability-based turn-taking function models the behavior in conversations under the influence of transmission delay. For this, we simulate two different goal-oriented conversation scenarios standardized by the ITU, namely the Short Conversation Test (SCT) with a low conversational interactivity and the Random Number Verification test (RNV) with a high conversational interactivity ([ITU-T Recommandation P.805, 2007](#)). We simulate conversations with 0ms, 800ms, and 1600ms delay and compare metrics of interactivity like speaker alternation rate, gaps, overlaps, and pauses, as well as unintended interruption rates to human-to-human conversations with the same delay conditions.

2 Related Work

Turn-taking in conversations is a long-studied phenomenon (Sacks et al., 1974), with recent work focusing on human turn-taking behavior in conversations (Lunsford et al., 2016), end-of-turn prediction (Liu et al., 2017; Skantze, 2017) and rule-based turn-taking models (Selfridge and Heeman, 2012; Baumann, 2008; Michael and Möller, 2020). While the effects of transmission delay on turn-taking conversations have been studied in the field of speech transmission quality (Kitawaki and Itoh, 1991; Egger et al., 2010), it has to the best of our knowledge not been modeled. However, the influence of delay on the perception of the conversational quality has been modeled by the E-model (ITU-T Recommendation G.107, 2011).

Due to the delayed arrival of turn-taking signals, transmission delay affects the flow of a conversation (Hammer, 2006). However, the degree to which turn-taking and the interactivity of a conversation is degraded also depends on the interactivity of the conversation itself (Raake et al., 2013; Egger et al., 2012). To evaluate those dependencies, conversation tests with distinct levels of conversational interactivity (CI) have been standardized, during which participants perform goal-oriented tasks with an interlocutor. One prominent conversation test with a high CI is the RNV test, where participants alternately exchange a list of numbers organized in 4 blocks (Kitawaki and Itoh, 1991). An example of a conversation test with low CI is the SCT, where participants solve real-world tasks like ordering pizza or booking a flight.

Parametric Conversation Analysis (P-CA) is a framework to assess the structure of conversations programmatically (Hammer, 2006). With an independent voice activity detection of the two speakers, four conversation states can be derived: *M* (“mutual silence”), *D* (“double talk”), *A* (“speaker A”) and *B* (“speaker B”) (Lee and Un, 1986; ITU-T Recommendation P.59, 1993). Based on these four states interactivity metrics like the speaker alternation rate (SAR), interruption rate (IR), as well as turn-taking information like gaps and overlaps between speaker turns, can be calculated (Hammer et al., 2005; Lunsford et al., 2016). For delayed conversations, the unintended interruption rates (UIR) measures the number of interruptions that were caused by the delay and were not intended to be interrupting the interlocutor (Egger et al., 2010).

As conversation simulations focus on turn-

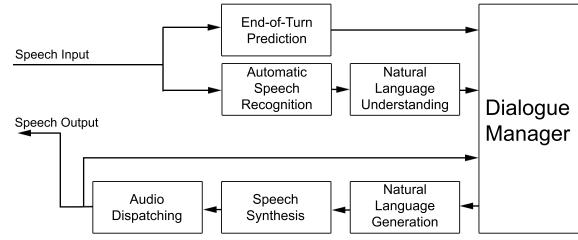


Figure 1: Schematic of an incremental spoken dialogue network containing parts for speech understanding and end-of-turn prediction on the top, and dialogue managing unit on the right and the speech generation and audio dispatching on the bottom.

taking, they need to respond to incoming signals in a timely manner and thus need to process data *incrementally*. The incremental processing on the scale of a complete dialogue system has been proposed by Skantze and Schlangen (Schlangen and Skantze, 2011) and implemented in InproTK (Baumann and Schlangen, 2012) and Retico (Michael and Möller, 2019).

3 Simulation Setup and Turn-Taking

The simulation is based on a set of conversation tests carried out with 58 untrained participants who were 18 to 71 years of age (M: 32, SD: 13.48), of which 48.2 percent identified as female. During the experiments, each pair of participants carried out SCT and RNV conversations with end-to-end one-way transmission delays of 0ms, 800ms, and 1600ms, resulting in 174 recorded conversations. For the simulation, one scenario was selected from each conversation type, and 20 SCT conversations and 20 RNV conversations at 0ms delay were annotated with dialogue acts, transcripts, and turn-taking information. 20 different conversations from each conversation type were used to evaluate the simulation.

The simulation was implemented using the incremental processing pipeline of the retico framework (Michael and Möller, 2019). It consists of two spoken dialogue systems (agents) that are connected through a simulated transmission network that is able to introduce delay to both agents. A schematic view of the incremental modules of one agent in the simulation is shown in Figure 1. The speech input and output, as well as natural language understanding modules, are created by specifically recognizing the annotated empirical conversations. Language generation and synthesis is handled by

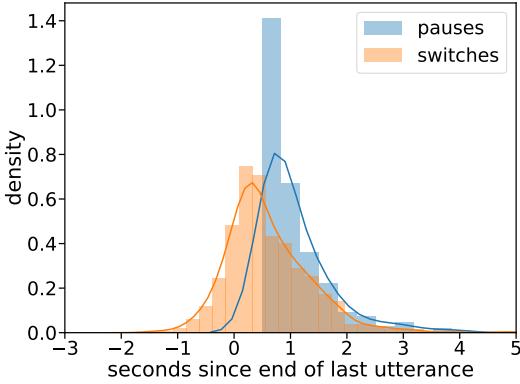


Figure 2: Distribution of speaker switches (orange) and pauses/turn-keeps (blue) in SCT conversations without delay as measured by the seconds since or until the end of the last utterance.

transmitting utterances cut from the empirical data so that the length and content of the utterances match. An end-of-turn prediction module predicts the time until the end of the utterance, and an audio dispatching module reports the progress of the current utterance to the dialogue manager of the same agent. The dialogue manager uses agenda-based dialog management to fulfill the goal-oriented tasks of the SCT and RNV scenarios, and it also handles turn-taking.

The turn-taking of the agents in the simulation is modeled by probability distributions that are based on the work by Lunsford et al. (Lunsford et al., 2016). We calculated the distributions of turn-switches (*gaps* and *overlaps*) as well as turn-keeping (*pauses*) as shown in Figure 2. These distributions are measured respective to the end of the last utterance so that negative values correspond to double-talk, and positive values correspond to mutual silence. The cumulative distribution of the pauses and switches were fitted with a logistic regression and inverted to form a model for turn-switches (Equation 1) and turn-keeping (Equation 2).

$$0.27 - 0.322581 \log\left(\frac{1}{r} - 1\right) \quad (1)$$

$$1.10641 - 0.161705 \log\left(\frac{1}{r} - 1\right) \quad (2)$$

By selecting $r \in [0, 1]$ randomly from a uniform distribution and treating switches and pauses as equal alternatives, the agent in the simulation can perform turn-taking in the simulation. Depend-

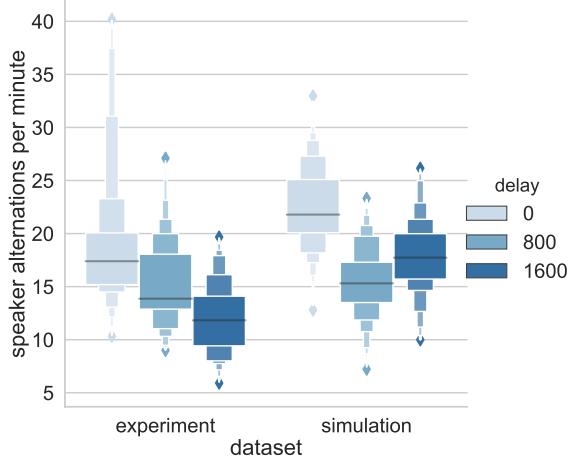


Figure 3: Speaker alternation rate for empirical and simulated SCT conversations at 0, 800, and 1600 ms delay.

ing on which agent is currently speaking, the dialogue manager decides when to make a pause or a speaker switch. This way, the models of pauses and switches compete at every end of a turn. To prevent prolonged interruption (e.g., when both agents start speaking at the same time), the dialogue manager stops the speech production when double talk occurs in the middle of utterances.

4 Results and Discussion

To evaluate the simulation approach, we simulated 100 RNV and 100 SCT conversations, each with 0, 800, and 1600 ms transmission delay. This results in 600 simulated conversations that we compare to the 174 conversations recorded in the experiment.

The comparison of the states of the SCT conversations (Figure 4) and RNV conversations (Figure 5) shows that the distinct levels of interactivity between these two types of conversations are also visible in the simulated conversation. When introducing delay, the state probabilities of the empirical data and the simulated conversation for mutual silence, speaker a and speaker b show similar changes. However, these effects stagnate for the simulated conversations at 1600 ms. This can also be seen when comparing the speaker alternation rate (Figure 3) of the simulated SCT conversations. There, the drop in speaker alternations due to increased delay is visible for 800ms but increases again for 1600ms, contrary to the behavior of the empirical conversations. This seems to indicate changes in the turn-taking behavior with an increased level of transmission delay.

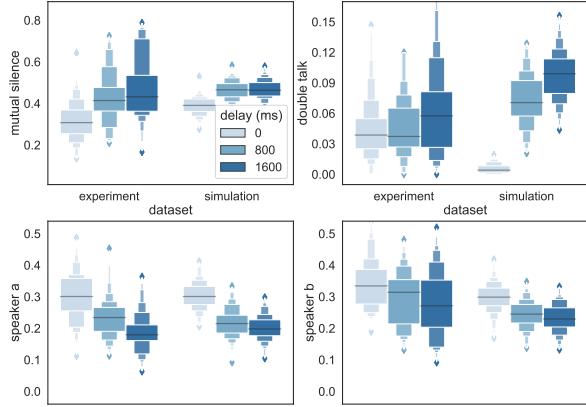


Figure 4: Comparison of states probabilities mutual silence, double talk, speaker a and speaker b between the empirical and simulated SCT conversations at 0, 800, and 1600 ms delay.

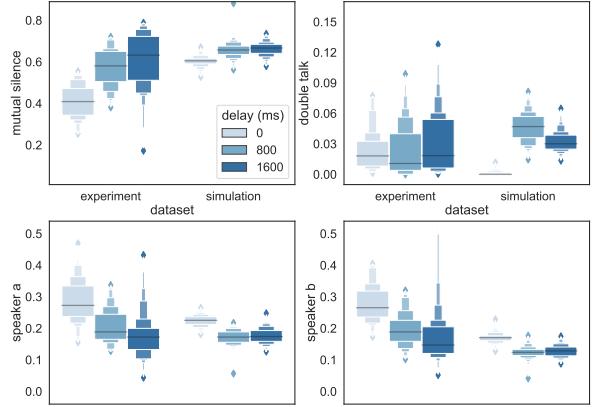


Figure 5: Comparison of states probabilities mutual silence, double talk, speaker a and speaker b between the empirical and simulated RNV conversations at 0, 800, and 1600 ms delay.

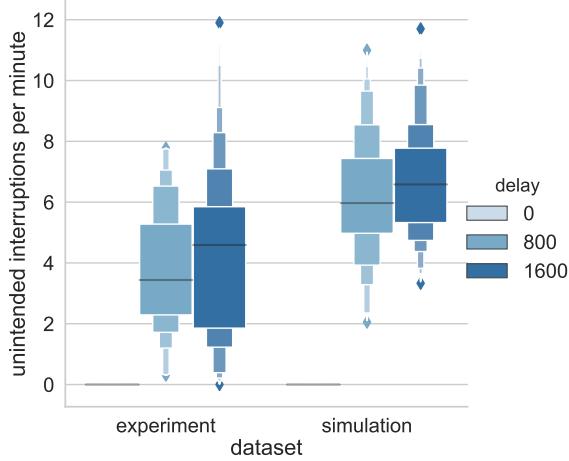


Figure 6: Unintended interruption rate for empirical and simulated SCT conversations at 0, 800, and 1600 ms delay.

While the state probabilities for double talk stay almost constant for SCT and RNV empirical conversations, it increases strongly in the simulations. It also stagnates at 1600ms delay for RNV conversations (Figure 5). The mismatch in double talk in conversations without delay might stem from errors in the end-of-turn prediction, where the model is too pessimistic in the prediction of the end of an utterance.

In general, the simulations seem to have less variance in almost all metrics (state probabilities, speaker alternation rate, interruption rates). One reason for that might be the limited amount of possible utterances that are available in the simulation, resulting in less variance.

Figure 6 shows the unintended interruption rate (UIR), i.e., the interruptions that are caused by de-

lay and were not intended by the interrupting participant. While the increase in UIR is visible for empirical as well as simulated conversations, the number of unintended interruptions in the simulations is generally higher.

5 Conclusion

In this work, we modeled human turn-taking based on the distribution of turn-switches and -pauses. We applied this model in a conversation simulation. We evaluated how well the interactivity of real-world conversations with distinct levels of interactivity and different transmission delay can be modeled with this approach. The simulated conversations show the distinction between the interactivity of RNV and SCT scenarios as well as differences in speaker alternations and interruptions when introducing transmission delay. However, the influence of delay on turn-taking in the simulations seems to saturate with high delay levels. This might hint to a change in turn-taking behavior when large amounts of delay are present.

In future work, we plan to identify the changes in turn-taking behavior and model them based on events in the conversation (e.g., continued interruptions). We also plan to evaluate the proposed turn-taking model for the use in spoken dialogue systems.

Acknowledgments

This work was financially supported by the German Research Foundation DFG (grant number MO 1038/23-1).

References

- Timo Baumann. 2008. Simulating Spoken Dialogue With A Focus on Realistic Turn-Taking. *13th ESS-LI Student Session*, pages 17–25.
- Timo Baumann and David Schlangen. 2012. The IN-PROTK 2012 release. In *NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*, pages 29–32. Association for Computational Linguistic.
- Sebastian Egger, Raimund Schatz, and Stefan Scherer. 2010. It Takes Two to Tango - Assessing the Impact of Delay on Conversational Interactivity on Perceived Speech Qualit. In *Eleventh Annual Conference of the International Speech Communication Association*, pages 1321–1324. ISCA.
- Sebastian Egger, Raimund Schatz, Katrin Schoenenberg, Alexander Raake, and Gernot Kubin. 2012. Same but different? — Using speech signal features for comparing conversational VoIP quality studies. In *Communications (ICC), 2012 IEEE International Conference on*, pages 1320–1324. IEEE.
- Florian Hammer. 2006. *Quality Aspects of Packet-Based Interactive Speech Communication*. Forschungszentrum Telekommunikation Wien.
- Florian Hammer, Peter Reichl, and Alexander Raake. 2005. *The well-tempered conversation: interactivity, delay and perceptual VoIP quality*. In *IEEE International Conference on Communications*, volume 1, pages 244–249. Institute of Electrical and Electronics Engineers (IEEE).
- ITU-T Recommandation G.107. 2011. *The E-model: a computational model for use in transmission planning*. International Telecommunication Union, Geneva.
- ITU-T Recommendation P.59. 1993. *Artificial Conversational Speech*. International Telecommunication Union.
- ITU-T Recommendation P.805. 2007. *Subjective Evaluation of Conversational Quality*. International Telecommunication Union, Geneva.
- Nobuhiko Kitawaki and Kenzo Itoh. 1991. Pure delay effects on speech quality in telecommunications. *IEEE Journal on selected Areas in Communications*, 9(4):586–593.
- H Lee and C Un. 1986. A study of on-off characteristics of conversational speech. *IEEE Transactions on Communications*, 34(6):630–637.
- Chaoran Liu, Carlos Ishi, and Hiroshi Ishiguro. 2017. Turn-Taking Estimation Model Based on Joint Embedding of Lexical and Prosodic Contents. In *Proc. Interspeech 2017*, pages 1686–1690.
- Rebecca Lunsford, Peter A Heeman, and Emma Renne. 2016. Measuring Turn-Taking Offsets in Human-Human Dialogues. In *Proceedings of INTERSPEECH*, pages 2895–2899.
- Thilo Michael and Sebastian Möller. 2019. Retico: An open-source framework for modeling real-time conversations in spoken dialogue systems. In *30th Konferenz Elektronische Sprachsignalverarbeitung (ESSV)*, pages 238–245, Dresden. TUDpress.
- Thilo Michael and Sebastian Möller. 2020. Simulating turn-taking in conversations with varying interactivity. In *31th Konferenz Elektronische Sprachsignalverarbeitung (ESSV)*, pages 208–215, Dresden. TUDpress.
- Alexander Raake, Katrin Schoenenberg, Janto Skowronek, and Sebastian Egger. 2013. Predicting speech quality based on interactivity and delay. In *Proceedings of INTERSPEECH*, pages 1384–1388.
- Harvey Sacks, Emanuel Schegloff, and Gail Jefferson. 1974. *A Simplest Systematics for the Organization of Turn-Taking for Conversation*. *Language*, 50(4):696–735.
- David Schlangen and Gabriel Skantze. 2011. A General, Abstract Model of Incremental Dialogue Processing. *Dialogue and Discourse*, 2(1):83–111.
- Ethan O Selfridge and Peter A Heeman. 2012. A temporal simulator for developing turn-taking methods for spoken dialogue systems. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 113–117. Association for Computational Linguistics.
- Gabriel Skantze. 2017. Towards a General, Continuous Model of Turn-taking in Spoken Dialogue using LSTM Recurrent Neural Networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 220–230.

Is this Dialogue Coherent? Learning from Dialogue Acts and Entities

Alessandra Cervone and Giuseppe Riccardi

Signals and Interactive Systems Lab, University of Trento, Italy

{alessandra.cervone, giuseppe.riccardi}@unitn.it

Abstract

In this work, we investigate the human perception of coherence in open-domain dialogues. In particular, we address the problem of annotating and modeling the coherence of next-turn candidates while considering the entire history of the dialogue. First, we create the Switchboard Coherence (SWBD-Coh) corpus, a dataset of human-human spoken dialogues annotated with turn coherence ratings, where next-turn candidate utterances ratings are provided considering the full dialogue context. Our statistical analysis of the corpus indicates how turn coherence perception is affected by patterns of distribution of entities previously introduced and the Dialogue Acts used. Second, we experiment with different architectures to model entities, Dialogue Acts and their combination and evaluate their performance in predicting human coherence ratings on SWBD-Coh. We find that models combining both DA and entity information yield the best performances both for response selection and turn coherence rating.

1 Introduction

Dialogue evaluation is an unsolved challenge in current human-machine interaction research. This is particularly true for open-domain conversation, where compared to task-oriented dialogue (i.e., restaurant reservations), we do not have a finite set of entities and intents, and speakers' goals are not defined *a priori*. In this work, we address the problem of dialogue evaluation from the perspective of dialogue *coherence* and how this concept can be formalized and evaluated. Our approach could be applied to both task-oriented and non-task-oriented dialogue.

Coherence in language, i.e., the property which determines that a given text is a logical and consistent whole rather than a random collection of sentences, is a complex multifaced concept which has

been defined in different ways and to which several factors contribute (Redeker, 2000), e.g., rhetorical structure (Hobbs, 1979), topics discussed, and grounding (Traum, 1994).

While much recent work has focused on coherence for response generation (Serban et al., 2016; Li et al., 2016; Yi et al., 2019), we argue that there is still much to be understood regarding the mechanisms and substructures that affect human perception of dialogue coherence. In our approach, in particular, we are interested in studying the patterns of distribution of entities and Dialogue Acts (DAs), in regards to dialogue coherence.

Approaches to coherence based on entities have been studied extensively by the Natural Language Processing literature (Joshi and Kuhn, 1979; Grosz et al., 1995), especially in text (e.g., news, summaries). Coherence evaluation tasks proposed by this literature (Barzilay and Lapata, 2008) have the advantage of using weakly supervised training methodologies, but mainly considering documents as-a-whole, rather than evaluating coherence at the utterance level. The dialogue literature (Sacks and Jefferson, 1995; Schegloff, 1968), on the other hand, has focused mainly on coherence in connection to DAs, a generalized version of intents in dialogue (e.g., *yes-no-question*, *acknowledgement*). Recent work (Cervone et al., 2018), in particular, showed the importance of both DAs and entities information for coherence modeling in dialogue. However, even in this case dialogue coherence was rated for entire dialogues rather than studying turn coherence structures.

In this work, we investigate underlying conversation turn substructures in terms of DA and entity transitions to predict turn-by-turn coherence in dialogue. We start by annotating a corpus of spoken open-domain conversations with turn coherence ratings, the Switchboard Coherence corpus (SWBD-

Coh)¹, and perform an analysis of the human perception of coherence in regards to DAs and entities. A multiple regression analysis shows the importance of both types of information for human rating of coherence. Secondly, we present novel neural models for turn coherence rating that combine DAs and entities and propose to train them using response selection, a weakly supervised methodology. While previous work on response selection (Lowe et al., 2017; Yoshino et al., 2019) is mainly based on using the entire text as input, we deliberately choose to use only entities and DAs as input to our models, in order to investigate entities and DAs as a signal for turn coherence. Finally, we test our models on the SWBD-Coh dataset to evaluate their ability to predict turn coherence scores².

The main contributions of this work are:

- creating the Switchboard Coherence corpus, a novel human-annotated resource with turn coherence ratings in non-task-oriented open-domain spoken conversation;
- investigating human perception of coherence in spoken conversation in relation to entities and DAs and their combination;
- proposing novel neural coherence models for dialogue relying on entities and DAs;
- exploring response selection as a training task for turn coherence rating in dialogue.

2 Related work

Coherence evaluation in text Coherence models trained with weakly supervised methodologies were first proposed for text with applications to the news domain and summarization (Barzilay and Lapata, 2008). These models rely on the entity grid, a model that converts the entities (Noun Phrases) mentioned in the text to a sentence-by-sentence document representation in the form of a grid. The tasks on which coherence models in this line of research are usually evaluated are *sentence ordering* (Barzilay and Lapata, 2008), i.e., ranking original documents as more coherent than the same documents with the order of all sentences randomly permuted, and *insertion*, i.e., ranking original documents as more coherent than documents with only one sentence randomly misplaced. These tasks

¹The Switchboard Coherence corpus is available for download at: <https://github.com/alecervi/switchboard-coherence-corpus>

²The code for the models presented in this work can be found at: <https://github.com/alecervi/turn-coherence-rating>

are still considered standard to this day and found wide applications, especially for text (Farag et al., 2018; Clark et al., 2018). Recent models proposed for these tasks are based on Convolutional Neural Networks (Nguyen and Joty, 2017), also applied to thread reconstruction (Joty et al., 2018), while the current State-of-the-art is based on a combination of bidirectional Long Short-Term Memory encoders and convolution-pooling layers (Moon et al., 2019). These tasks, however, consider documents as-a-whole and rely mainly on entities information.

Coherence evaluation in dialogue Models for dialogue coherence evaluation have mainly been explored using supervised approaches, i.e., training on corpora with human annotations for coherence, mostly at the turn level (Higashinaka et al., 2014; Gandhe and Traum, 2016; Venkatesh et al., 2017; Lowe et al., 2016; Yi et al., 2019). Different approaches tried to apply the standard coherence tasks to conversational domains such as dialogue and threads, but mainly considering the evaluation of dialogues as-a-whole (Purandare and Litman, 2008; Elsner and Charniak, 2011; Cervone et al., 2018; Vakulenko et al., 2018; Joty et al., 2018; Mesgar et al., 2019; Zhou et al., 2019). In particular, Cervone et al. (2018) found that discrimination might be over-simplistic for dialogue coherence evaluation when considering Dialogue Act (DA) information. In this work, we propose a novel framework to model entities and DAs information for turn coherence prediction using a weakly supervised training methodology. Furthermore, our focus is on predicting coherence of single turns rather than entire dialogues.

Response Selection As a task, response selection has become a standard (Lowe et al., 2017; Yoshino et al., 2019; Kumar et al., 2019) for training both task-oriented and non-task-oriented retrieval-based dialogue models. The task proved to be useful for evaluating models in task-oriented (Ubuntu), social media threads (Twitter Corpus), and movie dialogues (SubTle Corpus) (Lowe et al., 2016). Recently the task has also been proposed for pre-training models for task-oriented dialogue (Henderson et al., 2019) and for Dialogue Act tagging (Mehri et al., 2019). In this work, we investigate response selection as a task for training coherence rating models for spoken dialogue. Additionally, while response selection models are usually based on the entire text as input (Lowe et al., 2017), we rely solely on entities and DAs information, in or-

der to investigate their effect on turn coherence perception.

3 Methodology

In this work, we are interested in the relation between Dialogue Acts (DAs) and entities and how they can be modelled to train automatic predictors of next turn coherence in non-task-based dialogue.

Our hypothesis is that both entities and DAs are useful to predict the coherence of the next turn. In order to verify such hypothesis, we first perform an analysis of entities and DAs patterns of distribution in the Switchboard Coherence (SWBD-Coh) corpus, a novel dataset of human-human telephone conversations from Switchboard annotated with human coherence ratings per turn.

Secondly, we hypothesize that we can model entities and DAs to predict next turn coherence ratings. Rather than using supervised data for coherence prediction, we use a weakly supervised training methodology, i.e. training on the task of response selection (which proved useful for other dialogue tasks (Henderson et al., 2019)) and testing on coherence ratings. In response selection given a *context*, i.e. the history of the dialogue up to the current turn, and a *list of next turn candidates*, models are asked to rank candidates according to their appropriateness with the previous dialogue history. The positive training samples for this task are automatically generated by randomly selecting a given turn in a dialogue, and considering this turn as a positive (coherent) example with the current history of the conversation (the context). Negative samples are generated by selecting other random dialogue turns, assuming that they will mostly be not appropriate as the next turn in the dialogue. In particular, we investigate two methodologies to generate negative samples from the training data automatically:

Internal swap: a random turn is selected from a subsequent part of the same conversation. We assume this task to be harder for coherence evaluation since typically conversations do not have radical topic shifts.

External swap: a random turn is selected from other conversations. We assume this task to be easier given the probable shifts in topic.

In our first set of experiments, we thus train our models on response selection. One of the possible shortcomings of the data generation procedure used in response selection, however, is the amount of

	Train	Dev	Test
No. source dialogues	740	184	231
No. insertion points	7400	1840	2310
No. pos/neg pairs	66600	16560	20790

Table 1: Train, development and test data size for response selection for both Internal and External Swap.

false negatives. Although it is assumed that the majority of negative samples generated with this methodology will not be appropriate for the context, there could still be cases in which they are.

In order to verify the performance of our models based on DAs and entities to predict real human coherence judgments, in our second set of experiments models are tested on SWBD-Coh. Analogously to response selection, in turn coherence rating models need to rank next turn candidates given the history of the dialogue. In this case, however, the ranking is not binary but is rather based on a graded coherence rating given by humans for next turn candidates (for further details on the SWBD-Coh corpus see Section 4).

4 Data

The dataset chosen for our experiments is the Switchboard Dialogue Act corpus (Stolcke et al., 2000) (SWBD-DA), a subset of Switchboard annotated with DA information. The Switchboard corpus is a collection of human–human dyadic telephone conversations where speakers were asked to discuss a given topic. This dataset was chosen both to ensure comparability with previous work on dialogue coherence and because it is open-domain. Also, this corpus has DA annotations. Interestingly, SWBD-DA is a real-world (transcribed) spoken corpus, so we have sudden topic changes, overlap speech, disfluencies and other typical characteristics of spoken interaction. Since our goal was to study coherence in a real-world spoken dialogue setting, rather than removing these features as errors, we considered them an integral part of spoken conversations and did not remove them.

Response Selection Source dialogues are split into train, validation, and test sets (see Table 1) using the same distribution as Cervone et al. (2018). For each dialogue, we randomly choose ten insertion points. Each insertion point is composed by a context (dialogue history up to that point) and the original turn following that context (regarded as positive). In order to have 10 next turn candi-

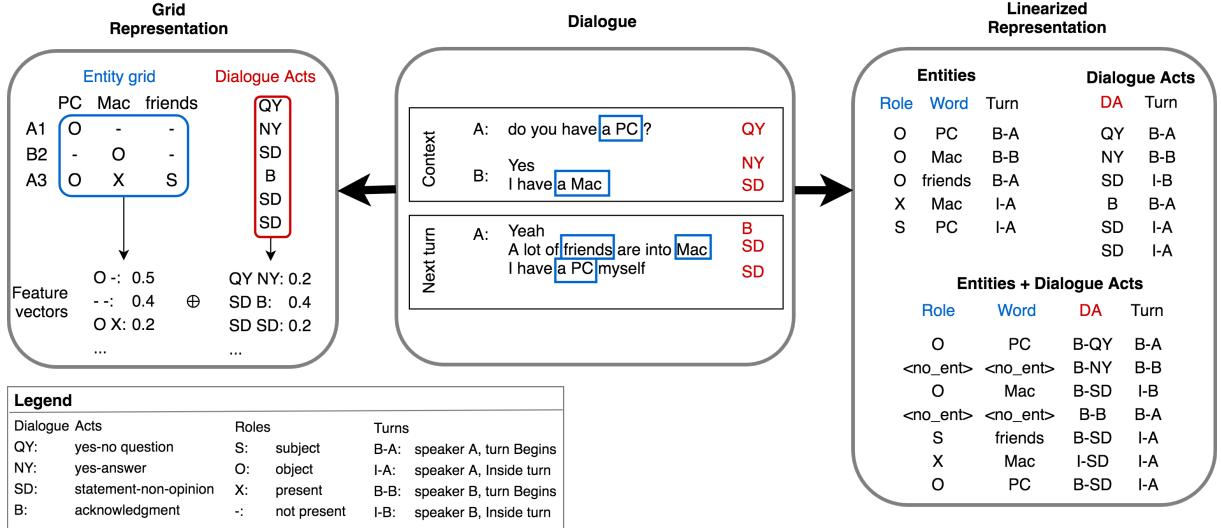


Figure 1: A source **dialogue** (at the center of the figure) is transformed into a **grid representation** (left) and into a **linearized representation** (right). In the grid representation, entities and Dialogue Acts (DAs) are transformed into feature vectors and can then be concatenated. Our linearized representation, i.e. the input to our neural models, shows 3 different possibilities: one where we only consider entity features at the turn level (top-left), another one which considers only DA features (top-right), and a joined one where DAs and entities are combined (bottom).

dates, for each insertion point 9 adversarial turns (regarded as negatives) are then randomly selected either from subsequent parts of the dialogue, i.e. Internal Swap (IS), or from other dialogues, i.e. External Swap (ES), within the same data subset, so that for example external adversarial turns for training are only taken from other source dialogues in the training set.

Switchboard Coherence corpus The dataset for turn coherence rating, the Switchboard Coherence corpus (SWBD-Coh), was created using as source dialogues the ones from SWBD-DA which are in the testset of Cervone et al. (2018). The data were annotated using Amazon Mechanical Turk (AMT). 1000 insertion points were randomly selected, following the constraints that the context (dialogue history up to the original turn) could be between 1 and 10 turns length. Since in this task we want to evaluate the coherence of a given turn with the previous dialogue history, 1 turn of context was the minimum required. We set the maximum length to 10 turns to reduce annotation time. For each insertion point, six adversarial turns were randomly selected, besides the original one (3 using the IS methodology, 3 using the ES one) for a total of 7 turn candidates. Overall the SWBD-Coh dataset is thus composed of 7000 pairs (1000 contexts \times 7 turns).

Each context and turns pair was annotated by 5 AMT workers with coherence ratings. More specif-

ically, for each dialogue workers were presented with the dialogue history up to the insertion point and the next turn candidates (randomly shuffled). Workers were asked to rate on a scale of 1 (not coherent), 2 (not sure it fits) to 3 (coherent) how much each response makes sense as the next natural turn in the dialogue. All workers (37) who annotated the dataset were first evaluated on a common subset of 5 dialogues where they had an average Weighted Kappa agreement with quadratic weights with two gold (internal) annotators of $\kappa = 0.659$ (min: 0.425, max: 0.809, STD: 0.101) and among each other an average leave-one-out correlation of $\rho = 0.78$ (i.e. correlating the scores of each worker with mean scores of all other workers who annotated the same data), following the approach used in other coherence rating datasets (Barzilay and Lapata, 2008; Lapata, 2006). ³ Scores for each candidate turn were then averaged across all annotators. Original turns were regarded on average as more coherent ($\mu = 2.6$, $SD = 0.5$) than adversarial turns, while turns generated with IS were considered more coherent ($\mu = 1.8$, $SD = 0.7$) than the ones generated via ES ($\mu = 1.4$, $SD = 0.6$).

5 Data analysis

In this section, we analyse the Switchboard Coherence (SWBD-Coh) dataset in regards to the dis-

³More details about our data collection procedure are available in Appendix A.

tribution of Dialogue Acts (DAs) and entities. In particular, we are interested in analysing which features might affect human judgement of coherence of a given next turn candidate. For entities, we analyse two features: the number of entities mentioned in the next turn candidate that overlap with entities introduced in the context and the number of novel entities introduced in the turn. Additionally, we create a binary feature for each DA type that registers the presence of that DA in the turn candidate.

We use multiple regression analysis to verify how these different features correlate with human coherence ratings. Table 2, reports the Multiple Correlation Coefficient (MCC) of regression models using R squared and Adjusted R squared (Theil, 1961), adjusted for the bias from the number of predictors compared to the sample size. The results of our analysis indicate that the best MCC, 0.41 when calculated with the Adjusted R squared, is achieved when combining all features, both from entities and DAs. Moreover, in the lower part of Table 2 we report some of the features that proved to be the most relevant for predicting human coherence ratings. In general, it seems that while the entities overlapping the previous context seems to affect positively human coherence judgements, the DAs that most affect ratings do so in a negative way and seem to be mostly contentful DAs, such as *statement-opinion*, rather than DAs which typically present no entities, such as *acknowledge*. Our interpretation is that, in cases when there are no overlapping entities with the context, these DAs might signal explicit examples of incoherence by introducing unrelated entities.

6 Models

We model dialogue coherence by focusing on two features that have been closely associated to coherence in previous literature: the entities mentioned and the speakers’ intents, modelled as Dialogue Acts (DAs), in a conversation. Our models explore both the respective roles of entities and DAs and their combination to predict dialogue coherence. We investigate both standard coherence models based on Support Vector Machines (SVM) and propose novel neural ones.

6.1 SVM models

The entity grid model (Barzilay and Lapata, 2008) relies on the assumption that transitions from one

	MCCR²	MCCAR²
Entities	0.27	0.26
DAs	0.34	0.29
All (Entities + DAs)	0.45	0.41
<i>Relevant features in All</i>	<i>Coeff.</i>	<i>Sign.</i>
Overlapping entities	0.26	**
DA: decl. yes-no-question	-0.48	*
DA: statement-opinion	-0.31	**
DA: statement-non-opinion	-0.30	**
DA: acknowledge	0.27	**

Table 2: Multiple Correlation Coefficients (MCC) from R squared (R^2) and Adjusted R squared (AR^2) of different multiple regression models that predict human coherence ratings for candidate turns given a dialogue context (turn coherence rating task) on the Switchboard Coherence corpus. Additionally, we report coefficients and significance (where * denotes $.05 \geq p \geq .01$ and ** $p < .01$) of some relevant features for the best-performing model (All).

syntactic role to another of the same entities across different sentences of a text indicate local coherence patterns. This assumption is formalized by representing a text (in our case, a dialogue) as a grid, as shown in Figure 1. For each turn of the dialogue we extract the entities, i.e. Noun Phrases (NPs), and their respective grammatical roles, i.e. whether the entity in that turn is subject (S), direct object (O), neither (X), or it is not present ($-$). Each row of the grid represents a turn in the dialogue, while each column represents one entity (in Figure 1, for example, the first turn of speaker A is represented by the first row of the grid $O - -$). Using this representation, we can derive feature vectors to be used as input for Machine Learning models by extracting probabilities of all role transitions for each column.

More formally, the coherence score of a dialogue D in the entity grid approach can be modelled as a probability distribution over transition sequences for each entity e from one grammatical role r to another for all turns t up to a given history h (see Eq. 4 in Lapata and Barzilay (2005)):

$$p_{cohEnt}(D) \approx \frac{1}{m \cdot n} \prod_{e=1}^m \prod_{t=1}^n p(r_{t,e} | r_{(t-h),e} \dots r_{(t-1),e}) \quad (1)$$

The probabilities for each column (entity) are normalized by the column length n (number of turns in the dialogue) and the ones for the entire dialogue by the number of rows m (number of entities in the dialogue). In this way, we obtain

the feature vectors shown in Figure 1 where each possible roles transition of a predefined length (e.g. $O-$) is associated with a probability. These feature vectors are then given as input to a Support Vector Machine (SVM) in the original model.

Following Cervone et al. (2018), we can use the same approach to construct similar feature vectors for DAs information:

$$p_{cohDA}(D) \approx \frac{1}{n} \prod_{i=1}^n p(d_i | d_{(i-h)} \dots d_{(i-1)}) \quad (2)$$

Here the coherence score of a dialogue is given by the probability of the entire sequence of DAs (d) for the whole dialogue, normalized by column length (n), i.e. the number of DAs for each turn.

The joint model, the one combining entity and DA information, concatenates feature vectors obtained from both. While other ways of combining DA and entities have been explored in Cervone et al. (2018), the authors report that practically a concatenation resulted in the best performances across all tasks, probably due to data sparsity issues.

Indeed among the limitations of the entity grid, there is data sparsity: for example for an entity appearing only in the last turn of a dialogue we need to add a column to the grid which will be mostly containing “empty” — transitions (see *friends* in Figure 1). Another problem of this approach is the fact that the model is not lexicalized since we only keep role transitions when computing the feature vectors for the entities. Furthermore, the model makes the simplifying assumption that columns, thus entities, are independent from each other.

6.2 Neural models

Our neural coherence models for dialogue are based on bidirectional Gated Recurrent Units (biGRU). While other neural coherence models (Nguyen and Joty, 2017; Joty et al., 2018) rely directly on the grid representation from Barzilay and Lapata (2008), we explore a novel way to encode the dialogue structure. The input to our biGRUs is a sequential representation of the dialogue.

6.2.1 Sequential input representation

We linearize the structure of a dialogue composed by entities, DAs and turns into flat representations for our neural models, as in Figure 1. These representations can then be mapped to an embedding layer and joined via concatenation. We consider three cases: (i) the case in which we model entity features; (ii) the one in which we consider DAs

information; (iii) the one in which we combine both.

Entities encodings In our approach, entities are Noun Phrases, as in the entity grid approach. For each dialogue, we consider the sequence of entities ordered according to their appearance in the conversation (see Figure 1). Entities are represented either by their grammatical roles ent_{role} in the dialogue (using the same role vocabulary V_r of the original grid), their corresponding words ent_{word} (from a vocabulary V_w), or by both. Another feature which can be added to this representation is the *turn* (whether A or B is talking). This feature could be useful to encode the dyadic structure of the dialogue and how this might be related to entity mentions. In order to better encode the boundaries of speaker turns, turns are mapped to the IOB2 format (where the Outside token is removed because naturally never used for turns), for a resulting turn vocabulary V_t size of 4 tags (2 speakers x 2 IOB tags used). Special tokens ($<\text{no_ent}>$) are added to both V_w and V_r for cases in which turns do not present any entities.

DAs encodings In case we consider only DAs features, our input representation becomes a sequence of DAs for the whole dialogue history so far, drawn from a vocabulary V_d . Also, in this case, *turn* features can be added to mark the turn-wise structure of the DA sequence, using the same vocabulary V_t previously described.

Entities + DAs encodings We combine entities and DAs by considering the sequence of entities in order of their appearance within each DA and encoding DAs into IOB2 format, as previously done for turn features. In this setting, thus, the vocabulary V_d has double the size, compared to the setting where we consider only DAs. Analogously to previous settings, turn features can be added to encode turn boundaries.

It can be noticed how our representation is less sparse compared to both the original grid (Barzilay and Lapata, 2008) and recently proposed models (Nguyen and Joty, 2017), which take as input grid columns directly. Furthermore, compared to the original grid, our representation is lexicalized.

6.2.2 Architecture

The architecture of our models is shown in Figure 2. In the first layer of the network each input feature (ent_{role} , ent_{word} , DA , $turn$) is mapped to

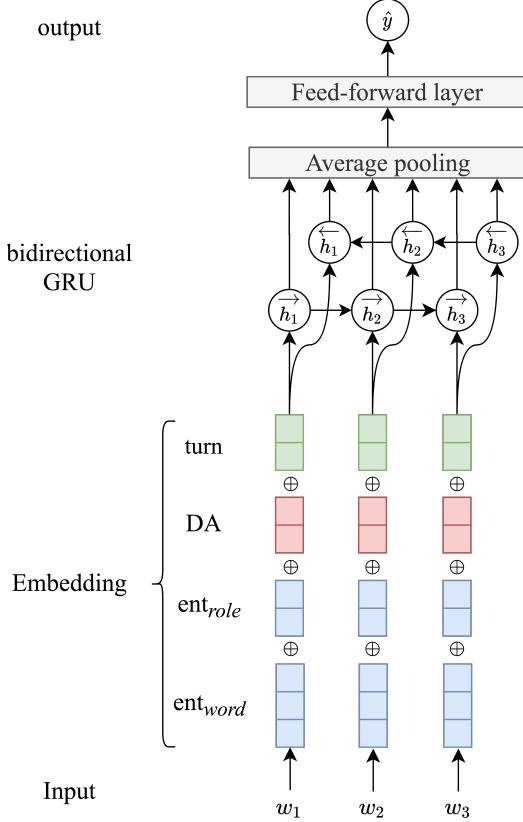


Figure 2: Our proposed architecture based on bidirectional GRUs with input entity word embedding (ent_{word}) and grammatical role (ent_{role}), Dialogue Act (DA) and speaker *turn* features.

a d -dimensional dense vector by looking up into their respective embedding matrix \mathbf{E} , one per feature type. All features vectors obtained can then be combined using concatenation. This vector is then recursively passed to the bidirectional GRU layers and then to a mean pooling layer. Finally, the output is passed through a feed-forward neural network with one hidden layer and ReLU as non-linearity.

Our models are trained using a Margin-ranking loss with a margin of 0.5 using the following equation:

$$\text{loss}(x, y) = \max(0, -y*(x_1 - x_2) + \text{margin}) \quad (3)$$

where x_1 and x_2 are respectively the original dialogue and the adversarial one and $y = 1$. In this way, the model is asked to rank the original dialogue higher (more coherent) than the adversarial one. The model is trained by Stochastic Gradient Descent, using the Adam update rule (Kingma and Ba, 2015).

7 Experimental set-up

Preprocessing Entities, i.e. Noun Phrases (NPs), and their syntactic roles were extracted and preprocessed with Cervone et al. (2018)'s pipeline ⁴. Following the original entity grid formulation (Barzilay and Lapata, 2008), only NPs heads were kept. The DAs are taken from annotations on SWBD-DA (using the standard reduction to 42 tags compared to the DAMSL ones).

Evaluation For evaluating response selection, we use pairwise Accuracy, the metric used in standard coherence tasks, which evaluates the ability of the model to rank original turns higher than each adversarial one. However, this metric is not indicative of the global ranking of all candidate turns for a given context. For this reason, we add two ranking metrics to evaluate our models: Mean Reciprocal Rank (MRR), which evaluates the average of reciprocal ranks of all candidate turns for a context, and Recall at One (R@1) and Two (R@2), also used in previous work on response selection (Lowe et al., 2017; Zhou et al., 2018) to assess the ability of the model to rank original turns respectively within the first or second rank among all candidates.

Compared to response selection, where we have a binary choice between coherent and negative turns, in turn coherence rating, we have a set of candidate turns each associated to a coherence score. In this case, we use Accuracy, MRR, R@1 and Normalized Discounted Cumulative Gain (nDCG) to evaluate our models. Accuracy was computed only for cases in which the rating of the turn was not identical across two candidate turns. MRR and R@1 were computed dynamically, that is considering the turn with the highest score within that particular context as the best one in the rank. The nDCG metric (Järvelin and Kekäläinen, 2002) assesses the gain of a candidate according to its rank among all candidates. Compared to previous metrics, nDCG allows taking into account the relevance (in our case, the coherence score) of candidates. For all metrics considered, if our models predicts the same score for two candidates, we always assume models made a mistake, i.e. among candidates with the same predicted score positive examples are ranked after the negative ones.

Models' settings Grid models, based on SVMs, were trained with default parameters using

⁴<https://github.com/alecervi/Coherence-models-for-dialogue>

	Internal Swap				External Swap			
	Acc.	MRR	R@1	R@2	Acc.	MRR	R@1	R@2
Random	50.0	0.293	0.099	0.198	50.0	0.293	0.099	0.198
SVM ent _{role} (Entity Grid)	36.6	0.260	0.103	0.178	39.5	0.246	0.096	0.126
SVM DA (Cervone et al., 2018)	60.6	0.398	0.206	0.335	61.3	0.403	0.212	0.346
SVM ent _{role} + DA (Cervone et al., 2018)	62.7	0.417	0.222	0.365	64.3	0.437	0.251	0.380
biGRU ent _{role}	41.8	0.294	0.120	0.217	45.5	0.293	0.117	0.210
biGRU ent _{role} + turn	43.3	0.295	0.120	0.214	45.9	0.293	0.115	0.211
biGRU ent _{word}	47.8	0.324	0.151	0.252	56.4	0.397	0.236	0.337
biGRU ent _{word} + turn	49.0	0.331	0.162	0.255	56.9	0.400	0.241	0.341
biGRU ent _{role} + ent _{word} + turn	48.6	0.327	0.156	0.253	56.1	0.394	0.232	0.338
biGRU DA	72.4	0.484	0.276	0.443	72.6	0.486	0.278	0.447
biGRU DA + turn	74.0	0.501	0.297	0.464	74.1	0.508	0.305	0.475
biGRU ent _{word} + DA + turn	75.1	0.520	0.321	0.484	77.3	0.550	0.355	0.530
biGRU all	75.0	0.521	0.321	0.489	77.2	0.549	0.354	0.529

Table 3: Average (5 runs) of Accuracy (Acc.), Mean Reciprocal Rank (MRR) and Recall at one (R@1) and two (R@2) for response selection using both data generation methodologies (Internal and External Swap) on Switchboard.

SVM^{light} preference kernel (Joachims, 2002)) as in the original model (Barzilay and Lapata, 2008). For saliency, i.e. the possibility of filtering entities according to their frequency, and transitions length we follow the default original grid parameters (saliency:1, transitions length:2). For neural models, implemented in Pytorch (Paszke et al., 2019), parameters were kept the same across all models to ensure comparability. The learning rate was set to 0.0005, batch size to 32, with two hidden biGRU layers of size 512. Embedding sizes for all features were set to 50–dimensions, except for word embeddings which had dimension 300. Models run for a maximum of 30 epochs with early stopping, based on the best MRR score on the development set.

8 Results

In this section, we report the results of our models for response selection. The best performing models on response selection are then evaluated on the turn coherence rating task using the Switchboard Coherence (SWBD-Coh) corpus as testset. For both tasks we compare our models to a random baseline. All reported results for neural models are averaged across 5 runs with different seeds.

Response selection The results for response selection are reported in Table 3. Neural models seem to capture better turn-level coherence compared to classic grid SVM-based approaches. In both data generation methodologies, Internal (IS) and Exter-

	Train	Acc.	MRR	R@1	nDCG
Random		50.0	0.479	0.234	0.645
biGRU	IS	42.7	0.395	0.174	0.621
ent _{word} + turn	ES	50.4	0.444	0.229	0.679
biGRU	IS	56.0	0.553	0.326	0.717
DA + turn	ES	56.0	0.558	0.337	0.725
biGRU	IS	58.5	0.575	0.358	0.738
ent _{word} + DA + turn	ES	61.1	0.583	0.369	0.760

Table 4: Average (5 runs) of Accuracy (Acc.), Mean Reciprocal Rank (MRR), Recall at one (R@1) and Normalized Discounted Cumulative Gain (nDCG) for turn coherence rating for models trained using either Internal (IS) or External Swap (ES) on the Switchboard Coherence corpus.

nal Swap (ES), SVM coherence models are outperformed by neural ones for all metrics considered. As expected, entity features (ent_{role}, ent_{word}) play a more prominent role in ES compared to IS. In both cases, entity features seem to be better captured by neural models relying on our proposed input representation. When considering lexical information (ent_{word}), however, ent_{role} features seem less relevant. This might be due to the fact that spoken dialogue has usually less complex syntactic structures compared to written text. Furthermore, parsers are usually trained on written text, and thus might be more error-prone when applied to dialogue where there are disfluencies, sudden changes of topics, etc. We notice that DAs alone (without entity information) play an important role in both IS and ES. Turn features capturing speaker infor-

mation seem helpful for both DAs and entities. In general, the combination of DAs and entities gives the best results both in SVM and neural models for both tasks, with the best performing one being the model combining ent_{word} , DA and turn features and without ent_{role} . Additionally, if we compare the IS setting to ES in terms of best MRR, Accuracy and Recall, the former seems more difficult. This confirms our expectations that IS might be an harder task for coherence.

Turn coherence rating A selection of best performing models for entities, DAs and their combination were tested on the SWBD-Coh dataset. Table 4 shows models’ results under both training conditions, i.e. either using IS or ES data. The lowest performing model seems to be the one based solely on entity features ($\text{ent}_{\text{word}} + \text{turn}$), while models combining DA with entities information ($\text{ent}_{\text{word}} + \text{DA} + \text{turn}$) are the best performing ones. Additionally, models trained on ES data perform better than those trained on IS across all conditions.

9 Conclusions

In this work, we investigate how entities and Dialogue Acts (DAs) are related to human perception of turn coherence in dialogue. In order to do so, we create a novel dataset, the Switchboard Coherence (SWBD-Coh) corpus, of transcribed open-domain spoken dialogues annotated with turn coherence ratings. A statistical analysis of the corpus confirms how both entities and DAs affect human judgements of turn coherence in dialogue, especially when combined. Motivated by these findings, we experiment with different models relying on entities and DAs to automatically predict turn coherence, i.e. standard coherence models and novel neural ones. In particular, we propose a less sparse alternative, compared to the entity grid, to encode entities and DAs information. Rather than using data annotated explicitly for the task, i.e. coherence prediction, we explore two response selection methodologies for training. We find that our newly proposed architecture outperforms standard ones in response selection. Finally, we test our models on the SWBD-Coh corpus in order to evaluate their ability to predict real human turn coherence ratings. Crucially, we find that the combination of DAs and entities gives the best performances.

For the future work, it would be interesting to investigate how to apply large pretrained models to our task, such as BERT (Devlin et al., 2019). While

pretrained models have recently been successfully explored for text-based response selection (Kim et al., 2019; Henderson et al., 2019), integrating them with our proposed input representation is not a straightforward task since such models typically rely on the whole textual context, while our models do not.

While there is still much to understand regarding turn coherence in dialogue, we believe our work could be a first step towards uncovering the relation between DAs and entities in open-domain spoken dialogue. Moreover, we believe that the SWBD-Coh corpus could become a useful resource for the community to study coherence in open-domain spoken dialogue.

Acknowledgments

The research leading to these results has received funding from the European Union H2020 Programme under grant agreement 826266: COAD-APT.

References

- Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.
- Alessandra Cervone, Evgeny Stepanov, and Giuseppe Riccardi. 2018. *Coherence models for dialogue*. *Proc. Interspeech 2018*, pages 1011–1015.
- Elizabeth Clark, Yangfeng Ji, and Noah A Smith. 2018. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2250–2260.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Micha Elsner and Eugene Charniak. 2011. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1179–1189.
- Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. Neural automated essay scoring and coherence modeling for adversarially crafted input. In *Proceedings of the 2018 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), volume 1, pages 263–271.

Sudeep Gandhe and David Traum. 2016. A semi-automated evaluation metric for dialogue model coherence. *Situated Dialog in Speech-Based Human-Computer Interaction*, page 217.

Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225.

Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. 2019. *Training neural response selection for task-oriented dialogue systems*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5392–5404, Florence, Italy. Association for Computational Linguistics.

Ryuichiro Higashinaka, Toyomi Meguro, Kenji Ima-mura, Hiroaki Sugiyama, Toshiro Makino, and Yoshihiro Matsuo. 2014. Evaluating coherence in open domain conversational systems. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Jerry R Hobbs. 1979. Coherence and coreference. *Cognitive science*, 3(1):67–90.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.

Aravind K Joshi and Steve Kuhn. 1979. Centered logic: The role of entity centered sentence representation in natural language inferencing. In *IJCAI*, pages 435–439.

Shafiq Joty, Muhammad Tasnim Mohiuddin, and Dat Tien Nguyen. 2018. Coherence modeling of asynchronous conversations: A neural entity grid approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 558–568.

Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, et al. 2019. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*.

Diederik P. Kingma and Jimmy Ba. 2015. *Adam: A method for stochastic optimization*. In *3rd International Conference on Learning Representations*,

ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Harshit Kumar, Arvind Agarwal, and Sachindra Joshi. 2019. A practical dialogue-act-driven conversation model for multi-turn response selection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1980–1989.

Mirella Lapata. 2006. Automatic evaluation of information ordering: Kendall’s tau. *Computational Linguistics*, 32(4):471–484.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI*, volume 5, pages 1085–1090.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL-HLT*, pages 110–119.

Ryan Lowe, Iulian V Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. On the evaluation of dialogue systems with next utterance classification. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 264.

Ryan Thomas Lowe, Nissan Pow, Iulian Vlad Serban, Laurent Charlin, Chia-Wei Liu, and Joelle Pineau. 2017. Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue & Discourse*, 8(1):31–65.

Shikib Mehri, Evgeniia Razumovskaya, Tiancheng Zhao, and Maxine Eskenazi. 2019. Pretraining methods for dialog context representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3836–3845.

Mohsen Mesgar, Sebastian Bücker, and Iryna Gurevych. 2019. A neural model for dialogue coherence assessment. *arXiv preprint arXiv:1908.08486*.

Han Cheol Moon, Muhammad Tasnim Mohiuddin, Shafiq Joty, and Chi Xu. 2019. A unified neural coherence model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2262–2272.

Dat Tien Nguyen and Shafiq Joty. 2017. A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1320–1330.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward

- Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- Amruta Purandare and Diane J Litman. 2008. Analyzing dialog coherence using transition patterns in lexical and semantic features. In *FLAIRS Conference*, pages 195–200.
- Gisela Redeker. 2000. Coherence and structure in text and discourse. *Abduction, belief and context in dialogue*, pages 233–263.
- Harvey Sacks and Gail Jefferson. 1995. Lectures on conversation. *Human Studies*, 18(2):327–336.
- Emanuel A Schegloff. 1968. Sequencing in conversational openings. *American anthropologist*, 70(6):1075–1095.
- Iulian Vlad Serban, Ryan Lowe, Laurent Charlin, and Joelle Pineau. 2016. Generative deep neural networks for dialogue: A short review. *arXiv preprint arXiv:1611.06216*.
- Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.
- Henri Theil. 1961. Economic forecasts and policy.
- David R Traum. 1994. A computational theory of grounding in natural language conversation. Technical report, Rochester Univ NY Dept of Computer Science.
- Svitlana Vakulenko, Maarten de Rijke, Michael Cochez, Vadim Savenkov, and Axel Polleres. 2018. Measuring semantic coherence of a conversation. In *International Semantic Web Conference*, pages 634–651. Springer.
- Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Benham Hedayatnia, Angeliki Metallinou, Rahul Goel, Shaohua Yang, and Anirudh Raju. 2017. On evaluating and comparing conversational agents. In *NIPS 2017 Conversational AI workshop*.
- Anthony J Viera, Joanne M Garrett, et al. 2005. Understanding interobserver agreement: the kappa statistic. *Fam med*, 37(5):360–363.
- Sanghyun Yi, Rahul Goel, Chandra Khatri, Alessandra Cervone, Tagyoung Chung, Behnam Hedayatnia, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2019. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 65–75, Tokyo, Japan. Association for Computational Linguistics.
- Koichiro Yoshino, Chiori Hori, Julien Perez, Luis Fernando D’Haro, Lazaros Polymenakos, Chulaka Gunasekara, Walter S Lasecki, Jonathan K Kummerfeld, Michel Galley, Chris Brockett, Jianfeng Gao, Bill Dolan, Xiang Gao, Huda Alamari, Tim K. Marks, Devi Parikh, and Dhruv Batra. 2019. Dialog system technology challenge 7. *NeurIPS Workshop: The 2nd Conversational AI: Todays Practice and Tomorrows Potential*.
- Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1127.
- Yunxiao Zhou, Man Lan, and Wenting Wang. 2019. Hierarchical intention enhanced network for automatic dialogue coherence assessment. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

A Appendix A: Switchboard Coherence corpus data collection procedure

Coherence rating is an inherently subjective task and could be challenging especially for a dataset of transcribed real-world open-domain human-human conversation like Switchboard, where we have possible interruptions, overlaps and disfluencies naturally occurring. Hence, in order to ensure we collected reliable judgements for turn coherence, we followed a multi-step procedure to build the Switchboard Coherence (SWBD-Coh) corpus using Amazon Mechanical Turk (AMT).

A.1 Experiment with internal annotators

First we performed a small-scale annotation experiment to evaluate the feasibility of the task. Two internal annotators, both with Linguistics education, were asked to rate a set of 150 different dialogues randomly selected from the testset from (Cervone et al., 2018). The 150 annotation pairs (context + set of candidate turns) were generated using the same procedure described in Section 4 of the paper. The coherence scale was divided into 1 (not coherent), 2 (not sure it fits) and 3 (coherent). Since we wanted to capture a general perception of coherence, rather than bias annotators towards our own intuitions, in the guidelines annotators the task was described as: “Your task is to rate each candidate on a scale of how much it is *coherent* with the previous dialogue context, that is *how much that response makes sense as the next natural turn in the dialogue*”.

Since in this case we only have two annotators, we were able to measure their inter-annotator agreement using a weighted kappa score with quadratic weights (since our categories are ordinal). The inter-annotator agreement was of 0.657 (which can be regarded as substantial (Viera et al., 2005)). Then, we averaged scores for each candidate turn from both annotators. As shown in Table 5, original turns had higher coherence scores ($\mu = 2.66$) compared to adversarial turns, while turns generated with Internal Swap were considered more coherent ($\mu = 1.78$) than the ones generated via External Swap ($\mu = 1.45$).

A.2 Experiment with AMT

After having assessed the feasibility of the task, we then proceeded to set up the data collection procedure on AMT.

	Orig	IS	ES
μ score 150	2.7 (0.5)	1.8 (0.7)	1.4 (0.7)
μ score SWBD-Coh	2.6 (0.5)	1.8 (0.7)	1.4 (0.6)

Table 5: Comparison of human annotation results for the experiment with two internal annotators (150 dialogues) and the Switchboard Coherence (SWBD-Coh) dataset. Mean scores (and standard deviation) are reported for each candidates group: originals (Orig), internal swap (IS) and external swap (ES).

In order to select workers for our coherence annotation task we first set up a qualification task on AMT. The qualification task consisted of 5 dialogues (taken from the 150 internally annotated) with 7 turn candidates using the same coherence rating scale as in the gold annotation. In order to pass the qualification task a worker had to have a weighted kappa score higher than 0.4 with both our gold annotators. This threshold was decided empirically by first running a small scale experiment with other 4 internal annotators on the qualification task. 37 workers passed the qualification task. The average weighted kappa agreement with the two gold annotators was 0.659 (min: 0.425, max: 0.809, STD: 0.101). In order to calculate the agreement among all the 37 workers on this batch we employ leave-one-out resampling. For each worker who annotated the data we calculate the correlation of her/his scores with the mean ones of all other annotators in the batch. This is repeated for all workers and then averaged. This technique has been used in other coherence annotation experiments (Barzilay and Lapata, 2008; Lapata and Barzilay, 2005).

Workers who passed the qualification test could then proceed to annotate the SWBD-Coh data. The data, consisting of 1000 dialogues, was divided into 100 batches of 10 dialogues each. Each batch was annotated by at least 5 workers. In order to remove possible workers who did not perform well on a given batch, we employed a combination of techniques including leave-one-out resampling and average scores given to original turns. The average leave-one-out correlation per batch for turn coherence rating achieved with this data collection procedure was: $\rho = 0.723$ (min: 0.580, max: 0.835, STD: 0.055). Interestingly, as shown in Table 5, the average scores per candidate group (original, Internal swap, External swap) match closely the ones obtained in our gold 150 annotation data.

Context	Score	Candidates	Models ranks		
			Ent	DA	Ent+DA
A: Okay. B: Well, if you are from Iowa , you must be very artsy crafty. Everyone I've ever known from the Midwest can do everything with their hands . A: Oh, well, actually I'm from California and before then I was from Utah . So.	3.0	I didn't know anyone ever moved from California to Iowa ?	1	4	1
	2.6	Anyway, we are supposed to be talking about crafts . Do you, um, do you have any hobbies that, that you do things with your hands	2	2	2
	2.2	Right.	4	3	3
	2.2	Uh-huh.	4	3	3
	2.0	Oh, sure.	4	3	3
	1.2	bags some, their most recent, uh, needle craft	3	4	4
	1.0	at least at the end.	5	1	5

Table 6: Example of how different models relying only on entities (biGRU $\text{ent}_{\text{word}} + \text{turn}$), only on DAs (biGRU $\text{DA} + \text{turn}$) or both (biGRU $\text{ent}_{\text{word}} + \text{DA} + \text{turn}$) rank the same group of candidates for a given context.

B Appendix B Models output example

Table 6 shows an example of the ranking given by different models to the same context-candidates pairs in the SWBD-Coh corpus, compared to the average coherence score given by annotators. In particular, we report the ranking given by a model based solely on entities information (biGRU $\text{ent}_{\text{word}} + \text{turn}$), another one considering only DAs (biGRU $\text{DA} + \text{turn}$) and a third one considering both types of information (biGRU $\text{ent}_{\text{word}} + \text{DA} + \text{turn}$). All models were trained on response selection using the External Swap methodology. The models output is reported in terms of position in the rank. Entities appearing in the text are highlighted in bold.

In this example we notice entities overlap information with the previous context proves rather important in order to rank candidates according to coherence. For example, to rank the candidate with the highest coherence as the first one (*I didn't know anyone ever moved from California to Iowa?*) information regarding the overlapping entities *California* and *Iowa* allows the models encoding entities information to assign the correct rank, while the model relying only on DAs gives the candidate the fourth position in the rank. We also notice how both annotators and all models assign very close or the same middle rank scores to three very similar candidates (*Right*, *Uh-huh* and *Oh, sure.*), which indeed all have the same DA (“acknowledgment”).

Analyzing Speaker Strategy in Referential Communication

Brian McMahan

Rutgers University

brian.c.mcmahan@gmail.com

Matthew Stone

Rutgers University

mdstone@rutgers.edu

Abstract

We analyze a corpus of referential communication through the lens of quantitative models of speaker reasoning. Different models place different emphases on linguistic reasoning and collaborative reasoning. This leads models to make different assessments of the risks and rewards of using specific utterances in specific contexts. By fitting a latent variable model to the corpus, we can exhibit utterances that give systematic evidence of the diverse kinds of reasoning speakers employ, and build integrated models that recognize not only speaker reference but also speaker reasoning.

1 Introduction

Language users are able to work together to identify objects in the world (Clark and Wilkes-Gibbs, 1986, among others). This ability involves formulating creative utterances, assessing their meaning in context, and anticipating listeners’ understanding and response (Dale and Reiter, 1995; Clark and Schaefer, 1989, among others). Despite long study, fundamental questions remain unanswered about how people manage this complex problem solving. This paper explores one question in particular: how speakers establish that references are likely to be successful. In general, such expectations can be underwritten either linguistically, by reasoning about the meanings and denotations of candidate linguistic expressions, or cooperatively, by reasoning about and anticipating their interlocutors’ collaborative problem solving. Both kinds of reasoning are undoubtedly common, and both play a significant role in the psychological and computational literature on referential communication.

In this paper, we use quantitative cognitive models, fit to naturalistic corpora, to characterize the contributions of linguistic and cooperative reasoning in the spontaneous strategies of human interlocutors in referential communication. Our re-

search offers a number of contributions for the SIGDIAL community.

- In Section 2, we provide a catalogue of phenomena and examples to distinguish linguistic reasoning and cooperative reasoning in reference. This analysis shows that linguistic reasoning and cooperative reasoning attribute different risks and rewards to utterances, and so explains why formalizations of linguistic reasoning, such as traditional plan-based approaches to generating referring expressions, and formalizations of cooperative reasoning, as often realized in machine learning approaches, can lead to different predictions about utterance choice.
- In Section 3, we refine approaches from the literature to capture the key phenomena we associate with different aspects of linguistic and cooperative reasoning. This modeling effort allows us to explore different inferences on an equal footing, using learned meanings with open-ended vocabulary and probabilistic, vague denotations.
- In Section 4, we evaluate the predictions of the models on human utterances in dialogue. By fitting a latent variable model to the corpus, we find strong evidence that while speakers often offer safe, conservative references, a sizeable fraction take risks that are only explained either by linguistic reasoning or by cooperative reasoning; these risky choices are broadly successful.

Our findings give new detail to the received understanding of collaborative problem solving in dialogue. Interlocutors often improvise, using risky strategies, in problematic situations; in these cases, they may have to work together interactively to achieve mutual understanding.

We believe that the researchers working on computational discourse and dialogue are uniquely positioned to take up these results to build more powerful models of the reasoning of human speakers, and to use analogous models in the choices of automated systems. At the same time, we argue that appreciating the diversity of dialogue is necessary to build interactive systems that understand and respond appropriately to their human users. Our work culminates in a mixture model that, given a description, predicts not only what the likely referent is, but also what reasoning the speaker was likely to have used to produce it.

2 Linguistic and Cooperative Reasoning

Our work is motivated by a distinction between reasoning linguistically, about meanings and denotations, and reasoning cooperatively, about understanding and collaboration. We begin by reviewing the theoretical and practical literature behind the two different approaches. To be clear, many reference problems have simple, good solutions that any reasoning will find. Differences arise in more complicated cases, when speakers need to exploit the flexibility of linguistic meaning or the ability of the listener to recognize implicatures, and when speakers need to trade off between specific and general referring expressions.

For clarity, our discussion illustrates these effects with concrete examples, even though this requires us to anticipate some results from later in the paper. In particular, we draw on attested examples from the Colors in Context (CIC) dataset of Monroe et al. (2017), where a director must signal one target in a display of three color patches. An example is shown in Figure 1.

We characterize the examples in terms of the quantitative predictions of models (described in full detail in Section 3), which formalize linguistic reasoning and cooperative reasoning. These mod-

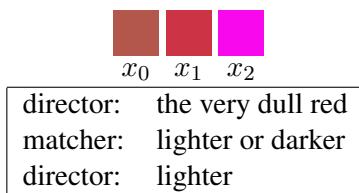


Figure 1: An example from the Colors in Context (CIC) dataset (Monroe et al., 2017) of the director and matcher coordinating so that the matcher can click on the correct color patch (x_0).

els adopt a decision-theoretic approach. Utterances achieve various outcomes with different probabilities. For example, we may be uncertain whether an utterance will be judged appropriate to the context, whether it will be understood correctly—and so whether it will be successful in advancing referential communication. Safer utterances, with a higher probability of success, contrast with riskier utterances, with lower probability of success.

In tandem, each utterance has a cost (fixed across models), which determines the utility obtained when the utterance is successful. A rare utterance, like *chartreuse*, is modeled as having a higher cost than a more frequent utterance with the same meaning, like *yellow-green*. In fact, general terms, like *blue-gray*, which describe a comparatively large subset of color space, are typically assigned a lower cost than more specific terms, like *slate*, which describe a narrower subset. This is because, in situations where general terms and specific terms both offer equal prospects of task success, human speakers tend to prefer the general ones. This preference is particularly strong for basic-level terms (Rosch, 1978; Berlin, 1991), like *blue*.

Overall then, the models assign each utterance an EXPECTED UTILITY, which combines risk and cost in a single preference ranking. As is common in empirical models of human choice (Luce, 1959; McFadden, 1973), speakers are modeled as stochastic, approximate utility maximizers. The greater the utility advantage of the best choice, the more likely speakers are to use it; less advantageous choices are unlikely but not impossible. This assumption translates the model of expected utility into a distribution over potential descriptions (w) conditioned on the target color patch (x_0) and context of all three color patches (C).

2.1 Linguistic Reasoning

For linguistic models of referential communication, reference is a matter of meaning. The referent of a definite referring expression must be the unique entity from the contextually salient set of candidates that satisfies the expression’s descriptive content. If there is no such unique entity, the referent is undefined.¹ Semantic reference is a proxy for successful communication. A speaker who establishes uniqueness can generally be confident that the listener will

¹In formal semantics and pragmatics, this requirement is typically modeled as a grammatically-encoded presupposition, with the contextually salient set derived via the general process of quantifier domain restriction (Roberts, 2003).

Linguistic Reasoning	
$P(\text{bright green } x_0, C)$	= 0.65
$P(\text{neon green } x_0, C)$	= 0.16
$P(\text{green } x_0, C)$	= 0.12
Cooperative Reasoning	
$P(\text{green } x_0, C)$	= 0.33
$P(\text{*bright green } x_0, C)$	= 0.18
$P(\text{lime green } x_0, C)$	= 0.11
x_0	
x_1	
x_2	
C	

Figure 2: Speakers can make their referring expressions more specific to come up with a description that’s true of the target and false of the distractors. The observed description is marked with *.

identify the same referent—without simulating the listener’s perspective or interpretive reasoning.²

Planning-based approaches to generating referring expressions in the tradition of Dale and Reiter (1995) implement linguistic reasoning: the fundamental task is to come up with a description that characterizes to the target object but excludes its distractors. Such uniquely identifying descriptions are successful; alternative descriptions that fail to characterize the target or fail to exclude distractors are not. See van Deemter (2016) for a recent survey. A consequence of this model is to favor more specific vocabulary when it is necessary to avoid ambiguity, as demonstrated in Figure 2 where the linguistic reasoning model heavily favors the attested description *bright green* that a human speaker uttered when presented with the context. Although individual items offer only anecdotal evidence, when human speakers reliably choose to use semantically-identifying descriptions (*bright green*) with higher costs than alternatives that cooperative reasoning predicts to be successful (*green*), we find systematic evidence that speakers do use linguistic reasoning to identify targets.

The vagueness of color terms complicates the story. The natural way to extend linguistic reasoning to vague descriptions is to follow Kennedy (2007) in defining vague predicates in terms of a contextually-determined threshold of applicability. Vague predicates apply to those items that meet the threshold and exclude those that do not. On this theory, vagueness arises because, in any real context, a range of thresholds (of indeterminate extent) will

²Of course, where the listener’s knowledge of language or the world is unexpectedly incomplete, linguistic reasoning may result in an expression that characterizes the referent uniquely but in a way the listener may not recognize (Clark and Marshall, 1981).

Linguistic Reasoning	
$P(\text{*yellow } x_0, C)$	= 0.69
$P(\text{mustard } x_0, C)$	= 0.06
$P(\text{greenish yellow } x_0, C)$	= 0.03
Cooperative Reasoning	
$P(\text{*yellow } x_0, C)$	= 0.30
$P(\text{yellow green } x_0, C)$	= 0.13
$P(\text{lime green } x_0, C)$	= 0.06
x_0	
x_1	
x_2	
C	

Figure 3: Linguistic flexibility. Speakers can tailor a denotation for vague predicates that distinguishes their target from its distractors. The observed description is marked with *.

typically be in play. There may be borderline cases that are neither clearly above all the thresholds in play nor clearly below them.

Speakers can exploit vagueness to communicate effectively (van Deemter, 2012). In particular, a speaker can implicitly choose to adopt further constraints on the threshold, leading to a more specific interpretation for the vague word. Once we take this possibility into account, a vague description refers uniquely as long as there are some (contextually-appropriate) thresholds where it identifies the target and none where it identifies a distractor (van Deemter, 2006; Meo et al., 2014). As an example, consider the attested utterance of *yellow* in Figure 3, where the target x_0 is a borderline case. Because x_0 is clearly a better yellow than the alternatives, there’s a natural specific interpretation for *yellow* (with threshold ranging from the yellowness of x_1 to that of x_0) that uniquely identifies the target. In contrast, if we do not track the specialized interpretations that arise from a semantic requirement of uniqueness, we predict that the term might still apply to the distractor objects, and create potential disambiguation problems even for a cooperative listener.

In using a vague description, the speaker may be uncertain about whether its interpretation as uniquely identifying is appropriate for the context. If this interpretation is too specific, meaning that the word draws a contrast between similar and salient items on either side of its threshold, the listener may judge it to be infelicitous (Graff Fara, 2000). This is a matter of degree; in evaluating descriptions that require relatively unusual or precise interpretations to uniquely identify the target (e.g., *blue* in Figure 4 below), linguistic reasoning predicts that they will be less likely to be contextually appropriate and so less likely to be used.

In short, when human speakers reliably exploit the flexibility of vague meanings to produce low-cost, linguistically-identifying descriptions, in ways that look comparatively risky on purely cooperative reasoning, as in Figure 3, we find evidence for linguistic reasoning.

2.2 Cooperative Reasoning

Cooperatively, meanwhile, listeners approach interpretation with preferences and expectations that efficient speakers can and should meet and exploit (Schelling, 1960; Clark, 1996, among others). A description doesn’t have to characterize the target uniquely—or even correctly—for the speaker to be confident that the listener will successfully retrieve the intended referent. Such cooperative effects are visible in the implicit strengthening of scalar implicatures (Horn, 1984; Frank and Goodman, 2012), where the listener naturally excludes a candidate interpretation that is technically possible but that the speaker could have been expected to signal differently. They are also visible in “loose talk” and exaggeration (Sperber and Wilson, 1986; Carston, 2002), where the description, while strictly speaking false, fits the target close enough to leave no doubt in the listener’s mind. These “inaccurate references” can even include cases of outright falsehood (Perrault and Cohen, 1981), if there’s a unique basis to link the false description with the intended target. Cooperative reasoning thus accommodates a diverse catalogue of non-unique descriptions that nevertheless succeed—what you might call, following Grice (1975), referential implicatures. A range of recent computational work has combined machine learning models of listener inference with probabilistic planning with the goal of generating such referential implicatures (Frank and Goodman, 2012; Monroe and Potts, 2015, among others).

Figure 4 shows an attested case, which we describe following Horn (1984). Interlocutors understand that *blue* can and will refer to the bright blue target in this context because it wouldn’t be rational to try to use *blue* to refer to either of the dull blue alternatives. Quantitatively, the linguistic judgment that the target but not the alternatives is in fact blue represents a very specific and unlikely interpretation of *blue*. By contrast, the cooperative speaker sees *blue* as a likely choice, because of the low cost of the expression being used, on the one hand, and the good likelihood of being (cooperatively and correctly) understood, on the other.

Linguistic Reasoning	
$P(\text{bright blue} x_0, C)$	= 0.34
$P(*\text{blue} x_0, C)$	= 0.24
$P(\text{royal blue} x_0, C)$	= 0.23
Cooperative Reasoning	
$P(*\text{blue} x_0, C)$	= 0.67
$P(\text{bright blue} x_0, C)$	= 0.07
$P(\text{royal blue} x_0, C)$	= 0.07

Figure 4: Referential implicature. A speaker who anticipates the listener’s cooperative reasoning can use a potentially ambiguous description if the intended target is the most salient fit. The observed description is marked with *.

Linguistic Reasoning	
$P(\text{orange} x_0, C)$	= 0.64
$P(*\text{red} x_0, C)$	= 0.13
$P(\text{red orange} x_0, C)$	= 0.08
Loose Talk	
$P(\text{orange} x_0, C)$	= 0.12
$P(*\text{red} x_0, C)$	= 0.12
$P(\text{peach} x_0, C)$	= 0.10

Figure 5: Loose talk. Even if this speaker judged the target x_0 to be orange, rather than red, she could be confident that her audience would resolve *red* to x_0 . The observed description is marked with *.

In such cases, when speakers reliably move forward with general expressions backed up by referential implicatures while linguistic reasoning favors more specific expressions, as in Figure 4, we find evidence for cooperative reasoning.

The description of Figure 4 is true of the target. What of inaccurate but comprehensible references? We show one possible attested case in Figure 5. Linguistic reasoning predicts the target should be described as *orange* rather than *red*. However, *red*, though a stretch, is unambiguous.

Unfortunately, we cannot be sure that the speaker intended the description *red* to be false but recognizable. An alternative explanation is that the speaker did categorize the patch as *red* (in a weird and idiosyncratic way). Our current data and methods cannot rule out such individual differences. In any case, our analysis suggests such examples are comparatively rare in this dataset, so our key models are designed to avoid loose talk.

In summary, prior linguistic research and prior computational models appeal to heterogeneous kinds of reasoning to explain how speakers plan

referential expressions. These models make incompatible predictions, particularly about how to handle vagueness and implicature, which are visible in their predicted trade-offs between specific and general referring expressions. How do these differences actually play out in natural dialogue? What evidence is there for linguistic reasoning and cooperative reasoning in the utterances of human speakers? And what effects might utterances with different origins have on the dynamics of interaction? The increasing availability of corpus data and the increasing power of machine learning methods makes it possible to adopt a quantitative approach to answering such questions. The remainder of this paper offers an initial experiment in this direction.

3 Learning Speaker Reasoning

We formulate computational models of speaker reasoning in two steps. First, as described in Section 3.1, we build the XKCD model based on the applicability and cost of color terms, following McMahan and Stone (2015); Monroe et al. (2016); McDowell and Goodman (2019). Second, we describe the linguistic and cooperative reasoning choices of speakers as a function of these learned parameters. As described in Section 3.2, our models of linguistic reasoning use probabilistic models of vagueness to formulate low-cost descriptions that denote the target uniquely (van Deemter, 2006; Meo et al., 2014). Meanwhile, as described in Section 3.3, our models of cooperative reasoning use probabilistic planning to find low-cost utterances likely to be understood by the listener, following the Rational Speech Acts approach (Frank and Goodman, 2012).

3.1 The XKCD Model

The linguistic and cooperative reasoning models depend on a shared model of meaning and cost which we name the XKCD model. We fit the XKCD model using a corpus of color patch descriptions that were freely labeled by volunteer crowd workers then cleaned in previous work (McMahan and Stone, 2015) resulting in 1.5M training, 108K development, and 544K testing examples.

Our assumption, in line to previous work (McMahan and Stone, 2015; McDowell and Goodman, 2019), is that speaker choices in this dataset can be attributed to two factors. The first is the APPLICABILITY of the description w_k to color patch x_i , denoted $\phi_{w_k}(x_i)$, which is a probabilistic mea-

sure of the degree to which a color description naturally fits a color patch. Applicability serves as a shared model of meaning for the models (which the models enrich pragmatically in different ways). The second factor is the AVAILABILITY, denoted α_{w_k} , which is a measure of the intrinsic frequency of a color description w_k . Availability inverts the intuitive notion of cost; descriptions with lower cost have higher availability (and higher utility).

We treat the XKCD model as a “literal speaker” in the specific sense that no referential implicatures factor in ϕ_{w_k} , since the speaker is not presented with alternative referential candidates and does not have the goal of identifying an intended target. As defined in Equation 1, the probability that the literal XKCD speaker uses the description w_k to describe patch x_i in context C is proportional both to w_k ’s applicability to x_i and to w_k ’s availability, and doesn’t depend on context.

$$S_0(w_k|x_i, C) = \frac{\phi_{w_k}(x_i)\alpha_{w_k}}{\sum_l \phi_{w_l}(x_i)\alpha_{w_l}} \quad (1)$$

In addition, we define a “literal listener” L_0 that leverages the applicability functions of the XKCD model in Equation 2:

$$L_0(x_i|w_k, C) = \frac{\phi_{w_k}(x_i)}{\sum_j \phi_{w_k}(x_j)} \quad (2)$$

L_0 quantifies the preference for interpretation x_i based on how appropriate the description w_k is for color patch x_i relative the other color patches.³

We implement the model as a neural network using the PyTorch deep learning framework (Paszke et al., 2019).⁴ Neural networks learn data-driven representations of color space and color categories, which leads to more flexible and accurate meanings (Monroe et al., 2016) compared to models that use handcrafted parameterizations for color space and color meanings as in McMahan and Stone (2015).

Starting from a Fourier feature representation of color patches (Monroe et al., 2016), we use a 3-layer Multilayer Perceptron (MLP) with layers of size 32 and ELU intermediate activation functions to map the features of a color patch x to an intermediate scalar value, \hat{x} . Next, we use a sigmoid function on \hat{x} to compute the applicability. The

³McMahan and Stone (2015) argue that S_0 and L_0 so defined represent an equilibrium, where naive interlocutors and strategic interlocutors converge on their interpretations.

⁴All code and data is available at <https://go.rutgers.edu/ugycmlb0>.

series of computations from an HSV color patch to applicability are shown in Equation 3.

$$\phi_{w_k}(x_i) = \sigma(\text{MLP}(\text{FFT}(x_i^{\text{HSV}}))) \quad (3)$$

We implement availability as a vector that is transformed to probability values using the sigmoid function and fit during the training routine.

The model is fit in a two-stage approach. The first stage uses a conditioned language modeling objective: minimize the negative log likelihood of $S_0(w_k|x_i, C)$ in Equation 1. In the second stage, we define a CALIBRATION technique so that the rates of applicability for a description do not encode its frequency in the training dataset. The technique, inspired by work on knowledge distillation (Hinton et al., 2015), forces the applicabilities for each description be close to 1 for at least one training data point. Calibration begins by using the model trained in the first stage to compute applicability values for every training data point. Each description’s vector of applicability values is normalized by their 99th percentile value and bounded in the 0-1 range.

The final step of the calibration technique trains a second model to minimize both the original language modeling objective and a binary cross entropy between the second model’s applicability predictions and the first model’s normalized applicabilities. Both models are trained using the RAdam optimization algorithm (Liu et al., 2019) with a learning rate of 0.01 and a learning rate annealing which decreases the learning rate by 75% if the perplexity of the validation set does not improve for 2 epochs. Training is terminated if the validation perplexity does not improve for 4 epochs.

3.2 The Linguistic Reasoning Model (RGC)

Our linguistic reasoning model extends the XKCD model to enable vague predicates that distinguish a target from its competing alternatives. Recall that, in the XKCD model, the applicability calculation for each description w_k concludes with a sigmoid operation. We conceptualize this as the cumulative distribution function over a random variable τ_{w_k} representing a contextual threshold: the probability w_k applies to color patch x is the probability that x exceeds the contextual threshold τ_{w_k} . Following Meo et al. (2014), a description w_k can then distinguish between the target x_0 and its competing alternatives x_1 and x_2 by committing to the thresholds that distinguish them. The goal of referring to

x_0 and not x_1 or x_2 with w_k requires corresponding comparisons to bound the cumulative distribution τ_{w_k} , shown in Equation 4 and simplified in Equation 5.

$$P(\max(x_1, x_2) < \tau_{w_k} < x_0) \quad (4)$$

$$\phi_{w_k}(x_0) - \max(\phi_{w_k}(x_1), \phi_{w_k}(x_2)) \quad (5)$$

$$:= \psi_{w_k}(x_0, \neg x_1, \neg x_2) \quad (6)$$

To compute the linguistic speaker’s probability distribution over descriptions, we utilize ψ_{w_k} in Equation 6 to replace ϕ_{w_k} in Equations 1 and 2. We refer to this model as REFERENTIAL GOAL COMPOSITION (RGC), reflecting the fact that it decomposes the goal of identifying the target to sub-goals of describing the target and excluding the alternatives.

3.3 The Cooperative Reasoning Model (RSA)

Our cooperative reasoning model extends the XKCD model by adapting the Rational Speech Acts (RSA) model of Monroe and Potts (2015). The basic idea is that the strategic speaker S_1^{RSA} chooses a description w_k for x_i in proportion to the probability that the literal listener, when presented with w_k , will recover the intended referent x_i .

$$S_1^{\text{RSA}}(w_k|x_i, C) = \frac{L_0(x_i|w_k, C)\alpha_{w_k}}{\sum_l L_0(x_i|w_l, C)\alpha_{w_l}} \quad (7)$$

Although RSA generates and interprets scalar implicatures, which assume that the listener will take salience into account in resolving reference, nothing in Equation 7 privileges descriptions that are more naturally applicable to the target referent. The literal listener L_0 ’s interpretations can easily stray from literal meaning—recovering the target object from utterances that fit the target poorly but fit alternative objects worse, as in the case of loose talk considered in Section 2. To model the data of Monroe et al. (2017), it’s important to stay closer to literal meaning and penalize utterances that are poor fits for the target object.

We do this by modifying the RSA formulation so the listener entertains the possibility that they are unfamiliar with or cannot identify the speaker’s intended referent. When the listener adopts this *out-of-context* interpretation (as they will if the speaker’s description is sufficiently unlikely to fit the target), the speaker has not communicated successfully. This gives a pragmatic speaker a reason not to rely on loose talk.

More formally, we define the *out-of-context* interpretation which the listener assigns a probability

$\psi_{w_k}(\neg x_0, \neg x_1, \neg x_2)$ that the description does not apply to any of the potential targets. This leads to a revised listener $L_{0+}(x_i|w_k, C)$ defined as in Equation 8:

$$\frac{\phi_{w_k}(x_i)}{\psi_{w_k}(\neg x_0, \neg x_1, \neg x_2) + \sum_j \phi_{w_k}(x_j)} \quad (8)$$

and a correspondingly revised speaker S_{1+}^{RSA} .

3.4 A Conservative Baseline Model (CB)

In addition to the linguistic reasoning model (RGC) and cooperative reasoning model (RSA), we evaluate a conservative baseline which prioritizes simple, unambiguous referring expressions. When speakers use such expressions, they don't show any evidence of relying on linguistic flexibility or referential implicatures. In fact, key recent results in modeling referential communication use models that exclusively use conservative referring expressions (McDowell and Goodman, 2019).

A conservative speaker uses a description w_k to identify x_i in context C by striking a balance between the literal listener and literal speaker:

$$S_1^{CB(\lambda)}(w_k|x_i, C) \propto L_0(x_i|w_k, C)^\lambda S_0(w_k|x_i, C) \quad (9)$$

The “rationality parameter” exponent λ is typically set to a value substantially greater than 1, which gives the model slim confidence that the listener will do cooperative reasoning to disambiguate. Consequently, w_k will be heavily penalized unless the literal meaning clearly indicates that the distractors do not fit the description. Using the XKCD model, $S_0(w_k|x_i, C)$ simplifies to $\phi_{w_k}(x_i)\alpha_{w_k}$. The difference with S_{1+}^{RSA} in Equation 7 is the additional factor $\phi_{w_k}(x_i)$, which says that w_k should be true of the target, and so penalizes both loose talk and linguistic flexibility.⁵

4 Experiments

Having presented mathematical abstractions that identify linguistic reasoning, cooperative reasoning, and the conservative baseline, we now evaluate how well they fit natural utterances in interactive referential communication. We approach this question in two ways. Section 4.2 takes the naive

⁵This factor was originally proposed by Andreas and Klein (2016) in the context of adding pragmatic reasoning to systems whose fundamental computational operation was sampling true descriptions.

approach of measuring how well each approach explains speaker choices on its own. Ultimately, however, we believe this is somewhat misleading. It's more instructive, we argue, to hypothesize that speakers can use different strategies in different situations. Section 4.3 uses a mixture model to provide evidence that the different models fit different aspects of speakers' language use.

4.1 The Colors in Context Dataset

The data we use to evaluate our models of speaker reasoning comes from Monroe et al. (2017), who asked participants to talk about items in a visual display using a free-form chat interface. On each round of interaction, one human subject, designated the director, was provided privately with a target item from a randomized display of three colors and tasked with instructing the other human subject, designated the matcher, to click on the correct item. The displays varied the relationship between the target and the distractors: in the FAR condition, all three colors were visually dissimilar; in the SPLIT condition, the target had a single visually similar distractor; and in the CLOSE condition, all three colors were visually similar. Overall, 775 subjects participated in 948 games with 50 rounds per game for a total of 47,041 rounds. As shown in Figure 1, some rounds have multiple utterances, resulting in 57,946 utterances in total. To eliminate any confounds of processing complex utterances, our experiments focus on a 23,801 utterance subset created by selecting rounds where the director made a single utterance before the matcher clicked a target and where the director's utterance matched an item from the XKCD lexicon.⁶

4.2 Analyzing Strategies Independently

Our first analysis measures how well each model predicts speaker choices in the filtered dataset. To start, we gathered predictions from the three strategies for every data point. RGC aggressively rules out descriptions which have a higher applicability for one of the alternate objects, resulting in 0 probabilities for 530 examples (6.8%) in the training data. To handle the 0 probabilities, we use Jelinek-Mercer smoothing (Jelinek, 1980) for each strategy's predictions, which uses a tuned hyper parameter to interpolate between the strategy's predictions and the relative frequencies of descriptions

⁶A regular expression approach was used to allow for descriptions like “the blue square” or “the red one”.

Model	Dataset Split		
	train	val/dev	test
S_0	15.50	14.88	13.28
RGC	16.15	15.03	13.32
S_{1+}^{RSA}	14.62	14.05	12.49
$S_1^{CB(2)}$	14.14	13.50	11.84
$S_1^{CB(15)}$	20.76	18.83	16.36
S^{EM}	13.47	12.75	11.30

Table 1: Perplexity scores on the CIC dataset for the linguistic (RGC), cooperative (RSA), conservative (CB), and mixture (EM) models. $S_1^{CB(2)}$ has the lowest independent perplexity and $S_1^{CB(15)}$ is selected in the mixture analysis. A Wilcoxon Signed-Ranks Test indicated all differences are significant ($p < 10^{-4}$).

in the colors-in-isolation dataset. This is the only parameter here that’s estimated from the Monroe et al. (2017) training set. We show the perplexity of each model using the interpolated probabilities in Table 1. Overall, cooperative reasoning S_{1+}^{RSA} and the conservative baseline with a small rationality parameter $S_1^{CB(2)}$ better predict what people say on average.

4.3 Analyzing Strategies as a Mixture

Ultimately, the different models all represent plausible reasoning for speakers. There is no reason to think all speakers are the same. We therefore use a mixture analysis (also known as a latent variable analysis) to understand the predictions for individual items (Zeigenfuse and Lee, 2010; Lee, 2018). Overall, the optimization goal is to maximize the likelihood of the data under a posterior distribution where an observed utterance w_i for color patch x_i is generated by a mixture of each model M_j :

$$P(w_i|x_i, C) = \sum_j P(w_i|x_i, C, M_j)P(M_j)$$

The posterior distribution is maximized using an Expectation-Maximization (EM) routine that iteratively computes the probability of a model conditioned on each data point (the “expectation” step) and the prior probabilities for each model (the “maximization” step) (Bishop, 2006, p. 430). The probabilities for observed items $P(w_i|x_i, C, M_j)$ are the non-smoothed probabilities from the independent model analysis in Section 4.2 and were not updated in the EM routine. We repeat the procedure until convergence.⁷ The result is a set of

⁷Since models are not updated during EM, we define convergence to be when the sum over absolute differences in

inferred prior probabilities for the models as well as overall perplexity for the dataset. The prior probabilities are computed for the training set only and used to evaluate the perplexity on the development and test portions of the dataset.

When generating referential expressions, speakers could be using linguistic or cooperative reasoning, they could be acting more conservatively, or they could even be behaving randomly. We structure the mixture analysis to evaluate these options by pitting the RGC model, the RSA model, a CB model, and two random baselines against each other. For the CB model, we set $\lambda = 15$ by evaluating the mixture analyses for the range $1 \leq \lambda \leq 26$ and selecting the λ that results in lowest perplexity on the development set. For the random baselines, we use both a uniform distribution and the normalized frequency distribution of the XKCD corpus.

Because model predictions typically overlap, EM mixture weights are highly sensitive to outlier predictions where models give low probabilities. Nevertheless, the inferred prior probabilities in Table 2 provide evidence that a heterogeneous mixture of speaker choices do exist in the dataset. We can see this clearly in particular utterances. For example, the EM analysis allowed us to find the divergent cases presented in Figures 2–5.

To better understand what each model explains and how the dialogue evolves, we partition the dataset by both difficulty condition as manipulated by Monroe et al. (2017) and by which model best predicted the speaker’s utterance. For each partition, Table 3 reports the number of cases and the matcher success rate. Additionally, we further break out the cases where the RGC model gave 0 probability to the speaker’s utterance. To test for significance, we use the Mann-Whitney U Test for matcher success and the Wilcoxon Signed-Rank

priors was less than 10^{-6} .

Model M_j	$P(M_j)$	Model M_j	$P(M_j)$
RGC	0.33	$freq^{XKCD}$	0.006
S_{1+}^{RSA}	0.46	Uniform	0.004
$S_1^{CB(15)}$	0.19		

Table 2: The EM-fit prior probabilities for linguistic reasoning (RGC), cooperative reasoning (RSA), conservative baseline (CB), and two random baselines, normalized XKCD frequencies ($freq^{XKCD}$) and a uniform distribution (Uniform). We show the perplexities using these priors as S^{EM} in Table 1.

Winning Model	Matcher Success By Condition		
	FAR (4055)	SPLIT (2657)	CLOSE (1889)
RGC (2437)	98.15% (971)	92.88% (839)	87.44% (637)
S_{1+}^{RSA} (582)	88.73% (222)	79.20% (226)	75.54% (134)
$S_1^{CB(15)}$ (5064)	99.46% (2803)	98.14% (1396)	97.46% (865)
RGC = 0	44.07% (518)	56.80% (59)	58.50% (206)
			(253)

Table 3: Matcher success rates in the test data by difficulty condition and best-explaining model. Counts are shown in parenthesis. The cases where RGC gave 0 probability to the utterance are counted separately (RSA is the overwhelming winner for these cases).

Test for utterance probabilities.⁸

Although RSA has the largest mixture weight, it actually doesn’t score the speaker’s utterance as highly as the other models most of the time (in all conditions, $p < 10^{-4}$), which suggests that cooperative reasoning predicts a wider range of descriptions (each with lower probability). By contrast, CB has a lower mixture weight, but scores higher on more data points than RSA in all conditions ($p < 10^{-4}$), RGC in the FAR condition ($p < 10^{-4}$), and RGC in the SPLIT condition ($p < 10^{-2}$); CB puts strong weight on a subset of likely descriptions that covers most, but not all cases. Indeed, CB seems to choose precise, unambiguous descriptions, while the matcher success rates for linguistic and cooperative reasoning are lower ($p < 10^{-3}$), suggesting that these models do embody risky choices. Linguistic reasoning, as embodied by RGC, seems to be somewhat more successful than cooperative reasoning, as embodied by RSA ($p < 10^{-2}$). Finally, cases where RGC was not able to give a probability to the speaker utterance have far lower matcher success rates ($p < 10^{-2}$); it seems in these cases the matcher was genuinely confused.

5 Discussion and Conclusion

This paper has argued that human speakers in collaborative reference dialogues take diverse strategies: they can stick with clear, precise descriptions;

⁸To accommodate multiple comparisons, we adjust the reported significance levels using Bonferroni correction.

alternatively, they can create innovative interpretations for words; alternatively, they can count on their audience to fill in the gaps in what they say. While computational models often focus on one specific kind of reasoning, we believe that our findings are broadly consonant with the psycholinguistics literature, with its evidence of the psychological difficulty of semantically identifying targets (Clark and Wilkes-Gibbs, 1986), its evidence of the psychological difficulty of taking the audience’s perspective into account (Keysar et al., 2000), and its concepts of “least collaborative effort” (Clark and Schaefer, 1989) in characterizing interaction as fundamental to success in conversation. We are optimistic that future work can continue to develop precise data-driven models that integrate these different explanations to understand and respond to user utterances in dialogue systems.

Our work has a number of limitations that we leave for future research. Even within the simple domain of identifying color patches, we see the utterances that RGC cannot explain—utterances where a speaker seems to refer to a target object with a description that fits the target less well than a distractor—as a strong indication of variability in meaning across individuals. This needs to be accounted for. In addition, it would be good to explore models of reference to colors in context that generalize from colors in isolation data using more flexible machine-learned models of choice.

What about more complex domains and interactions? The challenges of providing fine-grained and wide-ranging analyses of interlocutors’ referential problem-solving strategies remain substantial. Nevertheless, we do see promising directions. One is to follow Elsner et al. (2018) in conceptualizing reference production in terms of a high-level choice of strategy followed by detailed content choices, and build a corresponding probabilistic model of reference production. Another is cover more complex interactions, by including additional interactive strategies for framing alternatives, excluding wrong interpretations, asking clarification questions, and answering them.

Acknowledgments

This research was supported by NSF awards IIS-1526723 and CCF-1934924. Preliminary versions were presented at NYU and Bochum. Thanks to audiences there, the anonymous reviewers, Malihe Alikhani, and Baber Khalid for comments.

References

- Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182.
- Brent Berlin. 1991. *Basic Color Terms: Their Universality and Evolution*. Univ of California Press.
- Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.
- Robyn Carston. 2002. *Thoughts and Utterances: The Pragmatics of Explicit Communication*. Blackwell.
- Herbert H. Clark. 1996. *Using Language*. Cambridge University Press.
- Herbert H. Clark and Catherine R. Marshall. 1981. Definite reference and mutual knowledge. In Arivind Joshi, Bonnie Webber, and Ivan Sag, editors, *Elements of Discourse Understanding*, pages 10–63. Cambridge University Press.
- Herbert H. Clark and Edward F. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13(2):259–294.
- Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.
- Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 18:233–263.
- Kees van Deemter. 2006. Generating referring expressions that involve gradable properties. *Computational Linguistics*, 32(2):195–222.
- Kees van Deemter. 2012. *Not exactly: In praise of vagueness*. Oxford University Press.
- Kees van Deemter. 2016. *Computational Models of Referring: A Study in Cognitive Science*. MIT Press.
- Micha Elsner, Alasdair Clarke, and Hannah Rohde. 2018. Visual complexity and its effects on referring expression generation. *Cognitive science*, 42:940–973.
- Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.
- Delia Graff Fara. 2000. Shifting sands: An interest-relative theory of vagueness. *Philosophical Topics*, 28(1):45–81.
- H. P. Grice. 1975. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics III: Speech Acts*, pages 41–58. Academic Press.
- Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.
- Laurence R. Horn. 1984. Toward a new taxonomy for pragmatic inference: Q-based and R-based implicature. In Deborah Schiffrin, editor, *Meaning, Form, and Use in Context: Linguistic Applications*, pages 11–42. Georgetown University Press.
- Frederick Jelinek. 1980. Interpolated estimation of markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice*, 1980.
- Christopher Kennedy. 2007. Vagueness and grammar: the semantics of relative and absolute gradable adjectives. *Linguistics and Philosophy*, 30(1):1–45.
- Boaz Keysar, Dale J. Barr, Jennifer A. Balin, and Jason S. Brauner. 2000. Taking perspective in conversation: The role of mutual knowledge in comprehension. *Psychological Science*, 11(1):32–38.
- Michael D Lee. 2018. Bayesian methods in cognitive modeling. *Stevens' Handbook of Experimental Psychology and Cognitive Neuroscience*, 5:1–48.
- Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019. On the variance of the adaptive learning rate and beyond.
- R. Duncan Luce. 1959. *Individual Choice Behavior: A Theoretical analysis*. Wiley.
- Bill McDowell and Noah Goodman. 2019. Learning from omission. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 619–628.
- D. McFadden. 1973. Conditional logit analysis of qualitative choice behaviour. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press.
- Brian McMahan and Matthew Stone. 2015. A Bayesian model of grounded color semantics. *Transactions of the Association for Computational Linguistics*, 3:103–115.
- Timothy Meo, Brian McMahan, and Matthew Stone. 2014. Generating and resolving vague color references. In *SEMDIAL 2014: The 18th Workshop on the Semantics and Pragmatics of Dialogue*, pages 107–115.
- Will Monroe, Noah D Goodman, and Christopher Potts. 2016. Learning to generate compositional color descriptions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2243–2248.
- Will Monroe, Robert XD Hawkins, Noah D Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5:325–338.

Will Monroe and Christopher Potts. 2015. Learning in the Rational Speech Acts model. In *Proceedings of 20th Amsterdam Colloquium*, Amsterdam. ILLC.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. **Pytorch: An imperative style, high-performance deep learning library.** In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

C. Raymond Perrault and Philip R. Cohen. 1981. It’s for your own good: a note on inaccurate reference. In Aravind K. Joshi, Bonnie Lynn Webber, and Ivan Sag, editors, *Elements of Discourse Understanding*, pages 217–230. Cambridge University Press.

Craig Roberts. 2003. Uniqueness in definite noun phrases. *Linguistics and Philosophy*, 26(3):287–350.

Eleanor Rosch. 1978. Principles of categorization. In Eleanor Rosch and Barbara B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Erlbaum.

Thomas C. Schelling. 1960. *The Strategy of Conflict*. Harvard University Press.

Dan Sperber and Deirdre Wilson. 1986. *Relevance: Communication and Cognition*. Harvard University Press.

Matthew D Zeigenfuse and Michael D Lee. 2010. A general latent assignment approach for modeling psychological contaminants. *Journal of Mathematical Psychology*, 54(4):352–362.

Contextualized Emotion Recognition in Conversation as Sequence Tagging

Yan Wang Jiayu Zhang Jun Ma Shaojun Wang Jing Xiao

Ping An Technology

{wangyanj61, zhangjiayu470}@pingan.com.cn
{majun, wangshaojun851, xiaojing661}@pingan.com.cn

Abstract

Emotion recognition in conversation (ERC) is an important topic for developing empathetic machines in a variety of areas including social opinion mining, health-care and so on. In this paper, we propose a method to model ERC task as sequence tagging where a Conditional Random Field (CRF) layer is leveraged to learn the emotional consistency in the conversation. We employ LSTM-based encoders that capture self and inter-speaker dependency of interlocutors to generate contextualized utterance representations which are fed into the CRF layer. For capturing long-range global context, we use a multi-layer Transformer encoder to enhance the LSTM-based encoder. Experiments show that our method benefits from modeling the emotional consistency and outperforms the current state-of-the-art methods on multiple emotion classification datasets.

1 Introduction

With the prevalence of conversation-based service, emotion recognition in conversation (ERC) has been attracting attention recently (Majumder et al., 2019; Zhong et al., 2019; Ghosal et al., 2019). Due to great potential in many scenarios such as recommendation system, customer service feedback and health-care, researchers keep focusing on empowering machine to understand emotions in conversation with emotional dynamics, which is a work with challenges lying in several aspects such as modeling the emotion inertia for each speaker and the influence of the interaction between speakers on emotional dynamics (Poria et al., 2019).

Recent works on ERC rely on recurrent neural networks (RNNs) to compute context-dependent representations of utterances (Poria et al., 2017; Majumder et al., 2019). Due to a carefully designed cell, RNNs like long short-term memory

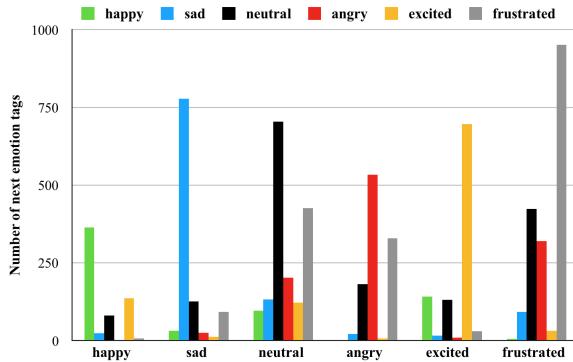


Figure 1: Emotional consistency on IEMOCAP (Busso et al., 2008). In a conversation, similar emotions tend to appear adjacently while dissimilar emotions seldom appear in the neighborhood. We call this phenomenon emotional consistency. For example, if the emotion of current utterance is happy, the tag of next utterance tends to be happy, excited or neutral rather than sad, angry or frustrated. This pattern also applies to other emotions.

(LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Chung et al., 2014) memorize the sequential context to model the dependency between utterances. Such scheme of contextualized emotion recognition has shown its superiority in tracking emotional dynamics by modeling self and inter-speaker dependency in conversations.

Nevertheless, including LSTM and GRU, RNNs are limited in their capability to process tasks involving very long sequences in practice (Bradbury et al., 2016; Khandelwal et al., 2018). For mitigating this issue, the Transformer architecture (Vaswani et al., 2017) and graph convolution networks (GCNs) (Defferrard et al., 2016) have been introduced to ERC for propagating contextual information among distant utterances and yielded state-of-the-art performance (Zhong et al., 2019; Ghosal et al., 2019).

These approaches leverage contextualized utter-

ance features to predict emotion tags, but they ignore the inherent relation between emotion tags. We observe, that the phenomenon of emotional consistency exists widely in conversations, that is, similar emotions are much more likely to appear adjacently than dissimilar emotions, as shown in Figure 1. We surmise modeling the emotional consistency is helpful to find a more reasonable distribution of emotion tags and thus further improves the performance of emotion classification.

In this work, we propose a method to address emotion classification as sequence tagging. For a given conversation, instead of predicting the distribution of emotion tags independently, we consider relations between nearby emotion tags and choose the globally best tag sequence for the entire conversation at once. Hence, we employ a CRF (Lafferty et al., 2001) to take into account the dependency between emotion tags in neighborhoods. Contextualized utterance representations fed into the CRF layer are computed by LSTM-based context encoders. By the aid of individual context encoder, our model tracks the self dependency which depicts emotional inertia of individual speakers. The inter-speaker dependency reflecting the influence of other speakers on a certain speaker is understood by the global context encoder. We use a multi-layer Transformer encoder to enhance the global context encoder so that our model can take advantage of long-range contextual information when computing contextualized utterance representations.

In summary, our contributions are as follows:

- For the first time we model ERC task as sequence tagging and use CRF to model the emotional consistency in conversation. The CRF layer exploits past and future emotion tags to jointly decode the best tag sequence for the entire conversation.
- We apply a multi-layer Transformer encoder to enhancing the LSTM-based global context encoder. The enhanced encoder is able to capture long-range sequential context which is essential for computing contextualized utterance representations.
- Extensive experiments demonstrate that modeling the emotional consistency and long-range contextual dependency promotes the performance of emotion classification. Our method advances the state of the art for ERC on three conversation datasets.

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 describes our sequence labeling architecture. Section 4 presents the experimental setting. Section 5 reports extensive experimental results and makes a detailed analysis. We conclude this paper in Section 6.

2 Related Work

Emotion Recognition in Conversation: Early researches on emotion recognition in conversation mainly use lexicon-based methods and audio features (Lee et al., 2005; Devillers and Vidrascu, 2006). Some open-source conversation datasets with visual, acoustic and textual features have been available in the past few years (Busso et al., 2008; Poria et al., 2018). Along with these datasets, a number of deep learning methods are applied to emotion recognition. Poria et al. (2017) proposes context LSTM to capture contextual information for sentiment classification. DialogueRNN (Majumder et al., 2019) models the emotional dynamics by its party GRU and global GRU. It employs attention mechanisms to pool information from global context for each target utterance. Zhong et al. (2019) proposes Knowledge-Enriched Transformer(KET), which learns structured conversation representation by hierarchical self-attention and external commonsense knowledge. DialogueGCN (Ghosal et al., 2019) applies the graph neural network to context propagation issues present in the current RNN-based methods for ERC and achieves the state-of-the-art performance on multiple conversation datasets.

Transformer: Transformer has achieved great success in various NLP tasks due to its rich representation and high computation efficiency. Self-attention mechanisms endow Transformer with the capability of capturing longer-range dependencies than RNNs. Recent works such as BERT (Devlin et al., 2018) and GPT (Radford et al., 2018) use Transformer encoder and decoder respectively to learn representations on large-scale datasets. These representations are transferred to down-stream tasks such as named entity recognition (NER) and question answering and achieves state-of-the-art results. Dai et al. (2019) introduces the notion of recurrence to address context fragmentation limitations of Transformer. Wang et al. (2019) explores Transformer with additional LSTM layers to better capture the sequential context while retaining the

high computation efficiency.

Sequence Tagging: Sequence tagging has drawn research attention for a few decades. It includes a bunch of NLP tasks such as part of speech tagging (POS), chunking and NER. The most common statistical models for sequence tagging includes hidden Markov model (HMM), maximum entropy Markov model (MEMM) and CRF (Ratnoff and Roth, 2009; Passos et al., 2014). These traditional sequence tagging methods rely heavily on hand-crafted features. In the past few years, convolutional neural networks (CNNs) and RNNs are introduced to tackle sequence tagging problems and achieves competitive performance against traditional methods (Graves et al., 2013; Chiu and Nichols, 2016). Huang et al. (2015) has pointed out that the combination of bidirectional LSTM and CRF can efficiently use both past and future input features as well as past and future tags information. Hence, BiLSTM-CRF model produces state-of-the-art results on many sequence tagging tasks.

3 CESTa: Contextualized Emotion Sequence Tagging

Existing works (Majumder et al., 2019; Zhong et al., 2019; Ghosal et al., 2019) define the ERC task as the prediction of emotion tags of constituent utterances. However, emotional consistency which is an important characteristic of the conversation is not taken into consideration. CESTa differs from those methods in that it treats ERC as a task of sequence tagging of which performance is generally improved by choosing the globally best set of tags for the entire sequence at once. To this end, CESTa employs a CRF to take advantage of past and future tags to predict the current tag. For the t th utterance in a conversation, the textual feature \mathbf{u}_t is extracted by a single-layer CNN and fed into the global and individual context encoders which learn inter-speaker and self dependency respectively. Moreover, the global context encoder is enhanced by a number of Transformer blocks to propagate long-range contextual information effectively. The concatenation of the global context encoding \mathbf{g}_t and individual context encoding \mathbf{s}_t is considered as a matrix of scores and fed into the final CRF layer. The overall architecture is shown in Figure 2.

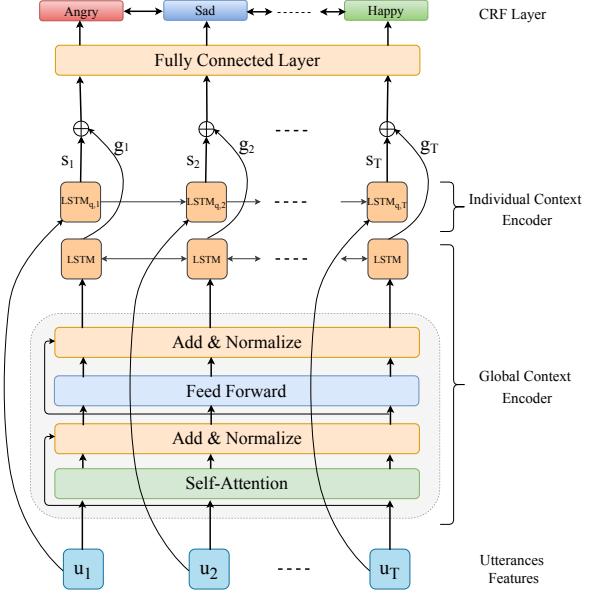


Figure 2: Overview of CESTa. The Transformer-enhanced global context encoder takes the textual feature \mathbf{u}_t of the t th utterance in a conversation as input and produces encoding \mathbf{g}_t . Also, \mathbf{u}_t is fed into the individual context encoder to update states for the corresponding speaker of which index is $q = q(\mathbf{u}_t)$ and outputs another encoding \mathbf{s}_t . A CRF layer is applied over the concatenation of each \mathbf{g}_t and \mathbf{s}_t to obtain the final prediction for each utterance in the conversation.

3.1 Utterance Feature Extraction

We employ convolutional neural networks (CNNs) to extract textual features for each utterance. Following Kim (2014), we use a simple architecture consisting of a single convolutional layer followed by max-pooling layer and a fully connected layer. Specifically, three distinct convolutional filter region sizes of 3, 4, 5 are used to obtain n-gram features. For each region size, we use 100 filters to learn complementary features. The max-pooling results of each feature map are activated by a rectified linear unit (ReLU) and concatenated before fed into a fully connected layer consisting of 100 hidden units, of which the activation forms the utterance representation.

We explore two methods to train this network. It can be trained jointly with CESTa and thus its gradients will be updated during the training of the whole architecture. On the other hand, it also can be trained as an individual task of utterance classification with emotion tags. According to characteristics of different datasets, we choose pertinent strategies for the utterance feature extraction. The strategy choices are reported in Section 4.3.

3.2 Global Context Encoder

It is essential to take the contextual information into account when classifying an utterance in a sequence since other utterances in this sequence have a substantial effect on the emotion of current utterance. In other words, the emotion of current speaker can be forced to change by utterances of counterparts. This fact reflects the inter-speaker dependency which is closely related to the tendency for speakers to mirror their counterparts during the conversation (Navarretta, 2016) and is crucial to model emotional dynamics in a conversation.

Given the sequential nature of the conversation, we employ a bidirectional LSTM (BiLSTM) to capture the contextual information. However, modeling the long-range contextual information is a weakness of RNNs. Due to self-attention mechanisms, the Transformer is superior to RNN-based models in processing long-range context. Hence, we use a multi-layer Transformer encoder to enhance the context encoder. Specifically, the enhanced context encoder takes textual features of utterances as input, applies a multi-head self-attention operation (Vaswani et al., 2017) over them followed by point-wise fully connected feed-forward layers to produce contextualized vectors of utterances. Finally, contextualized utterance representations are fed into the BiLSTM layer which fuses long-range sequential contextual information to produce the context encoding:

$$\begin{aligned} \mathbf{h}_0 &= (\mathbf{u}_1, \dots, \mathbf{u}_T) \\ \mathbf{h}_l &= \text{TransformerBlock}(\mathbf{h}_{l-1}), l \in [1, N] \\ \mathbf{g}_t &= \text{BiLSTM}_t(\mathbf{h}_N^t), t \in [1, T] \end{aligned} \quad (1)$$

where N is the number of Transformer layers, T is the length of conversation, \mathbf{g}_t is the context encoding that is formed by the concatenation of left context vector $\overrightarrow{\mathbf{g}}_t$ and right context vector $\overleftarrow{\mathbf{g}}_t$, which is generated by a forward LSTM and a backward LSTM respectively.

3.3 Individual Context Encoder

Individual context encoder keeps track of the self dependency which reflects the emotional influence that speakers have on themselves during the conversation. Under the effect of emotional inertia, each individual speaker in a conversation tends to maintain a stable emotional state during the conversation until counterparts lead into changes (Poria et al.,

2019). Since our model is only evaluated on textual modality, we hypothesize the self-dependency of each individual speaker could be deduced by its own textual utterances. This leads to an effective but simpler speaker-level context encoder than those used in other works (Majumder et al., 2019; Ghosal et al., 2019).

We implement an LSTM as the individual context encoder to output all speaker states for each time step. It exploits the current input utterance to update states only for the corresponding speaker. Specifically, for the t th utterance in a conversation, let $q = q(\mathbf{u}_t)$ denote the speaker of \mathbf{u}_t . The state $\mathbf{s}_{q,t}$ of an individual speaker q at timestep t in the conversation is updated by the following formula:

$$\mathbf{s}_{q,t} = \text{LSTM}_{q,t}(\mathbf{u}_t) \quad (2)$$

where $\mathbf{s}_{q,t}$ is specific to the speaker q and is updated by the current utterance \mathbf{u}_t while excluding utterances from other speakers.

3.4 CRF Layer

Inspired by the emotional consistency of conversations, we consider ERC as a task of sequence tagging which is beneficial to consider the correlations of nearby tags and choose the globally best chain of tags for a given input sequence. For this reason, a CRF is employed in CESTa to yield final predictions with the aid of neighboring tags. In our scenario, $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_T)$ represents an input sequence where \mathbf{u}_t is the feature vector of the t th utterance, $\mathbf{y} = (y_1, \dots, y_T)$ represents a generic sequence of tags for \mathbf{U} , $Y(\mathbf{U})$ represents all possible tag sequences for \mathbf{U} . The probability of \mathbf{y} is generated by a softmax over all possible tag sequences:

$$p(\mathbf{y} | \mathbf{U}) = \frac{e^{s(\mathbf{U}, \mathbf{y})}}{\sum_{\mathbf{y}' \in Y(\mathbf{U})} e^{s(\mathbf{U}, \mathbf{y}')}} \quad (3)$$

where $s(\mathbf{U}, \mathbf{y})$ is the score for \mathbf{y} which is given by the sum of two matrices: one $K \times K$ matrix of transition scores, one $T \times K$ matrix of scores comes from the concatenation of the global and individual context encoding, K is the number of distinct tags.

During training, we maximize the log-likelihood of correct tag sequences for a training set $\{(\mathbf{U}_i, \mathbf{y}_i)\}$, which is given by:

$$L = \sum_i \log(p(\mathbf{y}_i | \mathbf{U}_i)) \quad (4)$$

Dataset	#Dialogues(Train/Val/Test)	#Utterances(Train/Val/Test)	#Classes
IEMOCAP	108/12/31	4810/1000/1523	6
DailyDialogue	11118/1000/1000	87170/8069/7740	7
MELD	1038/114/280	9989/1109/2610	7

Table 1: Statistics of training, validation and test datasets. For IEMOCAP, we use 10% of the training dialogues as the validation dataset. For DailyDialogue and MELD, we split train/val/test according to the same ratio provided by Zhong et al. (2019).

While decoding, we search for the tag sequence that obtains the maximum score, given by:

$$\mathbf{y}^* = \arg \max_{\mathbf{y} \in Y(\mathbf{U})} s(\mathbf{U}, \mathbf{y}) \quad (5)$$

Since we only model interactions of two successive tags, both the training and decoding can be solved efficiently by dynamic programming (Rabiner, 1989). In addition, it is favourable for improving results to apply a non-linear transformation to the concatenation of the global and individual context encoding before feeding it into the CRF layer (Lample et al., 2016). Accordingly, results with our method reported in Section 5 incorporate an extra hidden layer.

4 Experimental Setting

4.1 Datasets

For ease of comparison with state-of-the-art methods, we evaluate CESTa on three artificial conversation datasets: IEMOCAP (Busso et al., 2008), MELD (Poria et al., 2018) and DailyDialogue (Li et al., 2017) rather than natural emotions corpus such as LEGO (Schmitt et al.; Ultes et al., 2015). IEMOCAP and MELD are both multimodal datasets with visual, acoustic and textual features, while DailyDialogue only contains textual features. For this work, we focus on emotion recognition in textual conversation. These three datasets are all split into training, validation and test datasets. The statistics are reported in Table 1.

IEMOCAP: This dataset contains five sessions, each of them was recorded from two actors. Training dataset is composed of dyadic conversations from session one to four. Annotations of utterances include six basic emotions, namely happy, sad, neutral, angry, excited and frustrated.

DailyDialogue: DailyDialogue is a human-written dyadic conversation dataset, reflecting daily communication way and covering various topics about human daily life. Emotion labels contains

anger, disgust, fear, happiness, sadness, surprise and other. Since DailyDialogue does not provide speaker information, we treat utterance turns as speaker turns by default.

MELD: Multimodal Emotion Lines Dataset (MELD) is collected from TV-series *Friends* containing 1438 multi-party conversations. Each utterance is annotated with one of the seven emotion labels including happy/joy, anger, fear, disgust, sadness, surprise and neutral.

4.2 Baselines

CNN (Kim, 2014): A single-layer CNN which is identical to our utterance feature extraction network described in Section 3.1, which is the only baseline model without modeling contextual information.

CNN+cLSTM (Poria et al., 2017): Textual features of utterances are obtained by a CNN, over which a context LSTM (cLSTM) is applied to learn the contextual information.

DialogueRNN (Majumder et al., 2019): The RNN-based method that models both context and speaker information. After extracting textual features by a fine-tuned CNN, DialogueRNN applies global GRU and party GRU to the task of modeling speaker state and contextual information respectively.

DialogueGCN (Ghosal et al., 2019): Textual utterance features are extracted by a CNN in the same way as DialogueRNN does before they are fed into a bidirectional GRU to capture contextual information. After that, a graph convolutional network is applied to modeling speaker-level information. Contextual features and speaker-level features are concatenated and a similarity-based attention mechanism is used to obtain utterance representations for the final classification.

KET (Zhong et al., 2019): Enriched by the external commonsense knowledge, KET employs the

Models	IEMOCAP						DailyDialogue	MELD	
	Happy	Sad	Neutral	Angry	Excited	Frustrated	Avg.(w)	Avg.(micro)	Avg.(w)
CNN	35.34	53.66	51.61	62.17	50.66	55.56	51.28	49.27	55.86
CNN+cLSTM	33.90	69.76	48.40	57.55	62.37	57.64	56.04	51.84	56.87
DialogueRNN	37.94	78.08	58.95	64.86	68.11	58.85	62.26	51.64	57.07
DialogueGCN	42.75	84.54	63.54	64.19	63.08	66.99	64.18	-	58.10
KET	-	-	-	-	-	-	59.56	53.37	58.18
CESTa	47.70	80.82	64.76	63.41	75.95	62.65	67.10	63.12	58.36

Table 2: Comparisons with baselines and state-of-the-art methods. Best performances are highlighted in bold.

Transformer encoder to capture the contextual information and uses the Transformer decoder to predict the emotion tag for the target utterance.

4.3 Training Setup

All three datasets are preprocessed by lower-casing and tokenization¹. In order to relieve the effect of out-of-vocabulary (OOV) words, we also impose a stemming procedure on these datasets.

GloVe vectors trained on Common Crawl 840B with 300 dimensions are used as fixed word embeddings. We use a 12-layers 4-heads Transformer encoder of which the inner-layer dimensionality is 2048 and the hidden size is 100. The number of hidden units of both context BiLSTM and speaker LSTM is 30. Along with a batch size of 64 and learning rate of 0.0005, the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$ is used throughout the training process.

Note that due to utterances in the MELD dataset rarely contain emotion specific expressions, our model needs more expressive utterance features which can be extracted by a separate fine-tuned CNN. According to (Majumder et al., 2019; Ghosal et al., 2019), we train a CNN at utterance level with the emotion labels for MELD. As for datasets of IEMOCAP and DailyDialogue involving rich emotion representations in utterances, a CNN to extract textual features is trained jointly with the whole architecture of our model.

5 Results and Discussions

5.1 Comparison Results

We compare the performance of our model with baseline methods, as shown in Table 2. Note that

Dataset	Max.	Min.	Avg.
IEMOCAP	110	8	50
DailyDialogue	35	2	8
MELD	33	1	10

Table 3: Statistics of conversation length of three datasets.

statistics of conversation lengths which play an important role in ERC vary greatly between different datasets, as shown in Table 3, the performance of our model on different datasets changes accordingly, as what we analyze in the following.

IEMOCAP: The weighted macro-F1 is used as the evaluation metric following (Majumder et al., 2019; Ghosal et al., 2019; Zhong et al., 2019). F1 scores of individual labels are also reported since the six emotion labels in IEMOCAP are unbalanced. As evidenced by Table 2, our model is around 3% better than DialogueGCN, 5% better than DialogueRNN and at least 7.5% better than all other baseline models.

To explain the gap in performances, one major reason is that some models like CNN, CNN+cLSTM and KET neglect the speaker-level information modeling so that models will treat utterances equally from different speakers, leading to certain loss in performance. Besides, considering that the average conversation length in IEMOCAP is 50 and the maximum length exceeds 100, the Transformer is capable of better capturing long-range dependency compared to RNNs-based context encoders like LSTM or GRU. Moreover, our model utilizes CRF to exploit the influence that past and future tags have on the current tag, which is not taken into account by any of existing models. We surmise that the CRF layer takes the emotional consistency into consideration when classifying similar emotions, such as "happy" and "excited", hence CESTa is aware of the similarity between them and

¹https://www.tensorflow.org/datasets/api_docs/python/tfds/features/text/Tokenizer

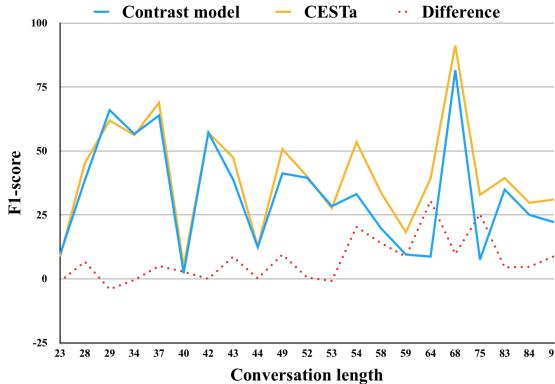


Figure 3: The performance of different models on conversations with different length. Yellow solid: our CESTa. Blue solid: the contrast model with only LSTM-based global context encoder. Red dotted: the difference between CESTa and the contrast model.

outperforms other models on these emotions.

DailyDialogue: In this dataset, the majority class(neutral) accounts for more than 80% in the test dataset. We use the micro-averaged F1 excluding the neutral class as the evaluation metric due to the imbalanced data distribution. DailyDialogue contains lots of short dyadic conversations of which average length is 8, this leads to frequent speaker turnovers. In this case, modeling speaker-level information with speaker encoder releases more ability in improving the performance. According to Li et al. (2017), DailyDialogue contains rich emotions so that our model can learn more expressive representations for utterances. Furthermore, DailyDialogue reflects human communication style, which means a definite emotional consistency can be utilized by the CRF layer in CESTa. This explains the reason of our model outperforming baselines by a large margin.

MELD: On MELD, we follow the same metric used on IEMOCAP. The performance differences between baseline models and our CESTa is not as contrasting as they are on IEMOCAP and DailyDialogue. This is mostly because of the nature of MELD. In MELD, there are many conversations containing more than 5 speakers while the average conversation length is only 10 and the minimum length is only 1. For short conversations, the advantage of the Transformer which is superior to RNNs in capturing the long-range inter-speaker dependency is not obvious. In general, majority of the speakers attending the conversation in MELD only utter a small amount of utterances. This leads

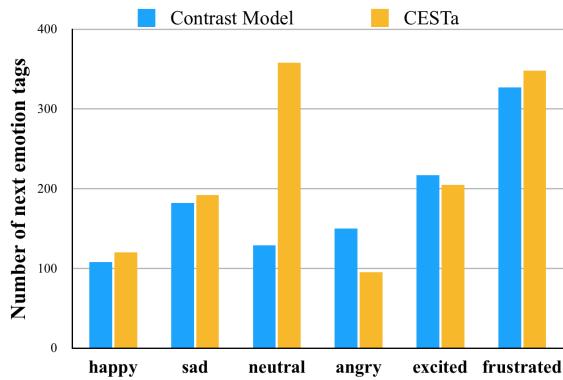


Figure 4: Statistics of pairs consisting of two identical tags which are consecutive in the conversation given by different models. Yellow: our CESTa. Blue: the contrast model without the CRF layer.

the difficulty of modeling the self dependency. Additionally, utterances in MELD suffer a shortage of emotion specific expressions, this further increases the difficulty for emotion modeling. Nevertheless, CESTa achieves better results than baselines. We attribute this to the CRF layer which has an insight into the emotional consistency.

5.2 Model Analysis

Analysis of Transformer Enhancing: We evaluate the effect of Transformer enhancing on conversations with different lengths. On the test dataset of IEMOCAP, conversations are grouped by length and fed into two models: one is our CESTa with the Transformer-enhanced global context encoder, another is the contrast model that only uses LSTM-based global context encoder. The average F1 score of different groups are shown in Figure 3.

It is easy to observe that both context encoders have similar effect on relatively short conversations. However, the advantage of Transformer enhancing are more obvious as the length of conversation exceeds 54. This confirms the contribution of Transformer to the modeling of long-range contextual information.

Analysis of Emotional Consistency: We experiment on the test dataset of IEMOCAP to check the fitting of emotional consistency. We compare two models: one is our CESTa with the CRF layer, another is the contrast model that uses a softmax layer instead of CRF for classification. Statistics are given by Figure 4.

For most emotion tags, CESTa demonstrates a more obvious emotional consistency, that is, the same tag are more likely appear adjacently in given

Transformer	LSTM-based Global Context Encoder	Individual Context Encoder	CRF	IEMOCAP	DailyDialogue
No	Yes	Yes	Yes	64.25	60.28
Yes	No	Yes	Yes	64.86	59.13
Yes	Yes	No	Yes	62.35	57.10
Yes	Yes	Yes	No	65.31	60.17
Yes	Yes	Yes	Yes	67.10	63.12

Table 4: Ablation results on IEMOCAP and DailyDialogue.

conversations. We assume CESTa has learnt the emotional consistency very well and thus achieves a better performance. For the tag of “angry” and “excited”, CESTa reflects less emotional consistency than the contrast model. However, we find that the quantitative distribution of next tags of “angry” and “excited” given by CESTa is closer to the ground truth than the contrast model. This trade-off between the emotional consistency and evaluation of performance is worth to further study.

Ablation Study: We conduct ablation study to investigate the necessities of the Transformer enhancing, global context encoder, individual context encoder and the CRF layer. The study is performed on IEMOCAP and DailyDialogue by removing one component at a time. Results are given in Table 4.

The results align with our analysis as the four components all improve performance by varying extents. The individual context encoder contributes most of the improvements against the baseline on both datasets. This shows the individual context encoder can capture emotional inertia for each speaker.

For IEMOCAP, the Transformer enhancing brings CESTa almost 3% increase of performance, which is the second biggest increase only after the increase 4.75% brought by the individual context encoder. For DailyDialogue, the dataset of which conversations are generally short, the Transformer enhancing leads to the minimum growth of performance. This demonstrates the importance of the Transformer enhancing for processing long conversations.

For both datasets, the performance falls by 2.24% and 3.99% respectively if we remove the LSTM-based global context encoder while keeping only the Transformer encoder. This demonstrates the importance of sequential contextual information captured by LSTM. Also, the CRF layer contributes 1.69% and 2.95% respectively to our model performance on IEMOCAP and DailyDialogue by

optimizing globally with past and future emotion tags which contain information of emotional consistency.

Together these results provide important insights into what really counts in ERC. First, long-range sequential global context encoder is essential for emotion recognition in conversation. Modeling adequate contextual information enables the model to know the background of the current utterance. Besides, with the help of individual context encoder, emotion inertia can be learned by our model to seize the personality of the current speaker. Finally yet importantly, emotion tags flowing throughout a conversation to some extent have coherence naturally, which makes it meaningful to exploit the influence that past and future emotion tags have on the current tag with CRF.

6 Conclusion

We have introduced a new method, CESTa, to model ERC task as sequence tagging. Based on the contextualized utterance representations, it leverages past and future emotion tags to jointly decode the best tag sequence for the entire conversation at once. We conduct numerous experiments on three benchmark datasets. Through ablation studies, we have confirmed modeling the emotional consistency via CRF and enhancing the context encoder via the Transformer are beneficial to our model. Experimental results show that CESTa leads to a further performance improvement against strong baselines and achieves new state-of-the-art results.

Future works will focus on the representation of emotional consistency for each interlocutor in the conversation. We also plan to incorporate multi-modal information into CESTa and evaluate it on more natural conversation datasets. Since CESTa needs to use emotion information of the whole dialogue, we will study its performance on the online dialogue system which has no access to the information of future emotions.

References

- James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2016. Quasi-recurrent neural networks. *arXiv preprint arXiv:1611.01576*.
- Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeanette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335.
- Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *CoRR*, abs/1901.02860.
- Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852.
- Laurence Devillers and Laurence Vidrascu. 2006. Real-life emotions detection with lexical and paralinguistic cues on human-human call center dialogs. In *Ninth International Conference on Spoken Language Processing*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. Dialoguegen: A graph convolutional neural network for emotion recognition in conversation. *arXiv preprint arXiv:1908.11540*.
- Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.
- John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.
- Chul Min Lee, Shrikanth S Narayanan, et al. 2005. Toward detecting emotions in spoken dialogs. *IEEE transactions on speech and audio processing*, 13(2):293–303.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.
- Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2019. Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6818–6825.
- Costanza Navarretta. 2016. Mirroring facial expressions and emotions in dyadic conversations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 469–474.
- Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. *CoNLL-2014*, page 78.
- Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–883.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *arXiv preprint arXiv:1905.02947*.

Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155.

Alexander Schmitt, Stefan Ultes, and Wolfgang Minker. A parameterized and annotated spoken dialog corpus of the cmu let’s go bus information system.

Stefan Ultes, María Jesús Platero Sánchez, Alexander Schmitt, and Wolfgang Minker. 2015. Analysis of an extended interaction quality corpus. In *Natural Language Dialog Systems and Intelligent Assistants*, pages 41–52. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Chenguang Wang, Mu Li, and Alexander J. Smola. 2019. Language models with transformers. *CoRR*, abs/1904.09408.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. *arXiv preprint arXiv:1909.10681*.

How Self-Attention Improves Rare Class Performance in a Question-Answering Dialogue Agent

Adam Stiff, Qi Song, Eric Fosler-Lussier

Department of Computer Science and Engineering

The Ohio State University

stiff.4@osu.edu, song.1602@osu.edu,

fosler-lussier.1@osu.edu

Abstract

Contextualized language modeling using deep Transformer networks has been applied to a variety of natural language processing tasks with remarkable success. However, we find that these models are not a panacea for a question-answering dialogue agent corpus task, which has hundreds of classes in a long-tailed frequency distribution, with only thousands of data points. Instead, we find substantial improvements in recall and accuracy on rare classes from a simple one-layer RNN with multi-headed self-attention and static word embeddings as inputs. While much research has used attention weights to illustrate *what* input is important for a task, the complexities of our dialogue corpus offer a unique opportunity to examine *how* the model represents what it attends to, and we offer a detailed analysis of how that contributes to improved performance on rare classes. A particularly interesting phenomenon we observe is that the model picks up implicit meanings by splitting different aspects of the semantics of a single word across multiple attention heads.

1 Introduction

Many semantic classification tasks have seen a huge boost in performance in recent years (Wang et al., 2018, 2019), thanks to the power of contextualized language models such as BERT (Devlin et al., 2019), which uses a Transformer (Vaswani et al., 2017) architecture to produce context-specific word embeddings for use in downstream classification tasks. These large, data-hungry models are not always well suited to tasks that have a large number of classes or relatively small data sets (Mahabal et al., 2019).

One task having both of these inauspicious properties is the Virtual Patient corpus (Jin et al., 2017), a collection of dialogues between medical students and a virtual patient experiencing back pain. The

corpus contains examples of nearly 350 questions that the virtual patient knows how to answer, and the interaction is modeled as a text-based conversation in which the human, as the interviewer of the patient, always has the conversational initiative. Thus, the corpus represents a question identification task from the perspective of the dialogue agent, in which natural language inputs must be mapped to semantically equivalent classes, so that the appropriate fixed response can be returned to the user to achieve the desired pedagogical objectives.¹

Many of the classes in this task are distinguished in subtle ways, e.g., in degree of specificity (“Are you married?” vs. “Are you in a relationship?”) or temporal aspect (“Do you [currently] have any medical conditions?” vs. “Have you ever had a serious illness?”). A few classes are very frequent, but many appear only once in the data set, with almost three quarters of the classes comprising only 20 percent of the examples (Jin et al., 2017).

The current best approach to this task uses an ensemble of Text CNNs (Kim, 2014) combined with a rule-based dialogue manager (Wilcox, 2019) via a logistic regression model, to leverage complementary performance characteristics of each system on the rare classes (Jin et al., 2017). This approach naïvely treats all classes as orthogonal, so the semantic similarity of the classes above can be problematic. Ideally, a model should be able to learn the semantic contributions of common linguistic substructures from frequent classes, and use that knowledge to improve performance when those structures appear in infrequent classes.

We hypothesize that multi-headed attention mechanisms may help with this kind of generalization, because each head is free to specialize, but should be encouraged to do so cooperatively to

¹We are currently working to anonymize this corpus, and we will release code and data at <https://github.com/OSU-slatelab/> when it is available.

maximize performance. Three different methods of utilizing BERT-based architectures for this task surprisingly did not improve upon the performance of the CNN models of Jin et al. (2017). In contrast, a very simple RNN equipped with a multi-headed self-attention mechanism improves performance substantially, especially on rare classes. We assess the reasons for this using several techniques, chiefly, visualization of severely constrained intermediate representations from within the network, and agglomerative clustering of full representations. We find evidence that independent attention heads: 1) represent the same concepts similarly when they appear in different classes; 2) learn complementary information; and 3) may learn to attend to the same word for different reasons. This last behavior leads to discovery of idiomatic meanings of some words.

2 Related Work

Self-attention, in which a model examines some hidden representation to determine which portions of that representation should be passed along for further processing, became prominent relatively recently (Vaswani et al., 2017; Lin et al., 2017). These models have been very successful for some tasks (Wang et al., 2019), but other approaches may work better for classification tasks with many classes and few examples (Mahabal et al., 2019). We explore two types of self-attentive models for a virtual patient dialogue task (Danforth et al., 2013; Jaffe et al., 2015), which has many classes and scarce data. Previous authors have used memory networks (Weston et al., 2015) to improve performance on rare classes for this task (Jin et al., 2018).

Despite the contrast presented above, our self-attentive model actually shares characteristics with the work by Mahabal et al. (2019), as we find that individual word tokens carry parallel meanings.

We present a detailed analysis of our model’s behavior using clustering and visualization techniques; this bears a resemblance to the analysis by Tenney et al. (2019), although they use internal representations to make predictions for linguistic tasks, rather than examining correlations between representations and individual input tokens.

3 Task and Data

As described above, our task is a text-based question-answering task for an agent that has a fixed set of responses. The goal is to classify input queries as paraphrases of canonical questions that

the agent knows how to answer, so we call this a question identification task.

Data are collected from actual user interactions with a virtual patient, which is a graphical avatar with a text input interface and subtitles as output. After collection, the system’s responses are annotated as correct or not, and if not, annotated with the correct label. Jin et al. (2017) used a data set consisting of 4,330 inputs, comprising 359 classes. We extended this data set by replicating the hybrid system described in their work, and deploying it to collect more data. This resulted in a combined data set of 9,626 examples over 259 dialogues.

We noticed that the annotation method for the data used by Jin et al. (2017) introduced a bias for classifications that produce acceptable *responses*, since only examples deemed to be incorrect were reviewed to identify the correct class. Since our evaluation metrics are on the basis of classes and not the agent’s responses, we re-annotated every example, with the aim of maximizing the semantic equivalence of members of the same class. This resulted in the elimination and addition of some classes, leaving 348 in the re-annotated set. The long-tailed distribution is no less a problem in the re-annotated set than in the original, but our baseline outperforms theirs since we use cleaner data.

We hold out 2,799 examples from the combined set as a test set, and perform tenfold cross-validation on the training set for development. The test set only contains 268 classes, but fifteen are unseen in the training data (other than the canonical question, see Appendix A).

4 Experimental Design and Results

We start from a **Text-CNN** baseline for this task (Jin et al., 2017), utilizing a single stream system for comparisons. This system convolves GloVe word embeddings with 300 filters of widths 3, 4, and 5; the max of each filter over the sequence serves as input to a fully connected leaky ReLU layer (Nair and Hinton, 2010), followed by a softmax layer.

We compare this against two contextual models: the relatively well known **Fine-tuned BERT** (Devlin et al., 2019) using the pretrained base model², as well as a variant of a simpler **RNN model with**

²<https://github.com/google-research/bert>

System	Acc. (%)	F1
Baseline CNN	80.7	55.6
BERT Fine-tune	79.8	46.6
Self-attention RNN	82.6	61.4
BERT Static CNN	76.9	49.4
BERT Contextual CNN	75.3	45.2
Mean-pool RNN	81.8	59.4
Bottleneck RNN	80.8	57.2

Table 1: Dev set results comparing different models (top, Sec. 4), word embeddings (middle, Sec. 5.1), and attentional mechanisms (bottom, Sec. 5.2).

self-attention (Lin et al., 2017).³ Note that despite extensive experimentation, only minor modifications of the work of Lin et al. (2017) proved beneficial for our task, so the architecture we describe here is not a novel contribution.

The self-attentive RNN is a single-layer BiGRU (Cho et al., 2014) equipped with a two-layer perceptron that takes hidden states as inputs, and produces one attention score for each of eight attention heads, for each input step. These scores are then softmaxed over the input, and the attention-weighted sum of the corresponding hidden states serves as the value of the attention head. These values are concatenated and fed into a fully connected layer with tanh activations, and a softmax output determines the class. We use dropout of 0.5 in the attention module and in the fully connected layer. The size of hidden states in the BiGRU is 500 dimensions (in each direction), the size of the hidden layer in the attention module is 350 units, and the fully connected classification layer has 500 dimensions. The original model utilizes an orthogonality constraint on the attention vectors for each attention head, but we find that this is detrimental to our task, so we disable it.

Training parameters for all three models are provided in Appendix A.

The development set results (top 3 lines of Table 1) were a bit surprising to us: while we expected that contextual models would outperform the baseline CNN, fine-tuned BERT performed comparatively poorly. The Self-attention RNN, however, performed significantly better than the baseline CNN, which carries over to a smaller degree to the test set (CNN: 76.2% accuracy, 51.9% F1; RNN:

79.1% accuracy, 54.7% F1).⁴ A breakdown of accuracy by class frequency quintiles for the test results is shown in Figure 1, to emphasize the relationship between F1 and rare class performance.

In particular, the BERT model has a very low F1, likely because of the large number of subtly distinguished classes, the relatively small data set, and the high degree of freedom in the BERT model. That is, BERT may be representing semantically similar sentences in nearby regions of the representation space, but with enough variation within those regions that our training set does not permit enough examples for the classifier to learn good boundaries for those regions. Alternatively, the masked language modeling task may simply not induce the grammatical knowledge required to distinguish some classes well.

The success of one attention-based contextual model (Self-attention RNN) and the failure to improve of another (Fine-tuned BERT) led us to ask two analytical questions: first, are the BERT representations not as appropriate for the Virtual Patient dialog domain compared to GloVe embeddings? Second, is there something that we can learn about how the attention-based method is helping over the CNN (and particularly on F1)?

5 Analysis

5.1 Why did BERT perform less well?

The difference in accuracy from the baseline CNN model to the BERT fine-tuning result is fairly small, while the drop in F1 is substantial. Since there are many more infrequent than frequent classes, this suggests that BERT is seriously underperforming in the least frequent quintiles, and making up for it in the most frequent. That, in turn, supports the interpretation that small numbers of examples are inadequate to train a classifier to handle the variation in representations that come out of a contextualized model. This would be consistent with other research showing poor performance of BERT in low-data regimes (Mahabal et al., 2019). Some of the discrepancy may also be explained by a domain mismatch. The BERT base model is trained on book and encyclopedia data (Devlin et al., 2019), to provide long, contiguous sequences of text. In contrast, our inputs are short, conversational, and full of typos. GloVe.42B, trained on web data (Pennington et al., 2014), may simply be a better fit for

³<https://github.com/ExplorerFreda/Structured-Self-Attentive-Sentence-Embedding>

⁴We only tested on the baseline and best system in this paper to minimize use of the test set for future work.

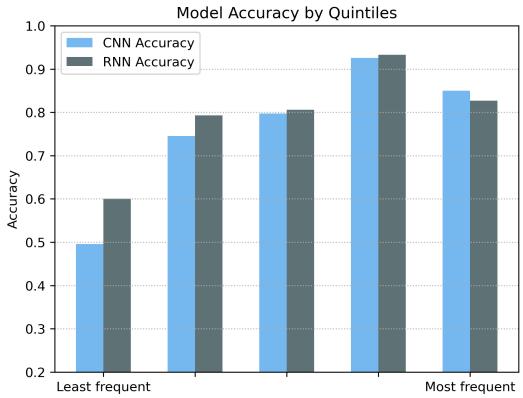


Figure 1: Quintile accuracies for the tested RNN and CNN baseline

our corpus.

To try to tease apart the contributions of model architecture and learned representations, we utilized two different embeddings within the CNN: the **contextual BERT** embeddings from the first layer⁵ of the BERT model, and a **static BERT** embedding for each token calculated from the average contextual embedding over all instances of the token in our corpus.

The worst of our BERT-based models is the full contextualized embeddings fed into the baseline CNN. Since the classification architecture is the same as the baseline, this suggests that a significant contributor to the reduced performance of the BERT-based models is the contextualized representations themselves. It seems that stable representations of lexical items are beneficial for generalizing to unseen sentences when few training examples are available. Consistent with this, the static BERT CNN result, despite a lower accuracy than the fine-tuning result, shows a gain in F1. Again, this supports the idea that variation is harmful for rare classes, since stable representations of informative words for those classes help.

5.2 Analyzing the Self-attention RNN

One question is how much attention versus recurrence is playing a role in the Self-attention RNN’s improvements. We replaced the attention mechanism with **mean pooling** over the input, controlling for parameter counts by replicating the mean hidden state once for each attention head; Table 1 shows that performance is intermediate between the CNN and the self-attentive RNN, suggesting

⁵Empirically, and surprisingly, this worked better than other layers.

that the attention does play a role.

To better understand the behavior of the self-attentive RNN, we employ a relatively novel method of analyzing attention: we insert **bottleneck layers** of just eight dimensions after each attention head, with sigmoid activations and no dropout. This adds another nonlinearity into the model, but reduces the total number of parameters substantially. Color coding gives an easily interpretable representation of both *what* each head is attending to, as well as *how* it represents it. Examples are shown in Figure 2. The bottleneck RNN and CNN have similar overall performance (Table 1), but the RNN’s performance on the least frequent classes is still superior.

By finding the greatest Jensen-Shannon divergence between predictions made by the baseline CNN and the RNN, as well as the largest change in class recall between the systems, we can identify interesting cases illustrating the benefit of the RNN system. One compelling case is the difference between *Do you drink [alcohol]?*, *Do you drink coffee?*, and *Do you drink enough fluid?* (classes 85, 86, and 87 in development data). The *Do you drink?* class is very frequent, while the other two are in the least frequent quintile. Since *drink* by itself implies alcohol, the trigram *do you drink* is highly predictive of the alcohol class, and the CNN almost always errs on the other classes.

The RNN, on the other hand, handles this distinction quite well. In all cases, *drink* is attended by multiple heads (Figure 2), but across the set most of the heads are focused on representing the verb itself, while the magenta and tan representations (third and last row, respectively) are representing the object of the drinking. In the absence of an object, the object-focused head lands on the verb itself, and learns the implicit meaning of alcohol from the supervision.

We confirm that this behavior persists in the full model by performing agglomerative clustering on the full head representation in the test RNN. We see that the head that attends most strongly to *water* and *coffee* also often represents *alcohol* and *drink* in the same cluster. Meanwhile, other heads attend to the verbal meaning of *drink*, and encouragingly, these representations cluster nearby to similar consumption verbs such as *use* in the context of illegal drugs (Stiff, 2020). This may be expected due to the pretrained word vectors, but we also observe clusterings of apparently unrelated words like *take*

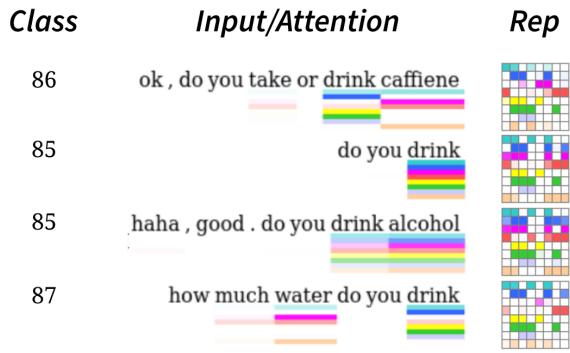


Figure 2: Example inputs with bottleneck attention head representations. The colored underlines show the foci of the attention heads, with opacity reflecting attention weights. The activation patterns in the correspondingly-colored rows of the grid representations reflect *how* the attended tokens are represented by each head. Note that heads consistently attending to “drink” (e.g. yellow and green) have similar representations across classes, while heads attending to the object of drinking (e.g. magenta and tan) have distinct representations for each class; further, the object-focused heads accept the verb as a stand-in for its implicit object when alcohol is not explicitly mentioned.

and *on*, which are similarly predictive of questions about prescribed medication (e.g. “Are you *on* any prescriptions?”), but which word senses are unlikely to converge representationally from pretraining on a general domain corpus. We take this as evidence of the BiGRU’s ability to disambiguate word senses based on context, especially since we occasionally observe the same word types in different clusters within the same head. Finally, we observe some very broad concepts being captured by some attention heads that generalize across many classes, such as the notion of temporal existential quantifiers (*ever*, *before*, *experienced*).

6 Conclusion

In some sense, our analysis is unsurprising. Words having the same input representations should cluster together in model-internal representations, and members of the same class should similarly cluster. However, we have shown evidence that the self-attentive RNN does some amount of word sense disambiguation that generalizes across classes, and this behavior is driven only by semantic classification. From a human perspective, it makes sense that learning the most generalizable representation should be effective, but it’s not clear that a model would need to learn those generalizations in order to perform the classification task. Clearly it ben-

efits from doing so, so it seems the multi-headed self-attention at least allows for learning these generalizable concepts and the corresponding better optimum.

There are some interesting questions and open issues that should be addressed with future work. Additional experiments should do more to control for parameter counts; these should be matched for comparisons of the Bottleneck RNN to the full Self-attentive RNN, to more robustly characterize the effects of the additional nonlinearity in the bottleneck model. The Bottleneck representations also seem to reflect something like rudimentary “concepts,” insofar as similar semantics often cluster together in the representation space. This raises the intriguing possibility that “metacognitive” processes could improve performance, for example with deductive or abductive inferences about relationships between representations across attention heads.

Overall, our analysis supports the claim that representations learned in frequent classes are transferring to, and improving performance on, rare classes, and further supports the value of a data set with a large number of subtly distinct classes.

References

- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.
- Douglas Danforth, A. Price, K. Maicher, D. Post, B. Liston, D. Clinchot, C. Ledford, D. Way, and H. Cronau. 2013. Can virtual standardized patients be used to assess communication skills in medical students. In *Proceedings of the 17th Annual IAMSE Meeting, St. Andrews, Scotland*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

- Evan Jaffe, Michael White, William Schuler, Eric Fosler-Lussier, Alex Rosenfeld, and Douglas Danforth. 2015. Interpreting questions with a log-linear ranking model in a virtual patient dialogue system. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 86–96.
- Lifeng Jin, David King, Amad Hussein, Michael White, and Douglas Danforth. 2018. Using paraphrasing and memory-augmented models to combat data sparsity in question interpretation with a virtual patient dialogue system. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 13–23.
- Lifeng Jin, Michael White, Evan Jaffe, Laura Zimmerman, and Douglas Danforth. 2017. Combining cnns and pattern matching for question interpretation in a virtual patient dialogue system. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–21.
- Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1746–1751.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.
- Abhijit Mahabal, Jason Baldridge, Burcu Karagol Ayan, Vincent Perot, and Dan Roth. 2019. Text classification with few examples using controlled generalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3158–3167.
- Vinod Nair and Geoffrey E Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, 3, pages 807–814.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Adam Stiff. 2020. *Mitigation of Data Scarcity Issues for Semantic Classification in a Virtual Patient Dialogue Agent*. Ph.D. thesis, The Ohio State University, Columbus, Ohio, USA.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3261–3275.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.
- Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory Networks. In *ICLR*, pages 1–15.
- Bruce Wilcox. 2019. Chatscript. [Online; accessed 23-July-2019].
- Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR*.

A Model Training

A.1 CNN Baseline

During cross-validation, we take ten percent of the data as test, and another ten percent for validation. We train on the remainder using Adadelta ([Zeiler, 2012](#)) for up to 25 epochs, and the model that produces the best validation accuracy is tested on the test set. Each training fold is augmented with the canonical questions for each class, so that no class is entirely unseen at test time. At test time, we take ten percent of the training data as a validation set, train on the other 90 percent, and use the same method of choosing which model to test. We use batch sizes of 50, and use GloVe.42B ([Pennington et al., 2014](#)) pretrained word vectors as input. We follow ([Jin et al., 2017](#)) for initialization and optimization parameters.

A.2 BERT Fine-tuning

We follow the recommended procedure for fine-tuning BERT to our task. We used the uncased base pretrained BERT model as input to a dense layer followed by a softmax for classification. All parameters were tuned jointly. The grid search optimized hyperparameters were a max sequence length of 16, a batch size of 2, 10 training epochs, and a learning rate of 2e-5.

A.3 CNN with Static BERT Embeddings

We expect that BERT model may be over-parameterized and under-trained for our relatively small data set. Thus, we collect non-contextual representations for the words in our dataset from the pretrained model. We then feed these as input to the baseline CNN model instead of the GloVe vectors.

We collect these static BERT embeddings by running the training set through the BERT model, and taking the state of the first layer from the BERT model as the embedding of the correspond token. We then average these representations for each word type in the data set, and use that as the input wherever the word occurs. Note that since BERT is trained with positional embeddings instead of ordering, representations from this layer likely retain a lot of positional information, which could be an important source of noise in the averaged representations. Training the CNN is otherwise the same as in the baseline experiment.

A.4 CNN with Contextual BERT Embeddings

Finally among our experiments with BERT, we feed the fully contextual representations into the baseline CNN architecture. Here, we again take the representation extracted from bottom layer of the BERT model.

A.5 RNN Training

The RNN with self-attention is trained using the same fold splits and canonical query augmentation as the CNN baseline. Here we use the Adam optimizer ([Kingma and Ba, 2014](#)) with default parameters. We initialize layer weights uniformly at random in the range $[-0.1, 0.1]$, and tokenize inputs using default SpaCy tokenization ([Honnibal and Montani, 2017](#)). We use GloVe.42B vectors again, and batch sizes of 32. We train for 40 epochs with an initial learning rate of 0.001, take the best model, reinitialize an optimizer with learning rate of 2.5×10^{-4} , and train for another 20 epochs, taking the best model of all 60 epochs to test.

Filtering conversations by dialogue act labels for improving corpus-based convergence studies

Simone Fuscone^{1,2} and Benoit Favre² and Laurent Prévot^{1,3}

¹ Aix-Marseille Univ, CNRS, LPL, Aix-en-Provence, France

² Aix Marseille Univ, CNRS, LIS, Marseille, France

³ Institut Universitaire de France, Paris, France

Abstract

During an interaction the tendency of speakers to change their speech production to make it more similar to their interlocutor's speech is called *convergence*. Convergence had been studied due to its relevance for cognitive models of communication as well as for dialogue system adaptation to the user. Convergence effects have been established on controlled data sets while tracking its dynamics on generic corpora has provided positive but more contrasted outcomes. We propose to enrich large conversational corpora with dialogue acts information and to use these acts as filters to create subsets of homogeneous conversational activity. Those subsets allow a more precise comparison between speakers' speech variables. We compare convergence on acoustic variables (Energy, Pitch and Speech Rate) measured on raw data sets, with human and automatically data sets labelled with dialog acts type. We found that such filtering helps in observing convergence suggesting that future studies should consider such high level dialogue activity types and the related NLP techniques as important tools for analyzing conversational interpersonal dynamics.

1 Introduction

The way participants engaged in a conversation speak tends to vary depending on their interlocutor's speech. The tendency to co-adjust speaking styles in response to the partner speaking style is known as *convergence*. Convergence is presented in general and influential models of communication such as *accommodation theory* (Giles et al., 1991) or *interactive alignment* (Pickering and Garrod, 2004). This variation due to the other party has been studied for many levels of speech and language, for example in phonology (Street, 1984; Pardo, 2006) or in prosody (Levitin and Hirschberg, 2011; Truong and Heylen, 2012; Bonin

et al., 2013).

Our approach aims at deepening and generalizing the investigation of convergence and related effects in real-life corpora. Considered from the angle of speech and linguistic variables, an essential aspect of conversations is their extreme variability. This is due to a large extent to different conversational activities speakers can participate in. For instance, they can enter in a storytelling sequence in which one interlocutor starts to produce mostly back-channels (Yngve, 1970) while the main speaker develops lengthy monologues. This variability makes comparison of values across participants very problematic. We propose to create subsets of similar **dialogues acts (DA)** (e.g. 'statements' and 'back-channels', see Table 1 for an illustration), resulting in homogeneous data used as a proxy to characterize the conversational activity of a given turn. We intend to create subsets consisting of turns belonging to a specific function using current Dialogue Act tagging Techniques.

Our work concerns more specifically acoustic convergence. Our definition of *convergence* comes from several studies (Edlund et al., 2009; Truong and Heylen, 2012; Cohen Priva et al., 2017), and consists of comparing distance between speakers in different parts of a conversation (See Section 3). We do not claim it is the best measure to approach inter-personal dynamics (See (Priva and Sanker, 2019)) but it is an interesting way to assess convergence within a conversation and it allows to test whether our DA based approach can help this domain.

2 Related work

Convergence has been approached at different granularity levels and for a large range of variables. In terms of granularity, studies can be *Inter-conversation* comparisons or *Intra-conversation*

(focusing on the dynamics within conversations). Inter-conversation comparisons range from simple inter-speaker correlation studies (Edlund et al., 2009) or, when the data allows, comparison between speech values of a speaker and his conversational partners vs. a speaker and all other non-partner corpus participants (Cohen Priva et al., 2017). Intra-conversation studies vary a lot in terms of approaches ranging from "difference-in-difference" convergence (Edlund et al., 2009), (Truong and Heylen, 2012), (Cohen Priva and Sanker, 2018) approaches consisting of comparing distances between speakers in different intervals to finer grained *synchrony* methods typically using sliding windows to compare local speaker similarities (Truong and Heylen, 2012).

While a large body of carefully controlled experiments on lab speech provided results on convergence, the results on real corpora (from the studies listed in the previous paragraph) provide a more complex picture, with a relative fragility of the effects (Fuscone et al., 2018) and raised methodological comments (See (Truong and Heylen, 2012) and (Cohen Priva and Sanker, 2018)). More precisely, for *intra-conversation* studies, (Edlund et al., 2009) found that participants tend to be more similar (in terms of gaps and pauses duration) to their partners than chance would predict. However, the absence of significant results in comparing the inter-speaker distance in the first and second halves of the conversations makes the authors conclude that convergence cannot be captured with such an approach. (Truong and Heylen, 2012) conducted a similar experiment (on intensity and pitch) on English Map Task ((Anderson et al., 1991)) with partial positive results. The dynamic nature of the phenomenon as well as the social factors render such studies difficult to be performed. These two studies were grounded on conversational corpora that are sizeable but not huge (6 x 20 minutes for the (Edlund et al., 2009); and about 60 MapTasks dialogues for (Truong and Heylen, 2012)). (Cohen Priva and Sanker, 2018) used the much bigger Switchboard corpus but use only an inter-conversation approach.

Our hypothesis is that automatic *entrainment* and strategic *adaptation* are blending in to produce *convergence* and *synchrony* phenomena. Our hypothesis is that low-level variables (such as intensity) are more directly related to automatic entrainment, while higher-level variables (such as lexical or syntactic choices) are more prone to strategic adap-

tation. This could explain why more and firmer results seem to be obtained on low-level variables (Natale, 1975; Levitan, 2014).

To summarize, convergence dynamic can be difficult to track in real conversations. Our approach combines three ingredients that, to our best knowledge, were not yet brought together. First, we consider that a major reason for this difficulty comes from the heterogeneity of speech behaviors within the time-frame of a conversation. We propose to use DA to filter conversational activities from large corpora. Second, to account for *strategic adaptation* one must take precise care of speaker profiles. Our approach therefore focuses on relatively low level variables to avoid as much as possible the "adaptation" part of the interpersonal dynamics. Third, similarly to (Cohen Priva et al., 2017) our approach is based on a large conversational corpus with the intention of overcoming noise and effect small magnitude by increasing the amount of data considered.

3 Methodology

3.1 Convergence

Following (Edlund et al., 2009) and (Truong and Heylen, 2012) we divide each conversation into two halves and compare the distance between the average values of the target variables of each speaker. This provided us two values (first and second interval) for each variable and each conversation: $\Delta\bar{V}_i = |\bar{V}_{Ai} - \bar{V}_{Bi}|$, where $i = 1, 2$ refers respectively to the first and second interval, A and B indicate the speakers who take part in the conversation while V corresponds to **Energy (E)**, **Pitch (F0)** and **Speech rate (SR)**. Our aim is to test the hypothesis that convergence occurs during the interaction. We therefore computed the distance between both intervals, resulting in a distribution of these values in both intervals for the whole corpus. We then fitted a linear mixed regression model of this distribution to test if there is a significant difference across the intervals. Moreover, the sign of the estimate of the model provides us the direction of the evolution. We use the `lme4` library in R (Bates et al., 2014) to fit the models and provide t-values. The `lmerTest` package (Kuznetsova et al., 2014), which encapsulates `lme4`, was used to estimate degrees of freedom (*Satterthwaite approximation*) and calculate p-values. In the model, the $\Delta\bar{V}_i$ is the predicted value, the A and B identities as well as the topic of the conversation are set

as random intercepts. The model, in R notation, is $\Delta \bar{V}_i \sim t_i + (1 | topic) + (1 | speaker_A) + (1 | speaker_B)$.

3.2 Feature processing

E and F0 are computed from the audio files with *openSMILE* audio analysis tool (Eyben and Schuller, 2015) while SR is computed using time aligned transcripts.

Energy (E): One of the issues of telephonic conversation is the distance mouth-microphone that affects measured values of voice intensity can be different across speakers and even for the same speaker across conversations. So to reduce this effect we introduce a normalization factor by dividing each speaker E values by the average E produced by that speaker in the entire conversation. In addition, to reduce the environmental noise, we computed the average E using the temporal windows where the probability of voicing is above 0.65. Then we computed for each conversational unit (as provided by Switchboard transcripts) the average E.

Pitch (F0): We computed the average in each conversational unit for each speaker.

Speech Rate (SR): We used the approach proposed by Cohen-Priva (Cohen Priva et al., 2017) that defines SR for an utterance as the ratio between the actual duration of the utterance and its expected duration (computed by estimating every word duration into the whole corpus, for all speakers). Values above / below 1 correspond respectively to fast / slow speech compare to the average of the corpus. In order to make the measure SR more reliable we consider only utterances having more than 5 tokens.

4 Dialogue Act Filtering and Data Sets

Switchboard (SWBD) (Godfrey et al., 1992) is a corpus of telephonic conversations between randomly assigned speakers¹ of American English discussing a preassigned topic. The corpus consists of 2430 conversations (of an average duration of 6 minutes) for a total of 260 hours, involving 543 speakers. The corpus has audio, time aligned transcripts and a segmentation into *utterances*.

642 Switchboard conversations have been segmented and annotated for DA that we will call the

NXT data set (Calhoun et al., 2010). ² The DA-tagged set has been simplified to 42 tags but a few of them are dominating the distribution (Statement: 36%, Acknowledgment: 19%, Opinion: 13%), illustrated in Table 1. See (Stolcke et al., 1998) for details.

DA type	Example
Statement	"that was pretty heartrending for her"
Opinion	"money seems to be too big of an issue."
Backchannel	"Uh-huh."
Agree.	"you're right"

Table 1: Examples for the DA types used.

Automatically tagged data set We create a *turn tagger*, using 3 categories, corresponding to *Statement+Opinion* (STA+OPI), *Backchannel+Agreement* (BAC+AGR) and *Other* (OTH) which includes all the other DA. This grouping was obtained by first considering only the DA dominating the distribution. Then we manually checked many examples of each DA and figure out that although functionally different *statements* and *opinions* on the hand; and *backchannel* and *Agreement* those group were similar enough for our current purposes. The former has a *main speaker* nature while the later have a much more *listener* nature (see Table 1).

We used as train, development and test set the NXT Switchboard corpus that contains annotated DA for 642 conversations. Since the DA segmentation does not match the turn segmentation, we label each turn of the corpus by assigning the majority class, among the DA tags used in the turn. The resulting distribution is 52% STA+OPI, 25% BAC+AGR and 23% of OTH. The model we used is described in ((Auguste et al., 2018)) and inspired by the model of ((Yang et al., 2016)). It is a two levels hierarchical neural network (with learning rate = 0.001, batch size = 32, max length of each turn = 80, embeddings words dimension = 200). In the first level each turn is treated taking into account the words that form the turn while the second level is used to take into account the whole turn in the context of the conversation. Each level is a bidirectional Long Short Term (LSTM). We used 80% of Switchboard data as training set, 10% for development and 10% for the test set. The F1 score

²We use this version of DA as it contains alignment to the transcripts, contrarily to the SWDA bigger data set (Jurafsky et al., 1997).

¹Speakers therefore do not know each other.

of the DA tagger is 81% on the test set, the details for each category is reported in table 2. The F1 score of the class OTH is low compared to the other 2 classes.

Class	Precision	Recall	F1
Bc+Agr.	0.88	0.85	0.86
St.+Opi.	0.84	0.92	0.87
Oth.	0.62	0.49	0.55

Table 2: Prediction score of our turn tagger.

5 Results

Our question is whether we can observe more reliably interpersonal dynamics in raw, manually DA-tagged (small) or automatically DA-tagged (large) data sets. An underlying question is whether the noise introduced by the DA-tagging uncertainty and / or the data size reduction is compensated by the gain in homogeneity between the material that is compared.

5.1 DA-tagging contribution

We first report the results in the case of the whole data set without DA (SWBD) and manually DA-tagged (NXT). The results are summarized in Table 3.

	All	St.	Opi.	Bc.
SWBD (180h)	E - R	X	X	X
NXT	E - X (41h)	E - R (17h)	- - - (7h)	E - - (1h)

Table 3: Manual tagging results summary (E: Energy; P: Pitch ; R: Speech Rate; - : no significance; normal font : $p\text{-value} \leq 0.05$; **bold** : $p\text{-value} \leq 0.01$). See Table 5 in Appendix for details.

When differences are significant, it is always in the direction of reduction of the distance (See Appendix for the details). We observe that concerning *Statement*, with less than 10% of the original data, the method allows one to observe the same effect as in the whole Switchboard (and reaches a higher level of significance for SR). The *Statement* subset shows convergence for E and SR. *Statement*-filter seems to homogenize the data set by filtering out particular back-channels and strong disfluencies (type *abandoned*). This helps observing the effect for SR. Contrarily, the wide variety of *statements* in terms of utterance duration could be an issue

for F0 since contours and physiological-related decreasing slope could result in a lot of noise for this variable. There are no positive results on *Opinion* maybe due to larger variability or consistency in this label. *Back-channel* although keeps the effect on the E but, due the nature of this speech act, SR is not relevant. F0 doesn't show any significant results. This probably can be explained considering that F0 is a more complex variable and the average approach is not able to capture more subtle characteristics of F0 (Reichel et al., 2018).

5.2 Automatic Tagging results

As explained above, in the experiment on automating tagging we merged the most similar frequent DA. The automatically tagged corpus preserved the results from the raw data sets. Similarly for the manual version, automatic tags filtering helped reaching better significance for SR on *Statement+Opinion* utterances as summed-up in Table 4. Back-channels were excluded from the SR experiment since our measure of SR isn't reliable on such short utterances.

	All	St. + Opi.	Bc. +Agr.
SWBD	E-R	X	X
Auto	E - R	E - R	E - X

Table 4: Automatic tagging results summary (E: Energy; P: Pitch ; R: Speech Rate; - : no significance; normal font : $p\text{-value} \leq 0.05$; **bold** : $p\text{-value} \leq 0.01$). See Table 6 in Appendix for details.

6 Discussion

We scrutinized *convergence* during the course of a conversation and in a real world setting (Switchboard corpus). The positive results in our experiments complement the picture provided by the literature by showing that convergence effects do happen in the time course of conversation of generic corpus. Moreover, we open up the possibility of a range of new studies taking advantage on arbitrary large corpora partially controlled *a posteriori* thanks to automatic dialogue act tagging.

Acknowledgments

This project received funding from the EU Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No713750. Also, it has been carried out with the financial support of the Regional Council of Provence-Alpes-Côte d'Azur and with the financial support of A*MIDEX (ANR-11-IDEX-0001-02) and ILCB (ANR-16-CONV-0002).

References

- Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. 1991. The hcrc map task corpus. *Language and speech*, 34(4):351–366.
- Jeremy Auguste, Robin Perrotin, and Alexis Nasr. 2018. Annotation en actes de dialogue pour les conversations d’assistance en ligne. In *Actes de la conférence Traitement Automatique de la Langue Naturelle, TALN 2018*, page 577.
- Douglas Bates, Martin Maechler, Ben Bolker, Steven Walker, et al. 2014. lme4: Linear mixed-effects models using eigen and s4. *R package version*, 1(7):1–23.
- Francesca Bonin, Céline De Looze, Sucheta Ghosh, Emer Gilmartin, Carl Vogel, Anna Polychroniou, Hugues Salamin, Alessandro Vinciarelli, and Nick Campbell. 2013. Investigating fine temporal dynamics of prosodic and lexical accommodation. In *Proceedings of 14th Annual Conference of the International Speech Communication Association*, Lyon, France.
- Sasha Calhoun, Jean Carletta, Jason M Brenier, Neil Mayo, Dan Jurafsky, Mark Steedman, and David Beaver. 2010. The nxt-format switchboard corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language resources and evaluation*, 44(4):387–419.
- U Cohen Priva and C Sanker. 2018. Distinct behaviors in convergence across measures. In *Proceedings of the 40th annual conference of the cognitive science society. Austin, TX*.
- Uriel Cohen Priva, Lee Edelist, and Emily Gleason. 2017. Converging to the baseline: Corpus evidence for convergence in speech rate to interlocutor’s baseline. *The Journal of the Acoustical Society of America*, 141(5):2989–2996.
- Jens Edlund, Mattias Heldner, and Julia Hirschberg. 2009. Pause and gap length in face-to-face interaction. In *Tenth Annual Conference of the International Speech Communication Association*.
- Florian Eyben and Björn Schuller. 2015. opensmile: the munich open-source large-scale multimedia feature extractor. *ACM SIGMultimedia Records*, 6(4):4–13.
- Simone Fuscone, Benoit Favre, and Laurent Prevot. 2018. Replicating speech rate convergence experiments on the switchboard corpus. In *Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language*.
- H. Giles, N. Coupland, and J. Coupland. 1991. Accommodation theory: Communication, context, and consequence. In *Contexts of accommodation: Developments in applied sociolinguistics*, Studies in emotion and social interaction, pages 1–68. Cambridge University Press.
- John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 517–520. IEEE.
- D Jurafsky, E Shriberg, and D Biasca. 1997. Switchboard dialog act corpus. *International Computer Science Inst. Berkeley CA, Tech. Rep.*
- A Kuznetsova, P Bruun Brockhoff, and R Haubo Bojesen Christensen. 2014. lmertest: tests for random and fixed effects for linear mixed effects models. See <https://CRAN.R-project.org/package=lmertest>.
- Rivka Levitan. 2014. *Acoustic-prosodic entrainment in human-human and human-computer dialogue*. Ph.D. thesis, Columbia University.
- Rivka Levitan and Julia Hirschberg. 2011. Measuring acoustic-prosodic entrainment with respect to multiple levels and dimensions. In *Proceedings of Interspeech 2011*.
- Michael Natale. 1975. Convergence of mean vocal intensity in dyadic communication as a function of social desirability. *Journal of Personality and Social Psychology*, 32(5):790.
- Jennifer S Pardo. 2006. On phonetic convergence during conversational interaction. *The Journal of the Acoustical Society of America*, 119(4):2382–2393.
- Martin J Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and brain sciences*, 27(2):169–190.
- Uriel Cohen Priva and Chelsea Sanker. 2019. Limitations of difference-in-difference for measuring convergence. *Laboratory Phonology: Journal of the Association for Laboratory Phonology*, 10(1).
- Uwe D Reichel, Katalin Mády, and Jennifer Cole. 2018. Prosodic entrainment in dialog acts. *arXiv preprint arXiv:1810.12646*.
- Andreas Stolcke, Elizabeth Shriberg, Rebecca Bates, Noah Coccaro, Daniel Jurafsky, Rachel Martin, Marie Meteer, Klaus Ries, Paul Taylor, Carol Van Ess-Dykema, et al. 1998. Dialog act modeling for conversational speech. In *AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, pages 98–105.
- Richard L. Street. 1984. Speech convergence and speech evaluation in fact-finding interviews. *Human Communication Research*, 11(2):139–169.
- Khiet P Truong and Dirk Heylen. 2012. Measuring prosodic alignment in cooperative task-based conversations. In *Thirteenth Annual Conference of the International Speech Communication Association*.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.
- Victor H Yngve. 1970. On getting a word in edgewise. In *Chicago Linguistics Society, 6th Meeting*, pages 567–578.

A Appendices

SWBD	<i>Entire Corpus (180 hours)</i>		
Feature	Estimate	std	p-values
E-Mean	-0.063	0.012	7×10^{-7}
F0-Mean	-0.044	0.021	0.490
SR-Mean	-0.049	0.024	0.046
NXT	<i>Whole DA-tagged subset (41 Hours)</i>		
Feature	Estimate	std	p-values
E-Mean	-0.054	0.021	0.026
F0-Mean	-0.057	0.040	0.158
SR-Mean	-0.106	0.047	0.026
NXT	<i>Backchannel Tag Subset (1 Hour)</i>		
Feature	Estimate	std	p-values
E-Mean	-0.082	0.041	0.045
F0-Mean	0.043	0.022	0.491
NXT	<i>Statement Tag Subset (17 Hours)</i>		
Feature	Estimate	std	p-values
E-Mean	-0.071	0.023	0.032
F0-Mean	-0.025	0.038	0.653
SR-Mean	-0.123	0.049	0.012
NXT	<i>Opinion Tag Subset (7 Hours)</i>		
Feature	Estimate	std	p-values
E-Mean	-0.061	0.033	0.627
F0-Mean	-0.032	0.053	0.552
SR-Mean	-0.096	0.061	0.115

Table 5: Parameters our linear model for Energy, Pitch and Speech Rate for the raw corpus and for the manually tagged corpus. Speech rate was not considered for back-channels.

Auto	<i>Energy</i>		
CLASS	Estimate	std	p-values
STA+OPI	-0.055	0.011	$4 \cdot 10^{-6}$
BAC+AGR	-0.079	0.028	0.006
Auto	<i>Pitch</i>		
CLASS	Estimate	std	p-values
STA+OPI	-0.035	0.038	0.353
BAC+AGR	0.053	0.028	0.192
Auto	<i>Speech rate</i>		
CLASS	Estimate	std	p-values
STA+OPI	-0.075	0.021	0.008

Table 6: Parameters of our linear model for Energy, Pitch and Speech Rate for the corpus automatically tagged. Speech Rate was not considered for back-channels.

Nontrivial Lexical Convergence in a Geography-Themed Game

Amanda Bergqvist
Uppsala University
Uppsala, Sweden
amanda.bergqvist@nordiska.uu.se

Ramesh Manuvinakurike
and Deepthi Karkada
Intel Corp
United States
first.last@intel.com

Maike Paetzel
Uppsala University
Uppsala, Sweden
maiike.paetzel@it.uu.se

Abstract

The present study aims to examine the prevalent notion that people entrain to the vocabulary of a dialogue system. Although previous research shows that people will replace their choice of words with simple substitutes, studies using more challenging substitutions are sparse. In this paper, we investigate whether people adapt their speech to the vocabulary of a dialogue system when the system’s suggested words are not direct synonyms. 32 participants played a geography-themed game with a remote-controlled agent and were primed by referencing strategies (rather than individual terms) introduced in follow-up questions. Our results suggest that context-appropriate substitutes support convergence and that the convergence has a lasting effect within a dialogue session if the system’s wording is more consistent with the norms of the domain than the original wording of the speaker.

1 Introduction

The human habit of mirroring other’s choices of words could potentially provide a neat shortcut in the challenging task of building dialogue systems capable of understanding human language. Simply put, the dialogue system could nudge speakers to use words that are in its vocabulary by itself using those words in its output speech. The adaptation, known as *lexical entrainment* (mutual alignment) or *lexical convergence* (one-way adaptation) (Brennan, 1996; Beňuš, 2014), does not only apply to human–human interaction, but extends to human–computer interaction (Gustafson et al., 1997), as well as human–robot interaction (Iio et al., 2009). While natural languages offer innumerable ways of expressing the same idea (Furnas et al., 1987), a strategically designed system vocabulary could thus narrow down the range of words used by a human when speaking with an artificial partner.

In previous work, however, lexical convergence to a dialogue system has mostly been assessed in simple tasks, and the words suggested by the computer were close synonyms to the ones that the participant originally used. For humans, it might not make that much of a difference if a ticket is booked by saying “I’d like to go to” or “I’d like to travel to” (Gustafson et al., 1997). Results from Parent and Eskenazi (2010)’s study on a bus information system suggest that words that are frequent in day-to-day speech get entrained more often than less frequent “unnatural or harder” words. So, what if the substitutes proposed by the computer require more thought from the human than their initial phrasing, or do not come naturally to them?

In this paper, we aim to examine to what extent people imitate a dialogue system when the substitutions it suggests are nontrivial. We conducted an experiment using a cooperative two-player game in which people are asked to describe the location of countries on the world map. We hypothesized that human speech converges when the substitution requires minimal effort (changing between using *next to* and *borders*), but that convergence to cognitively straining substitutions (changing between *egocentric* and *cardinal* descriptions) is suppressed.

2 The RDG-Map Domain

We tested the lexical convergence in the context of a dialogue-based collaborative two-player game between a human and an unembodied female agent called Nellie (Paetzel and Manuvinakurike, 2019). The goal of the game is to locate as many countries as possible on a world map within the game time of 10 minutes. The human plays the role of the *Director* who receives target countries (cf. Figure 1) that s/he needs to verbally describe to the agent in the role of the *Matcher*. The targets were a predefined set of countries in a fixed or-

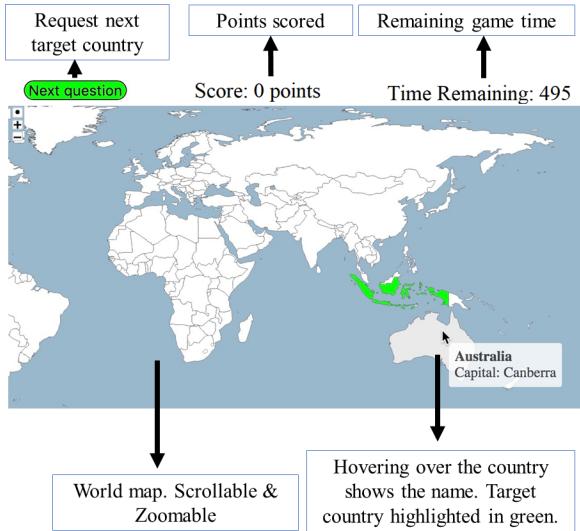


Figure 1: Game interface, the director's view. The current target is Indonesia (in green). The director is hovering the mouse over Australia (gray) to see its name.

der and were selected to evoke spatial directions. While participants believed they were playing with an autonomous agent, Nellie was, in fact, remote-controlled by a researcher.

Each game was divided into three stages: In the *baseline stage* (targets 1st to 4th), the agent did not mention any directions and the operator registered the natural word choices of the interlocutor (*borders* or *next to* resp. *cardinal* or *egocentric*). In the *priming stage* (targets 5th to 8th), the agent made use of an opposing set of expressions in the form of follow-up questions. Below is a minimal example in which the agent tries to prime the speaker using cardinal directions:

EXAMPLE 1 - PRIMING:

HUMAN: Austria is directly above Italy.

AGENT: Is it to the west of Hungary?

HUMAN: West of Hungary, yes.

AGENT: Got it.

In the *post-priming stage*, the agent returned to its original speech pattern. This stage could later be used to understand whether participants continued to use the vocabulary suggested by the agent in the second stage, or whether they fell back to their original lexical choices. A longer dialogue excerpt is shown in Figure 2.

3 Substitute Words

Two main strategies can be used to make spatial references on a map: describing *general relations*

between two countries and giving *directional descriptions*. General relations describe which countries border a certain country, but not which specific border they share. In this context, we identified “*A borders B*” and “*A is next to B*” to be simple substitutes that are interchangeable. Directional descriptions can be subdivided into *egocentric* (left, right, above, below) and *cardinal directions* (north, south, east, west). While bordering will always imply being “*next to*”, the cardinal direction corresponding to, e.g., left, depends on the position in a global reference frame. Swapping between egocentric and cardinal directions is thus not a simple matter of one-to-one translation, but involves changing strategy and can be considered more challenging than changing from “*borders*” to “*next to*”.

In contrast to most previous studies, we induced a swap of *referencing strategy* rather than a swap of *referencing terms*. In a study by Iio et al. (2009), participants adapt to the semantic framework that the system uses, not just individual terms, and Bell and Gustafson (2007) report a similar tendency in children playing a speech-enabled game. In our study, stimuli for *north* and *east* were thus expected to make players swap to *south* and *west* as well.

Previous studies mainly primed by swapping specific terms. When provoking a swap of terms, there are two options: the correction can be either embedded or exposed (Jefferson, 1987).

EXAMPLE 2 - EMBEDDED CORRECTION:

HUMAN: Austria is directly above Italy.

AGENT: I have selected the country north of Italy, got it.

EXAMPLE 3 - EXPOSED CORRECTION:

HUMAN: Austria is directly above Italy.

AGENT: By above, do you mean north?

Priming for a swap of referencing strategy allows for a third option: embedding members of the substitute referencing strategy without touching on the specific word used by the person. In Example 1, the agent does not mention the cardinal equivalent to *above*. Instead, it hints at its preference for cardinal directions by simply using them in its requests for further information. This makes for a smoother flow, as the conversation is actually progressing with respect to the goal of the game. In this study, primes for a referencing strategy were thus embedded in clarification requests.

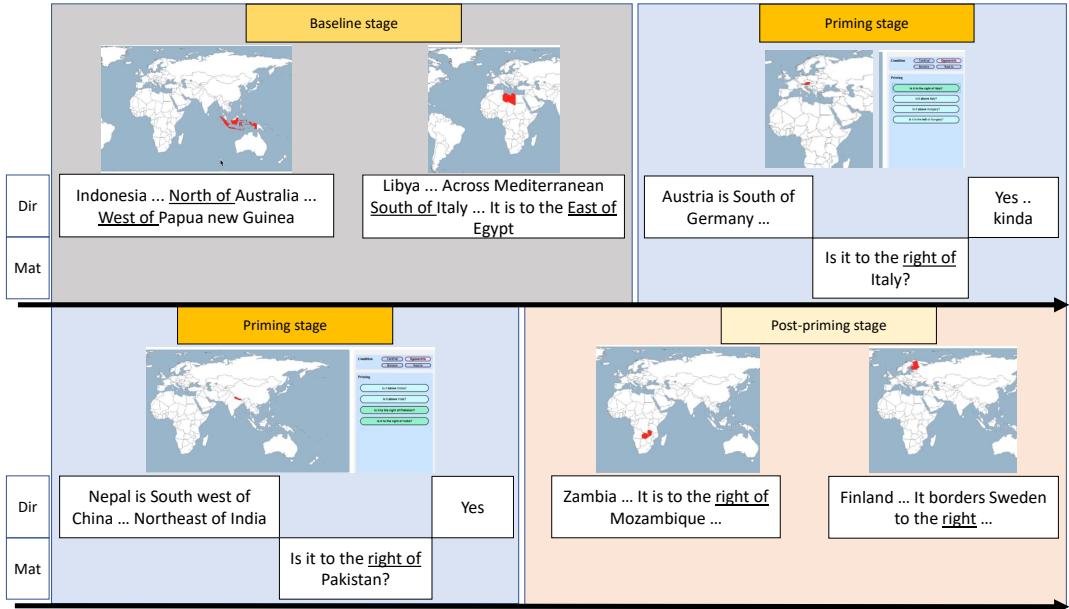


Figure 2: Excerpts from a sample conversation between a human Director (Dir) and agent Matcher (Mat) playing the game. In the baseline stage, the director uses cardinal descriptions. The director then faces questions from the matcher using egocentric descriptions. In the post-priming stage, the user converges to egocentric descriptions.

4 Experimental Design and Procedure

We conducted a study with a between-subject design in which the experiment group was subject to the more challenging swap of words between the directional relations, while the control group was subject to the simple swap between the words marking general relations. Except for the stimuli words, the setup was identical between conditions. In alignment with previous work, participants in the control condition were predicted to pick up the agent’s lexical choices, while those in the experiment group were predicted to converge less.

Participants first rated themselves in comparison to the average person in skills and pastimes involving navigation and travelling. In order to assess whether participants had a preference for egocentric or cardinal directions, they were also asked to fill in the revised Lawton’s Wayfinding scale (Lawton and Kállai, 2002). Before entering the game, players were randomly assigned to the experiment or the control group. The groupings within the experiment and the control group were based on the preference as determined in the baseline phase of the game, and players were assigned the condition opposite to their preference.

Participants were a convenience sample of 32 adult American native English speakers (age: $M = 34$, $SD = 9.38$; 45% female) who had never played the game before. All participants were

recruited on Amazon Mechanical Turk and paid \$3 upon completion of the experiment. In the experimental group ($N = 17$), 10 participants were exposed to cardinal directions and 7 to egocentric directions. In the control group ($N = 15$), 9 participants were exposed to “borders” and 6 to “next to”. Participants rated themselves as being averagely experienced with reading maps and using a GPS, but less experienced than the average person in using a compass. On average, participants had a higher route strategy score ($M = 3.45$, $SD = 0.6$) than orientation strategy score ($M = 2.51$, $SD = 0.82$). Since egocentric directions are related to the route strategy, this shows that participants are overall more accustomed to using egocentric descriptions in their daily life.

Automatically generated speech-to-text transcripts of the dialogues were manually corrected. They were then parsed, and occurrences of keywords were automatically counted. In addition, transcripts were manually annotated for the usage of descriptive strategy and false descriptions were flagged by the annotators (Paetzel et al., 2020).

5 Results

In both the experimental and the control condition, participants’ frequency of using the primed words increased during the priming stage of the experiment (cf. Table 1 and Figure 3). We performed a

		# words (frequency in %)				
Condition	Stage	cardinal	egocentric	borders	next to	total # words
cardinal	baseline	18 (15.25%)	108 (79.89%)	1 (0.63%)	4 (4.24%)	131
	priming	65 (52.81%)	73 (42.42%)	2 (1.05%)	7 (3.72%)	147
	post-priming	160 (65.4%)	116 (32.87%)	1 (0.26%)	5 (1.48%)	282
egocentric	baseline	34 (66.52%)	14 (18.4%)	7 (9.69%)	3 (5.39%)	58
	priming	41 (38.75%)	59 (53.47%)	8 (6.94%)	1 (0.84%)	109
	post-priming	98 (52.82%)	77 (40.11%)	8 (3.97%)	6 (3.1%)	189
borders	baseline	31 (37.87%)	50 (56.06%)	0 (0.0%)	5 (6.07%)	86
	priming	38 (38.31%)	29 (25.7%)	30 (30.9%)	4 (5.09%)	101
	post-priming	94 (37.98%)	79 (32.62%)	56 (26.26%)	9 (3.14%)	238
next to	baseline	29 (43.33%)	17 (35.6%)	3 (3.37%)	1 (1.04%)	50
	priming	27 (43.34%)	14 (21.34%)	5 (6.94%)	15 (28.38%)	61
	post-priming	80 (45.54%)	56 (33.84%)	26 (15.6%)	11 (5.03%)	173

Table 1: Instances of stimuli words (in absolute numbers and percentage) in player speech, grouped by condition and experiment stage. Cells representing the stimuli word(s) that a group was primed for are highlighted in green.

two-way ANOVA with the interaction stage (baseline, priming, post-priming) and the conditions (experiment: cardinal, egocentric; control: next to, borders) as independent variables.

The usage of the priming words “next to” and “borders” used for the control group was generally sparse. In the group primed for the word “borders”, the usage of the word increased significantly between the baseline and the priming stage, $p < .001$, and people continued using the word significantly more even in the post-priming stage, $p < .001$. In both the priming and the post-priming stage, the frequency of the word “borders” was significantly higher than in the same stage in all other three conditions. For the people being primed to use the words “next to”, we found a significant increase of the word usage during the priming phase, $p < .001$. However, the usage declines significantly after the priming stage, $p = .003$. During the priming stage, the usage of the word “next to” is significantly higher than during the priming stage in any other condition, while in the post-priming stage, it reaches the same level as in the other groups again.

In the experiment group, we found an increase of cardinal descriptions in the people primed to use the cardinal system. This increase is not significant between the baseline and the priming stage, $p = .15$, but becomes significant in the post-priming stage, $p = .009$. At the same time, the usage of egocentric descriptions in participants primed for the cardinal system declines between the baseline and the post-priming stage, $p = .012$. The group of people being primed to use the egocentric system slightly increase their usage of egocentric descriptions in the priming stage. This increase, however, is not significant, $p = .42$, and declines in the post-priming stage again.

Especially if a group converges towards the vocabulary of the dialogue system, it is relevant to examine whether communication suffers in other ways. If speakers comply with a computer by converging but commit errors because they are not accustomed to the proposed vocabulary, they may provide the computer with faulty information. However, in our system, we did not find a significant increase in the number of wrong descriptions given by participants in any condition. Similarly, we did not see an overall avoidance of giving directional descriptions in any of the conditions since the overall distribution between directional descriptions, size, or shape descriptions remained unchanged.

6 Discussion

As hypothesized, our results show that there was a statistically significant convergence of people’s vocabulary towards the vocabulary suggested by the agent in the control condition. This finding is in line with previous work and shows that *people are willing to adapt their vocabulary to an artificial agent even if substitute words are embedded in follow-up questions*, which is a weaker incentive for convergence compared to exposed corrections. Contrary to our expectations, however, we could also observe convergence in parts of the experimental group, specifically in the group exposed to cardinal directions. This finding is interesting as participants reported using egocentric directions more often in their daily lives, which would suggest they would be easier to adapt to than to the less common cardinal words.

A possible explanation for the higher convergence in the group naturally using egocentric descriptions lies in a core idea of lexical entrainment: conceptual pacts. According to Brennan and Clark

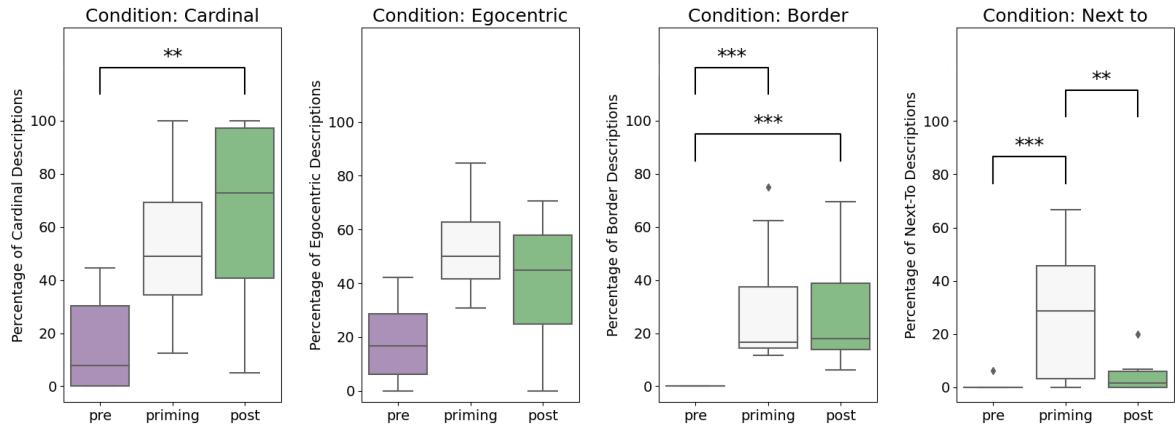


Figure 3: Visualization of the usage of the words participants were primed for in all four conditions. Significant differences are indicated with * ($p < .05$), ** ($p < .01$) and *** ($p < .001$)

(1996), entrainment is not merely a matter of repeating certain words, but rather a negotiation of a common reference system in a conversation. They suggest that in referring to an object with a certain word, the speaker is proposing a conceptualization of the object. In adopting the same word, the partner sends a message that they agree with the conceptualization. (S)he can also convey disagreement by rejecting the word and proposing a different one.

In the geography game, the high convergence of the participants who started with egocentric directions might reflect an acceptance of not just the cardinal words, but of the concept of referring to positions in a map by cardinal directions. Even though participants were, on average, more used to egocentric words, they obliged with the agent because they recognized the norm that links maps to cardinal directions. The lesser convergence of participants who started with cardinal directions may convey their disapproval of using egocentric directions in the given context. In accepting or rejecting the terms proposed by the computer, participants are thus not simply trying to or failing at facilitating the conversation. They are taking a stand as to whether the words proposed by the dialogue system make sense or not in the present context. Similarly, bordering is more commonly used to describe relations between countries and the convergence to the word “borders” was thus more lasting in the remainder of the conversation compared to the phrase “next to”. *Our findings suggest that people will replace their first choice of words if the alternative is more reasonable in a given context but will reject the alternative if they find it inferior to their initial choice.*

In our study, we did not measure whether people found the translation between the egocentric and the cardinal system to be more difficult than the swap between “borders” and “next to”, which reduces the conclusions we can draw when it comes to limits of lexical convergence due to cognitive load. In the future, we plan to conduct a larger experiment in which we measure the participant’s cognitive load explicitly. With a larger number of participants per group, we hope to be able to analyze further whether the differences in convergence in the experimental conditions are, in fact, the indicator of a significant trend.

7 Conclusion

The results of the present study provide further support for lexical convergence and the persuasiveness of lexical convergence in human–computer dialogue. They also indicate that convergence is related to the semantic appropriateness of the system vocabulary. More specifically, people are more likely to adopt substitute words that belong in the given context. In this particular study, the players of a geography-themed game rejected egocentric descriptions but adopted cardinal directions, likely since they were deemed better fitted for describing the location of a country. If high levels of lexical convergence are to be attained, we thus suggest that the vocabulary of a dialogue system needs to be harmonized with the domain at hand.

Acknowledgements

The authors thank K. Georgila and G. Castellano for their input and infrastructural support.

References

- Linda Bell and Joakim Gustafson. 2007. Children’s convergence in referring expressions to graphical objects in a speech-enabled computer game. pages 2209–2212.
- Štefan Beňuš. 2014. [Social aspects of entrainment in spoken interaction](#). *Cognitive Computation*, 6:802–813.
- Susan E. Brennan. 1996. Lexical entrainment in spontaneous dialog. In *International Symposium on Spoken Dialog*, pages 41–44.
- Susan E. Brennan and Herbert H. Clark. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22:1482–1493.
- G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. 1987. [The vocabulary problem in human-system communication](#). *Commun. ACM*, 30(11):964–971.
- J. Gustafson, A. Larsson, R. Carlson, and K. Hellman. 1997. How do system questions influence lexical choices in user answers. In *In Proc. Eurospeech '97*, pages 2275–2278.
- Takamasa Iio, Masahiro Shiomi, Kazuhiko Shinozawa, Takahiro Miyashita, Takaaki Akimoto, and Norihiro Hagita. 2009. Lexical entrainment in human-robot interaction: Can robots entrain human vocabulary? In *IROS*, pages 3727–3734.
- Gail Jefferson. 1987. On exposed and embedded correction in conversation. In *Talk and Social Organisation*, chapter 4, pages 86–100.
- Carol Lawton and János Kállai. 2002. [Gender differences in wayfinding strategies and anxiety about wayfinding: A cross-cultural comparison](#). *Sex Roles*, 47:389–401.
- Maike Paetzel, Deepthi Karkada, and Ramesh Manuvinakurike. 2020. Rdg-map: A multimodal corpus of pedagogical human-agent spoken interactions. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 593–602, Marseille, France. European Language Resources Association.
- Maike Paetzel and Ramesh Manuvinakurike. 2019. “Can you say more about the location?” The Development of a Pedagogical Reference Resolution Agent”. *Dialog for Good - Workshop on Speech and Language Technology Serving Society (DiGo)*.
- Gabriel Parent and Maxine Eskenazi. 2010. Lexical entrainment of real users in the let’s go spoken dialog system. pages 3018–3021.

A unifying framework for modeling acoustic/prosodic entrainment: definition and evaluation on two large corpora

Ramiro H. Gálvez^{1,2}, Lara Gauder^{1,2}, Jordi Luque³, Agustín Gravano^{1,2}

¹ Departamento de Computación, FCEyN, Universidad de Buenos Aires, Argentina

² Instituto de Ciencias de la Computación, CONICET-UBA, Buenos Aires, Argentina

³ Telefonica Research, Spain

{rgalvez, mgauder, gravano}@dc.uba.ar, jordi.luqueserrano@telefonica.com

Abstract

Acoustic/prosodic (a/p) entrainment has been associated with multiple positive social aspects of human-human conversations. However, research on its effects is still preliminary, first because how to model it is far from standardized, and second because most of the reported findings rely on small corpora or on corpora collected in experimental setups. The present article has a twofold purpose: 1) it proposes a unifying statistical framework for modeling a/p entrainment, and 2) it tests on two large corpora of spontaneous telephone interactions whether three metrics derived from this framework predict positive social aspects of the conversations. The corpora differ in their spoken language, domain, and positive social outcome attached. To our knowledge, this is the first article studying relations between a/p entrainment and positive social outcomes in such large corpora of spontaneous dialog. Our results suggest that our metrics effectively predict, up to some extent, positive social aspects of conversations, which not only validates the methodology, but also provides further insights into the elusive topic of entrainment in human-human conversation.

1 Introduction

A phenomenon that has been repeatedly documented in human-human conversations is the tendency of partners to become more similar to each other in the way they speak. This behavior, known in the literature as *entrainment*, has been shown to occur along several dimensions during human-human interaction (see Pardo, 2006; Brennan and Clark, 1996; Reitter et al., 2011; Levitan et al., 2015; Gravano et al., 2015; Fandrianto and Eskenazi, 2012, inter-alia), being one of these dimensions the behavior of *acoustic-prosodic* (a/p) features (see, for example, Ward and Litman, 2007; Levitan and Hirschberg, 2011).

A/p entrainment has been associated with multiple social aspects of human-human conversations,

such as the degree of success in completing tasks (Nenkova et al., 2008; Reitter and Moore, 2014), the perception of competence and social attractiveness (Street, 1984; Levitan et al., 2011; Štefan Beňuš et al., 2014; Michalsky and Schoormann, 2017; Schweitzer and Lewandowski, 2014), and the degree of speaker engagement (De Looze et al., 2014; Gravano et al., 2015). Nonetheless, empirical evidence also points toward these relations being quite complex. As an example, *disentrainment* (speakers actively adapting to become more dissimilar to each other) has also been associated with positive social aspects in conversations (see, for example, Healey et al., 2014; De Looze et al., 2014; Pérez et al., 2016).

In spite of these advances, research on the effects of a/p entrainment is still preliminary. In first place, because the way a/p entrainment in conversations is modeled is far from standardised. As an illustrative example, when estimating a/p entrainment metrics, some studies first approximate the evolution of each speaker's a/p features and then use these approximations to calculate a/p entrainment metrics (Gravano et al., 2015; De Looze et al., 2014; Koussidis et al., 2009; Pérez et al., 2016); others study the correspondence between adjacent inter-pausal units (IPUs) — defined as speech segments separated by a pause — from different speakers and derive metrics from it (Levitin and Hirschberg, 2011; Weise et al., 2019); and still other studies measure a/p features in different sections of speech for later comparing these values to compute a/p entrainment metrics (see, for example, Savino et al., 2016). Moreover, studies commonly differ in which metrics are analyzed. For this reason, a reliable, simple, general, and flexible framework able to unify the estimation of different types of entrainment metrics is needed. In second place, research is still preliminary because most of the reported findings rely on small corpora, or on corpora collected in experimental setups, making it hard to extrapolate their

conclusions to more general contexts. In this way, evidence is still needed on how a/p entrainment relates to social aspects of human-human conversation under different types of natural interactions.

The present article has a twofold purpose. First, it proposes a unifying approach for modeling a/p entrainment in conversations. The methodology is simple and flexible enough as to allow calculating adapted versions of several a/p entrainment metrics used in previous studies. Second, it evaluates three entrainment metrics derived from the proposed framework on two very different large corpora of spontaneous telephone interactions (the Switchboard corpus, in English, and a large collection of call-center conversations, in Spanish), testing whether these metrics predict positive social aspects of conversations. To our knowledge, this is the first article testing the relation between a/p entrainment and positive social outcomes in such large corpora of spontaneous natural dialog.

Overall, our results suggest that metrics derived from the proposed methodology effectively predict, up to some extent, positive social aspects in conversations, which not only validates the methodology, but provides further evidence suggesting that a/p entrainment relates to positive social aspects in human-human conversation under different types of natural settings. Additionally, insights on how a/p entrainment metrics relate to social outcomes predictions is provided.

The rest of the paper is structured as follows. Section 2 presents the proposed framework for modeling a/p entrainment. Section 3 details on how we empirically test for relations between metrics obtained using the proposed methodology and positive social aspects of conversations. Section 4 presents results from the empirical study. Section 5 provides discussion and concludes.

2 A unifying framework for modeling a/p entrainment

Here we present a methodology for modeling a/p entrainment. We divide the process in three steps: 1) extracting a/p features from IPUs, 2) estimating the speakers' a/p evolution functions, and 3) calculating a/p entrainment metrics from a/p evolution functions. The following sections describe each step.

2.1 Extracting a/p features from IPUs

First, for each speaker in a conversation (A and B for exposition) all of their IPUs are identified.¹ Then, for each IPU the value of their a/p features are extracted. In this study we used the Praat toolkit (Boersma and Weenink, 2019) to estimate the IPU's F0 maximum and mean; intensity max and mean; noise-to-harmonics ratio (NHR); and jitter and shimmer (computed over voiced frames only). We also extracted speech rate, measured in syllables per seconds.²

Figure 1 plots the F0 mean values for all IPUs in a sample Switchboard conversation. Each horizontal segment represents an IPU, graphically indicating its beginning and end times.

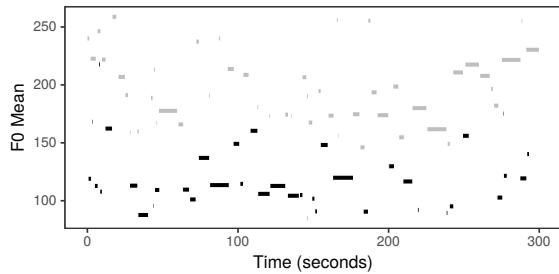


Figure 1: Estimated F0 mean values for all IPUs in a sample Switchboard conversation. *Note:* speaker A in gray, speaker B in black.

2.2 Estimating speakers' a/p evolution functions

Since speakers do not speak during the entirety of a conversation, the evolution of a/p features is undefined for several portions of a conversation. This stands as a challenge when modeling a/p entrainment. Previous research has dealt with this issue in multiple ways; for example, by pairing speakers adjacent IPUs (Levitin and Hirschberg, 2011; Weise et al., 2019) or by means of sliding windows (Kousidis et al., 2009; De Looze et al., 2014; Pérez et al., 2016). Instead, we propose filling these gaps by fitting a continuous function to approximate the evolution of a given a/p feature during a conversation. We do this by fitting a k -nearest neighbors

¹For the Switchboard corpus we used the MS-State transcripts (Deshmukh et al., 1998), where IPUs are annotated. For the call center conversations we used the output of an in-house automatic speech recognition system (Cartas et al., 2019), defining an IPU as a continuous segment of speech without a pause larger than 200 ms.

²Syllables were estimated using the *Pyphen* package (Pyphen, 2019).

(KNN) regression model to each speaker’s a/p feature values. Where, for each IPU, its x value is defined as its middle point in time (i.e., its start time plus its end time, divided by two). We refer to these estimated functions as f^A and f^B below.³ As we show below, adapting existing a/p entrainment metrics to take these functions as input is straightforward.

A few considerations should be made regarding the way to do these approximations. Due to the presence of outliers, before fitting these functions, all IPUs having an associated value more than three standard deviations away from the mean are dropped (the corresponding mean and standard deviation are measured at the conversation level for each speaker). Second, f^A is defined for the interval $[t_{min}^A, t_{max}^A]$, where t_{min}^A is the start time of A ’s first non-outlier IPU, and t_{max}^A is the end time of A ’s last non-outlier IPU (analogously for f^B). Third, we define the *common support* as all time values that go from $t^- = \max(t_{min}^A, t_{min}^B)$ up to $t^+ = \min(t_{max}^A, t_{max}^B)$ (i.e., all values of t where both functions are simultaneously defined). Fourth, approximations for speakers that do not have at least k non-outlier IPUs in a conversation are not computed (being k the number of neighbors used to estimate the functions).

Figure 2 plots the estimated approximation function for the IPUs plotted in Figure 1.

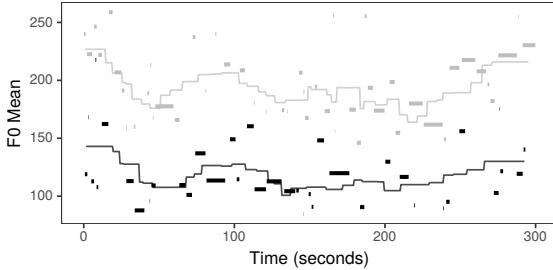


Figure 2: Estimated F0 mean evolution during a conversation. *Notes:* speaker A in gray, speaker B in black. The number of neighbors in the KNN regressions equals 7.

2.3 Calculating a/p entrainment metrics from a/p evolution functions

Existing a/p entrainment metrics can be easily adapted to take these functions as input. In this work we adapt and empirically test the three metric

³As we will see in Section 3.2, k (the number of neighbors) can be treated as a hyperparameter to be tuned during the model selection procedure.

types presented in Levitan and Hirschberg (Levit and Hirschberg, 2011): 1) *proximity* (a/p features having similar mean values across partners over the entire conversation), 2) *convergence* (a/p features increasing in similarity across partners over time), and 3) *synchrony* (speakers adjusting the values of their a/p features in accordance to those of their interlocutor).

2.3.1 Proximity

Proximity between f^A and f^B ($prox^{A,B}$) can be measured as the negated absolute difference of the mean values of f^A and f^B , that is:

$$-\left|\bar{f}^A - \bar{f}^B\right|$$

where, in general, \bar{g} stands for the mean value of function g over the common support, and is calculated as:⁴

$$\bar{g} = \frac{1}{t^+ - t^-} \int_{t^-}^{t^+} g(t) dt$$

Values of $prox^{A,B}$ close to zero indicate that f^A and f^B are on average close to each other, while values far from zero indicate that they are distant.

2.3.2 Convergence

Convergence between f^A and f^B ($conv^{A,B}$) can be measured as the Pearson correlation coefficient between $-\left|f^A - f^B\right|$ and t , which can be calculated as:

$$\frac{\int_{t^-}^{t^+} (D(t) - \bar{D}) \cdot (t - \bar{t}) dt}{\sqrt{\int_{t^-}^{t^+} (D(t) - \bar{D})^2 dt} \cdot \sqrt{\int_{t^-}^{t^+} (t - \bar{t})^2 dt}}$$

where $D(t)$ stands for $-\left|f^A(t) - f^B(t)\right|$. Positive/negative values of this metric indicate that f^A and f^B become closer to/further apart from each other as the conversation advances.

2.3.3 Synchrony

Synchrony between f^A and f^B ($sync^{A,B}$) can be measured as the Pearson correlation coefficient between f^A and f^B . Given that speakers are not expected to adapt to the other instantaneously, several studies consider a lag factor (δ) when calculating synchrony (see, for example, Kousidis et al., 2009; Pérez et al., 2016). In this study we also incorporate lags, which is a small departure from Levitan and Hirschberg (Levit and Hirschberg, 2011). Concretely, we calculated $sync^{A,B}$ as:

⁴In our empirical study, integrals are calculated using the Monte Carlo integration method.

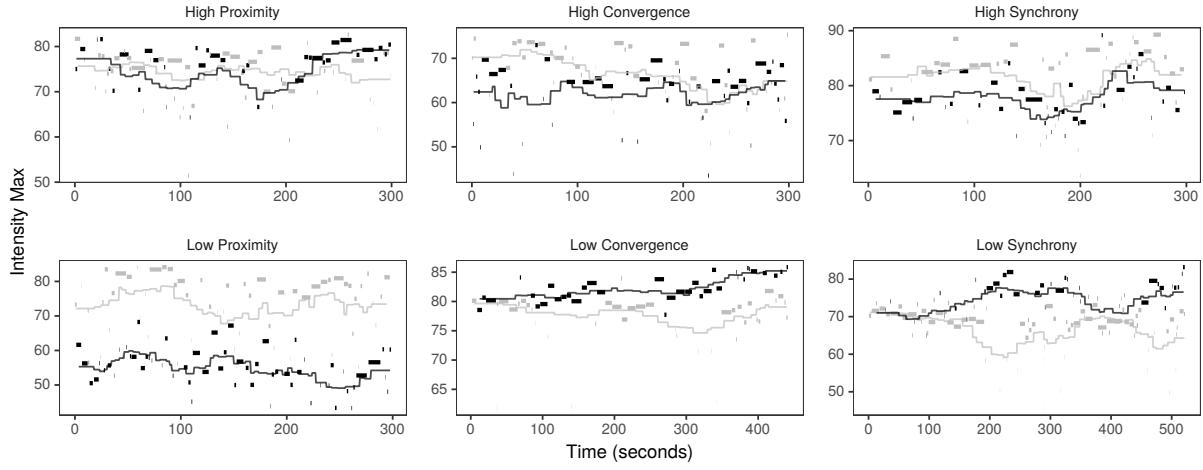


Figure 3: Sample conversations with different values of the estimated a/p entrainment metrics on intensity max.
Notes: speaker A in gray, speaker B in black. The number of neighbors in the KNN regressions equals 7.

$$\frac{\int_{t-}^{t+} (f^A(t + \delta) - \bar{f}^A) \cdot (f^B(t) - \bar{f}^B) dt}{\sqrt{\int_{t-}^{t+} (f^A(t + \delta) - \bar{f}^A)^2 dt} \cdot \sqrt{\int_{t-}^{t+} (f^B(t) - \bar{f}^B)^2 dt}}$$

where, in order to capture lags in synchrony between both functions, we test values of $\delta \in \{-15, -10, -5, 0, 5, 10, 15\}$ (being δ expressed in seconds). We take as the final value of $sync^{A,B}$ the one associated with δ resulting in the maximum value of $|sync^{A,B}|$.⁵ Positive values of $sync^{A,B}$ indicate that f^A and f^B evolve in synchrony with each other, while negative values indicate that they evolve in opposite directions (e.g., when one goes up the other goes down).

To illustrate the kind of behavior captured by these metrics, Figure 3 plots sample conversations with high and low values of these three a/p entrainment metrics calculated on the ‘intensity max’ feature.

3 Empirical study materials and methods

Next, we ran a series of machine learning experiments aimed at investigating whether the metrics derived from the proposed methodology have any predictive power over positive social aspects of conversations. This section describes the corpora and the methodology used.

3.1 Corpora

3.1.1 Switchboard corpus

The Switchboard Corpus (SWBD) (Godfrey et al., 1992) is a collection of 2,438 recordings of spon-

⁵Note that shifting a series slightly modifies the common support, something that should be taken into account.

taneous two-sided telephone conversations among 543 speakers (both female and male) from all areas of the United States. During collection, a robot operator system handled the calls, gave the caller appropriate recorded prompts, selected and dialed the callee, introduced one of about 70 topics for discussion (internally referred as IVIs), and recorded the whole speech from the two subjects into separate channels. Each conversation was annotated for degree of naturalness on Likert scales from 1 (very natural) to 5 (not natural at all).⁶ For this corpus, **perceived naturalness** is the target social outcome to predict.

After dropping from the analysis a few conversations for which naturalness scores were missing, we were left with a total of 2,426 conversations (average conversation length: 382.3 seconds; SD: 124.8 seconds). To make the analysis and results more interpretable (more on this in Section 3.2), we dichotomized the naturalness scores in the following way: We treated values 1 and 2 as *high* scores (which we set equal to 1 — 88.4% of all conversations) and values from 3 to 5 as *low* scores (which we set to 0 — 11.6% of all conversations).

In addition to the proposed metrics, our experiments included features referred to as *external features*, which are expected to be linked to the naturalness score, but are unrelated to the speakers’ a/p features. These features are used for building baseline models to compare against. For the case of SWBD these variables are: IVI indicator

⁶More details on naturalness annotations available at https://catalog.ldc.upenn.edu/docs/LDC97S62/swb1_manual.txt.

variables (indicating the IVI used as the conversation topic), conversation length (in seconds), dialect area indicator variables (indicating whether at least one speaker belonged to a given dialect area, and whether both speakers belonged to the same dialect area), three gender indicator variables (both-female-speakers, both-male-speakers, mixed-gender-speakers), and the absolute value of the age difference between the speakers (in years).

3.1.2 Call center corpus

The call center corpus (CCC) is a collection of 19,832 inbound call center conversations between clients and representatives of a telephone company (for further details see [Llimona et al., 2015](#); [Luque et al., 2017](#)) (average conversation length: 551.7 seconds; SD: 432.9 seconds). It was collected throughout one month and comprises a huge variety of interactions. All conversations are in Latin American Spanish. At the end of each call, the customer was called back and gently asked to complete a brief service quality survey. Concretely, they had to indicate their overall satisfaction with respect to their previous call center call. To do so, they had to press from 1 (very dissatisfied) to 5 (very satisfied). For this corpus, **self-reported customer satisfaction** is the target social outcome to predict.

We again dichotomized the target variable in the following way: 4 and 5 were treated as *high* scores (which we set equal to 1 — 80.5% of all conversations), and values from 1 to 3 were treated as *low* scores (which we set to 0 — 19.5% of all conversations).

For anonymity reasons, the availability of external variables was more limited for CCC. Thus, only conversation length and the three gender indicator variables ([Llimona et al., 2015](#)) were included as external features.

3.2 Testing for associations between a/p entrainment and social outcomes

To test if the proposed entrainment metrics predict the outcomes, we ran a series of machine learning experiments. For each corpus we trained several XGBoost models ([Chen and Guestrin, 2016](#))⁷ using different feature sets, and evaluated their predictive performance. For example, one such model

⁷XGBoost is an open-source and efficient software implementation of the gradient boosting framework ([Friedman et al., 2001](#)). XGBoost has the additional advantage of dealing with missing values, which, in our analysis, were present both in the external features and the a/p entrainment metrics.

used only the synchrony metrics computed on the 8 a/p features described in Section 2.1 (F0 max, F0 mean, intensity max, intensity mean, NHR, jitter, shimmer, speech rate); other model considered all 24 entrainment metrics (8 a/p features \times 3 metric types); other model considered only external features; and so on.

As the evaluation metric, we used the area under the receiver operating characteristic curve (AUC) (see [Alpaydin, 2020](#)). AUC goes from 0 to 1, where an AUC value equal to 0.5 indicates an equal-than-chance performance, while larger values indicate that the learning model effectively predicts the outcomes, up to some extent. To obtain our *out-of-sample* performance estimates we ran 10-fold cross validation experiments (see [James et al., 2014](#)). We tuned the hyperparameters following a random search strategy ([Bergstra and Bengio, 2012](#)): For each value of $k \in \{3, 5, 7, 9\}$ (number of neighbors used in the functional approximations) we tested 60 randomly sampled combinations of seven XGBoost hyperparameters.⁸ The chosen hyperparameters are those for which the model had the higher cross validation performance.

3.2.1 Model interpretability

Comparing performance across models provides valuable information regarding feature importance. However, further valuable information can be obtained by interpreting the models' inner workings. To do so, several strategies have been proposed (see [Molnar, 2019](#)). In our analysis we made use of the Shapley additive explanations (SHAP) technique ([Lundberg and Lee, 2017](#)). SHAP values are calculated for each observation and predictive feature in the dataset used to train the model being analyzed. Concretely, a given SHAP value $\phi_{i,j}$ estimates, for observation i , how feature j contributes to push the model output (in logit scale) from its base output (being the base output equal to the average model output over the training dataset). In this way, SHAP values can be used to estimate feature importance for a given feature j by calculating $\sum_i |\phi_{i,j}|$. They can also characterize how the outputs diverge from the base output as feature j grows, by using SHAP feature dependence plots (that is, plotting $\phi_{i,j}$ against all observed values of feature j).⁹

⁸Number of trees; tree depth; step size shrinkage coefficient; minimum loss reduction required to make a further partition; minimum child weight; number of columns sampled in each tree; and number of observations sampled in each tree.

⁹It is important to stress that any pattern derived from the model interpretability analysis does not imply that a feature

4 Empirical study results

4.1 Predictive performance

We trained models on eight different sets of inputs: 1) only proximity metrics, 2) only convergence metrics, 3) only synchrony metrics, 4) all a/p entrainment metrics, 5) external features and proximity metrics, 6) external features and convergence metrics, 7) external features and synchrony metrics, and 8) external features and all a/p entrainment metrics. The rest of this section presents the performances obtained for each corpus.

4.1.1 Switchboard corpus results

Table 1 presents the estimated performance for each set of input features. The top panel presents results excluding external features, while the bottom one includes them. Within each panel, models are sorted in descending AUC order.

Input Features	AUC
Excluding external features	
Only synchrony	0.575
All a/p entrainment metrics	0.566
Only proximity	0.561
Only convergence	0.547
Including external features	
External and synchrony	0.641
External and convergence	0.631
External and all a/p entrainment metrics	0.630
Only external	0.627
External and proximity	0.624

Table 1: Switchboard corpus AUC results

Table 1 shows that the trained models are able to predict up to some extent perceived naturalness. In all cases the obtained results are higher than chance (i.e., $AUC > 0.5$). But not all features have the same predictive performance. Synchrony entrainment metrics obtain the best results. Training on just synchrony metrics results in an AUC of 0.575, while using only convergence or proximity metrics leads to lower AUC values. Training on all a/p entrainment metrics results in an AUC of 0.566, lower than the one obtained with synchrony metrics.

Training only on external features results in an AUC of 0.627, higher than all values presented in the top panel. However, adding synchrony metrics to the external features is the combination that leads to the best overall results.

has a causal relationship with the outcome. It merely indicates that a given feature causes **the model** to predict the outcome in a particular way (see Molnar, 2019).

4.1.2 Call center corpus results

Input Features	AUC
Excluding external features	
All a/p entrainment metrics	0.582
Only synchrony	0.560
Only convergence	0.556
Only proximity	0.548
Including external features	
External and all a/p entrainment metrics	0.582
External and proximity	0.568
External and synchrony	0.564
External and convergence	0.560
Only external	0.537

Table 2: Call center corpus AUC results

For CCC, Table 2 shows that the trained models are also able to predict up to some extent self-reported customer satisfaction. However, we observe that combining all 24 entrainment metrics leads to better results ($AUC = 0.582$) than including just the synchrony ones ($AUC = 0.560$).

In this case the external features have low predictive power when compared to the entrainment metrics. Adding the external features to the model considering all a/p entrainment metrics yields exactly the same results as the ones obtained by the model trained only on all a/p entrainment metrics.

4.2 Model interpretability results

Switchboard corpus	
Input Feature	Feature Importance
Both-female-speakers	100.0
Speech-rate-synchrony	45.7
Conversation-length	27.8
Intensity-mean-synchrony	19.1
Jitter-synchrony	18.8
F0-max-synchrony	16.4
Age-difference	16.0
Shimmer-synchrony	9.9
F0-mean-synchrony	9.1
Intensity-max-synchrony	6.3

Call center corpus	
Input Feature	Feature Importance
Speech-rate-proximity	100.0
Speech-rate-synchrony	62.3
Conversation-length	54.9
Intensity-max-convergence	47.3
Jitter-convergence	36.6
F0-mean-proximity	30.2
Jitter-proximity	26.5
Speech-rate-convergence	23.8
F0-mean-convergence	19.9
NHR-convergence	19.9

Table 3: Feature importance ranking. *Note:* values are scaled such that the score associated to the most important feature equals 100.

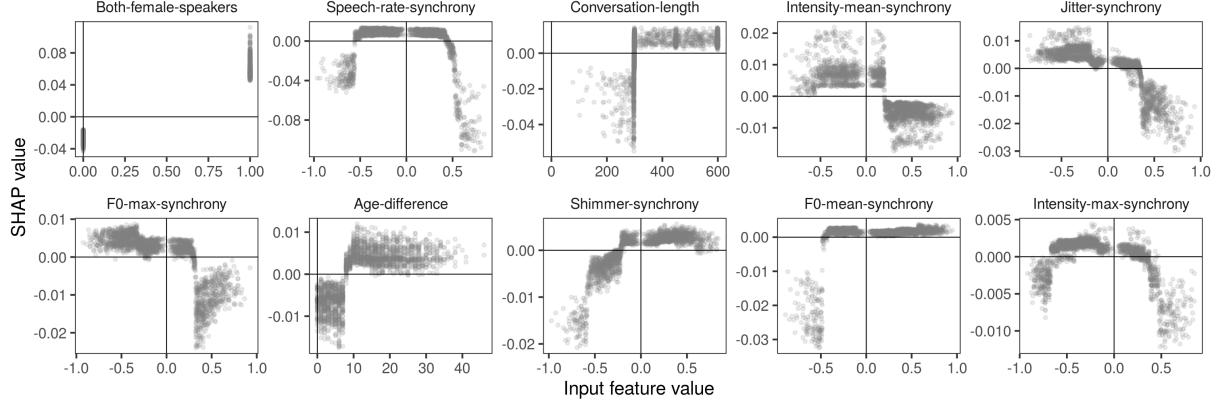


Figure 4: SHAP feature dependence plots for SWBD.

Table 3 contains the estimated feature importance values for both corpora analyzed. Each panel shows the top 10 features detected as most important in the best performing models (*external and synchrony* for SWBD; *external and all a/p entrainment metrics* for CCC). Both models were trained using the entirety of their respective corpus and making use of the best set of hyperparameters previously found.

For SWBD, Table 3 shows that the *both-female-speakers* indicator variable was by far the most important feature, followed by *speech-rate-synchrony*. For the case of CCC, speech rate entrainment metrics dominate the importance ranking, being *speech-rate-proximity* the most important one. Notably, no gender related feature is included in the ranking of the top 10 most important features for CCC.

Feature importance values are interesting in and of themselves, but say little of the way models make use of these features. To tackle this issue, Figure 4 presents SHAP feature dependence plots for SWBD’s 10 most important features. Horizontal lines centered at SHAP = 0 serve as a reference; values appearing above/below this line indicate that the model output tends to increase/decrease relative to the base output. Regarding the external features, Figure 4 shows that conversations in which both speakers are female were associated to higher values of predicted naturalness, that short conversations are predicted to be less natural (speakers were instructed to speak for at least five minutes, but were allowed to speak longer), and that large age differences lead to higher naturalness predictions.

It is interesting to note that high synchrony values are **not** necessarily associated with higher perceived naturalness predictions. It is the case for

shimmer-synchrony and *F0-mean-synchrony* (to a lesser extent). However, the opposite is observed for many a/p features. Moreover, *speech-rate-synchrony* and *intensity-max-synchrony* show an inverted U pattern, where extremely low or high entrainment values are associated to lower predicted values of the outcome.

Figure 5 presents a similar analysis for CCC. Regarding *conversation length*, the only high-ranked external feature, extremely short conversations are associated to lower predictions of self-reported satisfaction. Regarding a/p entrainment metrics, once again higher entrainment does not necessarily lead to higher predictions of the outcome variable. In fact, this is not the case for all entrainment metrics related to speech rate. Only *intensity-max-convergence* shows a positive relation between a/p entrainment and predicted satisfaction. Once again negative relations and inverted U patterns are observed (although the latter are less noticeable than in SWBD).

5 Discussion

In this work we proposed a unifying framework for modeling different types of a/p entrainment in natural conversations. We also tested on two very different corpora whether three metrics derived from our framework provide valuable information for predicting positive social outcomes in conversations (perceived naturalness in SWBD and self-reported customer satisfaction in CCC). Our results suggest that these metrics effectively relate to positive social outcomes. However, several remarks should be made.

First, the fact that the achieved AUC scores are greater than chance not only validates the proposed

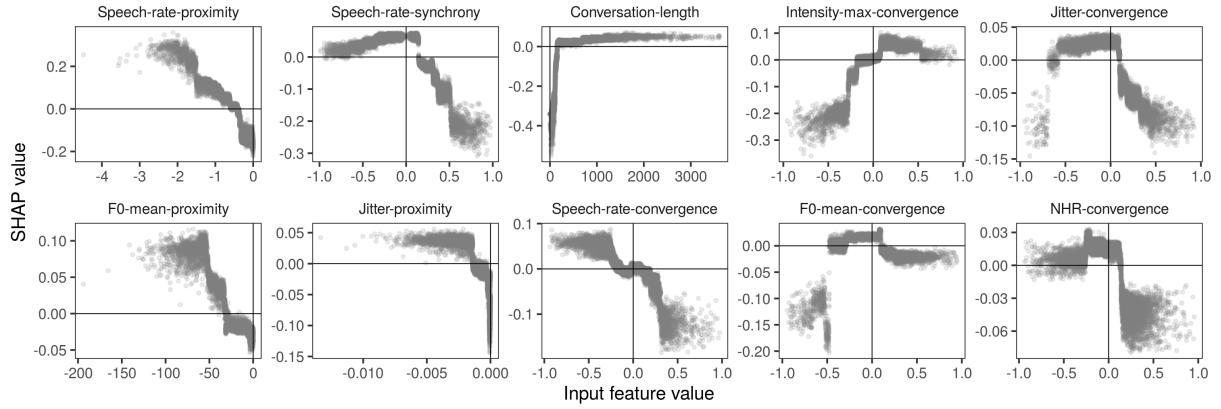


Figure 5: SHAP feature dependence plots for CCC.

metrics, but strongly suggests that a/p entrainment is related to positive social outcomes. Importantly, this result was not found on a single corpus, but on two independent ones. Both corpora had not only different positive social outcomes attached, but also differed in their domain and even in their language (English and Spanish). Future research should focus on testing if these results prevail across broader domains and for further social variables.

Second, even when the obtained results are higher than chance, they are far from being exceptionally high. This suggests that a/p entrainment metrics by themselves — at least the ones tested — may not contain enough predictive information as to achieve competitive results. Probably a competitive model should incorporate information regarding the semantic content of conversations and/or, for the case of a corpus like CCC, customer relationship management related information. However, this does not mean that the proposed metrics are of no use. Future research should focus on studying whether the information provided by a/p entrainment metrics complements the one provided by other sources. In our experiments we tested this up to some degree. In particular, we observed that a/p entrainment metrics complement the information contained in the external features. This effect is very strong for CCC and less strong for SWBD.

Third, the fact that the best set of features differs across corpora, suggests that which features predict positive social outcomes depends on the outcomes being predicted and the corpus itself. Note that a similar pattern was observed in Pérez et al. (Pérez et al., 2016) where, even on the same corpus, the significance of synchrony metrics calculated on different a/p features varied across different social outcomes.

Fourth, SHAP dependence plots suggest that the manner in which predictive models make use of entrainment metrics is quite complex. First of all, not always are higher entrainment values associated to higher predicted values of a positive social aspect. Rather, two more patterns are observed: a negative relation between a/p entrainment and positive outcomes, and an inverted U pattern. Additionally, in line with the third remark, it is interesting to note the effects of a given a/p entrainment metric are not the same across corpora, again suggesting heterogeneity across tasks and corpora. An illustrative case are the patterns observed for *speech-rate-synchrony*, for which an inverted U pattern is observed in SWBD and a negative relation is observed in CCC.

Finally, the reason why people do entrain is still unknown (see, for example, Natale, 1975; Giles et al., 1991; Chartrand and Bargh, 1999; Pickering and Garrod, 2004, 2013). Consequently, metrics such as the ones tested in this work, albeit noisy and imperfect, are likely to be capturing part of some more complex phenomenon that we do not fully understand yet. Further research on the causes of entrainment in human speech is still needed.

References

- Ethem Alpaydin. 2020. *Introduction to Machine Learning*, 4 edition. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA.
- James Bergstra and Yoshua Bengio. 2012. *Random search for hyper-parameter optimization*. *Journal of Machine Learning Research*, 13(10):281–305.
- Štefan Beňuš, Agustín Gravano, Rivka Levitan, Sarah Ita Levitan, Laura Willson, and Julia

- Hirschberg. 2014. Entrainment, dominance and alliance in supreme court hearings. *Knowledge-Based Systems*, 71:3 – 14.
- Paul Boersma and David Weenink. 2019. Praat: doing phonetics by computer [computer program]. Version 6.1.08, retrieved 5 December 2019 from <http://www.praat.org/>.
- Susan E. Brennan and Herbert H. Clark. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(6):1482–1493.
- Alejandro Cartas, Martin Kocour, Aravindh Raman, Ilias Leontiadis, Jordi Luque, Nishanth Sastry, Jose Nuñez Martinez, Diego Perino, and Carlos Segura. 2019. A reality check on inference at mobile networks edge. In *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking*, EdgeSys ’19, page 54–59, New York, NY, USA. Association for Computing Machinery.
- Tanya L. Chartrand and John A. Bargh. 1999. The chameleon effect: The perception–behavior link and social interaction. *Journal of Personality and Social Psychology*, 76(6):893–910.
- Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, page 785–794, New York, NY, USA. Association for Computing Machinery.
- Céline De Looze, Stefan Scherer, Brian Vaughan, and Nick Campbell. 2014. Investigating automatic measurements of prosodic accommodation and its dynamics in social interaction. *Speech Communication*, 58:11 – 34.
- Neeraj Deshmukh, Aravind Ganapathiraju, Andi Gleeson, Jonathan Hamaker, and Joseph Picone. 1998. Resegmentation of switchboard. In *ICSLP-1998*.
- Andrew Fandrianto and Maxine Eskenazi. 2012. Prosodic entrainment in an information-driven dialog system. In *INTERSPEECH-2012*, pages 342–345.
- Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*, volume 1. Springer series in statistics New York.
- Howard Giles, Nikolas Coupland, and Justine Coupland. 1991. *Accommodation theory: Communication, context, and consequence.*, Studies in emotion and social interaction., pages 1–68. Editions de la Maison des Sciences de l’Homme, Paris, France.
- John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *ICASSP-92, ICASSP’92*, page 517–520, USA. IEEE Computer Society.
- Agustín Gravano, Štefan Beňuš, Rivka Levitan, and Julia Hirschberg. 2015. Backward mimicry and forward influence in prosodic contour choice in standard american english. In *INTERSPEECH-2015*, pages 1839–1843.
- Patrick G. T. Healey, Matthew Purver, and Christine Howes. 2014. Divergence in dialogue. *PLOS ONE*, 9(6):1–6.
- Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2014. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated.
- Spyros Kousidis, David Dorran, Ciaran Mcdonnell, and Eugene Coyle. 2009. Time series analysis of acoustic feature convergence in human dialogues. In *SPECOM-2009*, St. Petersburg, Russian Federation.
- Rivka Levitan, Stefan Benus, Agustín Gravano, and Julia Hirschberg. 2015. Entrainment and turn-taking in human-human dialogue. In *AAAI Spring Symposium on Turn-Taking and Coordination in Human-Machine Interaction*.
- Rivka Levitan, Agustín Gravano, and Julia Hirschberg. 2011. Entrainment in speech preceding backchannels. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 113–117, Portland, Oregon, USA. Association for Computational Linguistics.
- Rivka Levitan and Julia Hirschberg. 2011. Measuring acoustic-prosodic entrainment with respect to multiple levels and dimensions. In *INTERSPEECH-2011*, pages 3081–3084.
- Quim Llimona, Jordi Luque, Xavier Anguera, Zoraida Hidalgo, Souneil Park, and Nuria Oliver. 2015. Effect of gender and call duration on customer satisfaction in call center big data. In *INTERSPEECH-2015*, pages 1825–1829.
- Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.
- Jordi Luque, Carlos Segura, Ariadna Sánchez, Martí Umbert, and Luis Angel Galindo. 2017. The role of linguistic and prosodic cues on the prediction of self-reported satisfaction in contact centre phone calls. In *INTERSPEECH-2017*, pages 2346–2350.
- Jan Michalsky and Heike Schoermann. 2017. Pitch convergence as an effect of perceived attractiveness and likability. In *INTERSPEECH-2017*, pages 2253–2256.
- Christoph Molnar. 2019. *Interpretable Machine Learning*. <https://christophm.github.io/interpretable-ml-book/>.

Michael Natale. 1975. Convergence of mean vocal intensity in dyadic communication as a function of social desirability. *Journal of Personality and Social Psychology*, 32(5):790–804.

Ani Nenkova, Agustín Gravano, and Julia Hirschberg. 2008. High frequency word entrainment in spoken dialogue. In *Proceedings of ACL-08: HLT, Short Papers*, pages 169–172, Columbus, Ohio. Association for Computational Linguistics.

Jennifer S. Pardo. 2006. On phonetic convergence during conversational interaction. *The Journal of the Acoustical Society of America*, 119(4):2382–2393.

Martin J. Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27(2):169–190.

Martin J. Pickering and Simon Garrod. 2013. An integrated theory of language production and comprehension. *Behavioral and Brain Sciences*, 36(4):329–347.

Pyphen. 2019. Pyphen is a pure python module to hyphenate text. retrieved: 2019-01-24.

Juan M. Pérez, Ramiro H. Gálvez, and Agustín Gravano. 2016. Disentrainment may be a positive thing: A novel measure of unsigned acoustic-prosodic synchrony, and its relation to speaker engagement. In *INTERSPEECH-2016*, pages 1270–1274.

David Reitter, Frank Keller, and Johanna D. Moore. 2011. A computational cognitive model of syntactic priming. *Cognitive Science*, 35(4):587–637.

David Reitter and Johanna D. Moore. 2014. Alignment and task success in spoken dialogue. *Journal of Memory and Language*, 76:29 – 46.

Michelina Savino, Loredana Lapertosa, Alessandro Caffò, and Mario Refice. 2016. Measuring prosodic entrainment in italian collaborative game-based dialogues. In *Speech and Computer*, pages 476–483, Cham. Springer International Publishing.

Antje Schweitzer and Natalie Lewandowski. 2014. Social factors in convergence of f1 and f2 in spontaneous speech. In *ISSP-2014*.

Richard L. Street. 1984. Speech convergence and speech evaluation in fact-finding interviews. *Human Communication Research*, 11(2):139–169.

Arthur Ward and Diane Litman. 2007. Measuring convergence and priming in tutorial dialog. *University of Pittsburgh*.

Andreas Weise, Sarah Ita Levitan, Julia Hirschberg, and Rivka Levitan. 2019. Individual differences in acoustic-prosodic entrainment in spoken dialogue. *Speech Communication*, 115:78 – 87.

Unsupervised Evaluation of Interactive Dialog with DialoGPT

Shikib Mehri and Maxine Eskenazi

Dialog Research Center, Language Technologies Institute

Carnegie Mellon University, USA

{amehri, max}@cs.cmu.edu

Abstract

It is important to define meaningful and interpretable automatic evaluation metrics for open-domain dialog research. Standard language generation metrics have been shown to be ineffective for dialog. This paper introduces the **FED** metric (fine-grained evaluation of dialog), an automatic evaluation metric which uses DialoGPT, without any fine-tuning or supervision. It also introduces the FED dataset which is constructed by annotating a set of human-system and human-human conversations with eighteen fine-grained dialog qualities. The FED metric (1) does not rely on a ground-truth response, (2) does not require training data and (3) measures fine-grained dialog qualities at both the turn and whole dialog levels. FED attains moderate to strong correlation with human judgement at both levels.

1 Introduction

Evaluation metrics often define the research direction of a field. As dialog systems begin to demonstrate human-level performance, the development and adoption of meaningful and interpretable automatic evaluation measures is essential (Zhang et al., 2019; Adiwardana et al., 2020). Since standard metrics (e.g., BLEU, METEOR) have been shown to be ineffective for dialog (Deriu et al., 2019; Liu et al., 2016), human evaluation is often used. However, it is typically only used as a final evaluation since it is costly. During development, systems are generally optimized for poorly correlated automatic metrics which can result in sub-par performance (Dinan et al., 2019). Automatic metrics must be meaningful and interpretable so that they can be used to compare dialog systems, understanding their respective strengths and weaknesses, and effectively guide dialog research.

Dialog evaluation is difficult for several reasons: (1) The one-to-many nature of dialog (Zhao

et al., 2017) makes word-overlap metrics ineffective for scoring valid responses that deviate from the ground-truth (Liu et al., 2016; Gupta et al., 2019). (2) Dialog quality is inherently multi-faceted (Walker et al., 1997; See et al., 2019) and an interpretable metric should measure several qualities (e.g., *interesting*, *relevant*, *fluent*). (3) Dialog systems have begun to be evaluated in an interactive setting (Ram et al., 2018; Adiwardana et al., 2020) where a real user has a back-and-forth conversation with a system. Interactive evaluation is not constrained to a static corpus and better captures the performance of a system in a realistic setting. However, the existing automatic metrics compare to a ground-truth response, making them unsuitable for assessing interactive conversations. To address these three problems, this paper presents the **FED** metric (fine-grained evaluation of dialog) which assesses eighteen qualities of dialog without relying on a reference response.

First, a dataset of human quality annotations is collected for the human-system (Meena and Mitsuku) and human-human conversations released by Adiwardana et al. (2020). Dialogs are annotated at both the turn level and the dialog level for eighteen fine-grained dialog qualities. This FED dataset can be used to benchmark the performance of automatic metrics relative to human judgement. Analysis of this data provides insight into the qualities of dialog that are most important to human annotators. It therefore highlights the qualities that should be the focus of attention in dialog research.

The FED dataset is intended only for evaluating automatic metrics relative to human judgement. It does not consist of any training data. As such, this paper addresses the task of developing an automatic evaluation metric which (1) does not compare to a reference response, (2) assesses eighteen different qualities of dialog and (3) relies on no training data or supervision. This paper is the first, to the best

of our knowledge, to address this important and challenging problem.

The FED metric described here leverages a massively pre-trained model, DialoGPT (Zhang et al., 2019), which can generate practically human-level responses. Kocijan et al. (2019) assert that pre-trained models implicitly capture world knowledge and can therefore perform common-sense reasoning. Similarly, we posit that DialoGPT has implicitly captured some notion of dialog quality and can therefore be used for dialog evaluation. Eskenazi et al. (2019) assessed the quality of a system utterance in an interactive setting by looking at the *following user response*. The proposed evaluation metric is based on the same intuition. Given a system response, its quality is measured by computing the likelihood that DialoGPT will respond to it with a particular follow-up utterance (e.g., “*That is really interesting!*”). DialoGPT is more likely to respond in this way to what it believes is an *interesting* system response. A set of follow-up utterances is constructed for each of the eighteen qualities and the likelihoods of these follow-up utterances are used to measure dialog quality.

The FED metric obtains moderate to strong correlation with human judgement for turn-level and dialog-level evaluation without any training data or ground-truth response. Analysis in this paper demonstrates that through large-scale pre-training, DialoGPT has implicitly captured some notion of dialog quality. These results suggest that pre-trained models can be leveraged to further improve dialog evaluation.

The contributions of this paper are as follows: (1) The FED dataset was collected for fine-grained evaluation of interactive dialog, with annotations for eighteen dialog qualities at both the turn- and the dialog-level. (2) Analysis of the FED dataset identifies the dialog qualities most important to human annotators. (3) DialoGPT is shown to implicitly capture an understanding of dialog quality. (4) The FED metric has moderate to strong correlation with human judgement by leveraging DialoGPT, without training data or reference responses.

2 Related Work

2.1 Automatic Dialog Evaluation

Standard automatic metrics for language generation have been shown to correlate poorly with human judgement of dialog (Liu et al., 2016; Lowe et al., 2017; Gupta et al., 2019). This poor per-

formance can largely be explained by the one-to-many nature of dialog (Zhao et al., 2017). To avoid comparing to a single reference response, several authors have proposed using multiple reference responses. Multiple reference responses can be obtained with retrieval models (Galley et al., 2015; Sordoni et al., 2015) or through data collection (Gupta et al., 2019). These multi-reference metrics show performance improvement, but it is infeasible to thoroughly cover the space of all potential responses. The FED metric does not rely on a ground-truth response.

Lowe et al. (2017) train ADEM to produce a quality score conditioned on the dialog context, the reference response and the generated response. Venkatesh et al. (2018) present a framework for evaluating Alexa prize conversations which attains moderate correlation with user ratings. Both methods are trained on explicit quality annotations. In contrast, the FED metric proposed here requires no supervision.

Mehri and Eskenazi (2020) introduce USR, an unsupervised and reference-free evaluation metric for dialog generation. Similar to FED, USR uses pre-trained models to assess several dialog qualities. However, they are limited to five qualities with hand-designed models and unsupervised tasks for each quality. In comparison, FED is more general and encapsulates eighteen dialog qualities.

2.2 Dialog Qualities

Human evaluation in dialog is often limited to only measuring overall quality or response appropriateness. However, dialog quality is multi-faceted and should not be reduced to a single measurement.

PARADISE (Walker et al., 1997), one of the first frameworks for dialog evaluation, measured several different properties of dialog and combined them to estimate user satisfaction. See et al. (2019) used a variety of human judgements for dialog including interestingness, making sense, avoiding repetition, fluency, listening and inquisitiveness. See et al. (2019) emphasize the importance of measuring multiple qualities when evaluating dialog systems. There are several examples of human evaluation of multiple dialog qualities. Gopalakrishnan et al. (2019) annotate system responses using: interesting, comprehensible, on-topic and use of knowledge. Shin et al. (2019) measure empathy, fluency and relevance. Zhang et al. (2019) evaluate responses using relevance, informativeness and

human-likeness. Adiwardana et al. (2020) evaluate in both static and interactive environments using specificity and sensibleness.

2.3 Pre-trained Dialog Models

The success of pre-trained language models (Radford et al., 2018; Devlin et al., 2018) has recently been extended to the domain of dialog. Zhang et al. (2019) pre-train DialoGPT on Reddit and attain human-level performance on the task of response generation. The open-source DialoGPT model was used to construct the FED metric presented in this paper. (Adiwardana et al., 2020) similarly pre-trained their Meena dialog system on an unspecified large conversational dataset.

3 Data Collection

A dataset of human quality annotations was collected to assess automatic metrics by measuring correlation with human judgements. Adiwardana et al. (2020) collected a set of conversations¹ between a human and two open-domain dialog systems, Meena (Adiwardana et al., 2020) and Mitsuku². In addition, they also released human-human dialogs collected in the same environment where one of the humans was selected to play the role of the system. We annotated a subset of these conversations with human quality judgements to create the FED dataset.

Workers on Amazon Mechanical Turk (AMT) annotated 40 Human-Meena conversations, 44 Human-Mitsuku conversations and 40 Human-Human conversations. For each conversation, three system responses were hand-selected to be annotated at the turn level, presented to the worker sequentially. Then the worker was shown the entire conversation and annotated on the dialog level. Five workers annotated each conversation. They did not know which system was involved in a conversation, since all mentions of the system name were replaced with the word “System.”

Since dialog quality is inherently multi-faceted it is important to measure several different qualities of dialog. Eighteen fine-grained dialog qualities are measured in the FED dataset: eight at the turn level and ten at the dialog level.

¹<https://github.com/google-research/google-research/tree/master/meena>

²<https://medium.com/pandorabots-blog/mitsuku-wins-loebner-prize-2018-3e8d98c5f2a7>

3.1 Turn-Level Annotation

Given a dialog context and a system response, the worker assessed the response according to eight fine-grained measures as well as for overall quality. The list of turn-level measures is shown in Table 1. The options for each of the fine-grained qualities were: *No*, *Somewhat*, *Yes*, *N/A*. For *understandable*, the *Somewhat* option was not provided, similar to prior past work (Gopalakrishnan et al., 2019). Responding *N/A* required written justification. The overall impression question was measured on a five-point Likert scale.

The workers were given detailed instructions and examples for each question presented in Table 1. These instructions are provided in the supplementary materials.

3.2 Dialog-Level Annotation

For dialog-level annotation, workers were asked to label the quality of a system over the duration of an entire conversation. The dialog-level questions listed in Table 2 cover ten fine-grained dialog qualities and an additional question on overall impression. The available options for each of the fine-grained qualities were *No*, *Somewhat*, *Yes*, *N/A*. For *consistency*, the *Somewhat* option was not provided because the existence of an inconsistency is binary. Overall impression was measured on a five-point Likert scale.

3.3 Dataset Statistics

A total of 124 conversations were annotated (40 Meena, 44 Mitsuku, 40 Human). Five different workers saw each conversation (HIT). Each conversation had one dialog-level annotation and three turn-level annotations for chosen system responses that were randomly sampled from the conversation. There were 9 questions for turn-level annotation and 11 for dialog-level annotation. In total, the FED dataset includes 3348 turn-level and 1364 dialog-level data points, for a total of 4712. This dataset intended to be used solely for the evaluation of metrics, as the number of annotated conversations is not large enough to accommodate both training and testing.

3.4 Data Processing

Given that each of the 4712 data points was labeled by five annotators, post-processing was used to improve the quality of the data through the removal of outliers. Given five annotations for a given ques-

Question	Used By
To the average person, is the response interesting ?	See et al. (2019); Gopalakrishnan et al. (2019); Mehri and Eskenazi (2020)
Is the response engaging ?	Yi et al. (2019)
Is the response generic or specific to the conversation?	Adiwardana et al. (2020)
Is the response relevant to the conversation?	See et al. (2019); Gopalakrishnan et al. (2019); Shin et al. (2019); Zhang et al. (2019); Mehri and Eskenazi (2020)
Is the response correct or was there a misunderstanding of the conversation?	None specifically
Is the response semantically appropriate ?	See et al. (2019)
Is the response understandable ?	Gopalakrishnan et al. (2019); Mehri and Eskenazi (2020)
Is the response fluently written ?	See et al. (2019); Shin et al. (2019); Zhang et al. (2019); Ghandeharioun et al. (2019); Mehri and Eskenazi (2020)
Overall impression of the response?	Many

Table 1: The questions asked for turn-level annotation. Examples of prior work that has used each dialog quality are listed. No one has specifically used *Correct*, however its meaning is often encapsulated in *Relevant*.

Question	Used By
Throughout the dialog, is the system coherent and maintain a good conversation flow?	See et al. (2019)
Is the system able to recover from errors that it makes?	None
Is the system consistent in the information it provides throughout the conversation?	Qin et al. (2019)
Is there diversity in the system responses?	See et al. (2019); Ghandeharioun et al. (2019)
Does the system discuss topics in depth ?	Guo et al. (2018)
Does the system display a likeable personality?	Shin et al. (2019); Ghandeharioun et al. (2019)
Does the system seem to understand the user?	See et al. (2019)
Is the system flexible and adaptable to the user and their interests?	Guo et al. (2018)
Is the system informative throughout the conversation?	Zhang et al. (2019)
Is the system inquisitive throughout the conversation?	See et al. (2019)
Overall impression of the dialog?	Many

Table 2: The qualities annotated at the dialog-level. Examples of prior work that has used each dialog quality are listed. To our knowledge, error recovery has not been used for human evaluation.

Quality	Spearman
Turn-Level	
Interesting	0.819
Engaging	0.798
Specific	0.790
Relevant	0.753
Correct	0.780
Semantically Appropriate	0.682
Understandable	0.522
Fluent	0.714
Overall Impression	0.820
Dialog-Level	
Coherent	0.809
Error Recovery	0.840
Consistent	0.562
Diverse	0.789
Topic Depth	0.833
Likeable	0.838
Understanding	0.809
Flexible	0.816
Informative	0.806
Inquisitive	0.769
Overall Impression	0.830

Table 3: Spearman correlation for each of the dialog qualities. The correlation was measured by correlating each annotation with the mean of the other annotations for the same question.

tion, the furthest label from the mean is removed if its distance from the mean is greater than half the standard deviation of the five annotations.

4 Data Analysis

The fine-grained nature of the FED dataset is grounds for a rich analysis. First, inter-annotator agreement is evaluated for all of the dialog qualities. Next, the dataset is used to better understand the comparative strengths and weaknesses of the three systems (Mitsuku, Meena, Human). Finally, detailed analysis of the data provides insight into the fine-grained qualities that most strongly contribute to the annotators' overall impression.

4.1 Inter-Annotator Agreement

To compute inter-annotator agreement, the correlation between each annotation and the mean of the five (or four, after outlier removal) annotations for the same question is measured. The Spearman correlation for each turn-level and dialog-level question is shown in Table 3

Quality	Mitsuku	Meena	Human
Turn-Level			
Interesting	2.30	2.58	2.35
Engaging	2.53	2.75	2.49
Specific	2.48	2.74	2.56
Relevant	2.80	2.88	2.74
Correct	2.74	2.84	2.66
Semantically-Appropriate	2.84	2.92	2.85
Understandable	0.97	0.97	0.94
Fluent	2.83	2.90	2.80
Overall	3.81	4.19	3.85
Dialog-Level			
Coherent	2.20	2.88	2.94
Error Recovery	2.22	2.69	2.86
Consistent	0.82	0.95	0.98
Diverse	2.23	2.46	2.88
Topic Depth	1.80	2.28	2.78
Likeable	2.10	2.61	2.97
Understanding	2.23	2.86	2.98
Flexible	2.22	2.72	2.97
Informative	2.10	2.60	2.85
Inquisitive	2.35	2.76	2.88
Overall	3.10	4.11	4.60

Table 4: Performance of each system on the fine-grained qualities. All scores are 1-3, except Understandable and Consistent are 0-1 and Overall is 1-5.

Inter-annotator agreement is high for all of the dialog qualities, suggesting that all of the qualities were well-understood by the annotators and relevant and that the instructions removed much of the ambiguity from the task. Two qualities, *understandable* and *consistent*, have slightly lower correlations, in the 0.5 - 0.6 range. These qualities did not include *Somewhat* as an answer. This probably contributed to the lower inter-annotator agreement.

4.2 System Performance

While Adiwardana et al. (2020) presented a performance comparison between Mitsuku, Meena and Humans in an interactive setting, their evaluation only used two qualities: *specificity* and *sensitivity*. In contrast, the FED dataset has eighteen fine-grained qualities thus providing more information about the strengths and weaknesses of each system.

The fine-grained performance of each system shown in Table 4. For all of the turn-level qualities,

Meena outperforms both Mitsuku and Human. The strength of Meena is most noticeable for *interesting*, *engaging* and *specific*.

However, turn-level qualities are insufficient to evaluate a dialog system. Dialog is by definition a multi-turn interaction. Thus, in some cases, a sub-optimal system response might result in a better long-term dialog. Humans significantly outperform the two systems for dialog-level qualities. The difference between Meena and Mitsuku is very pronounced at the dialog level, with a 1 point difference in overall score. The higher variance in scores and the stronger performance of human dialogs, shows that dialog-level evaluation is reliable than turn-level. Meena’s scores suggest that it is fairly *coherent*, *understanding* and *flexible*. However, it struggles with *diversity*, *topic depth* and *likeable*.

4.3 Fine-Grained Quality Analysis

The FED dataset can be used to examine the relative importance of each fine-grained dialog quality by measuring its contribution to the overall impression. For both turn-level and dialog-level, a regression is trained to predict the overall score given the fine-grained qualities as input. The regression weights provide insight into the fine-grained qualities that most contribute to the overall impression as labeled by human annotators. A softmax is computed over the regression weights to determine the relative contribution of each fine-grained dialog quality. A dialog quality with a higher weight contributes more to the human’s overall impression. The results are shown in Table 5.

The most important turn-level qualities are *interesting*, *relevant* and *fluent*. This suggests that developing a system that is consistently interesting, relevant and fluent will result in the highest improvement in the user’s overall impression. There is less variance in the importance of dialog-level qualities than in the turn-level qualities possibly because there is less overlap in meaning amongst the qualities and all of the dialog-level qualities seem somewhat important. The most important dialog-level qualities are *coherent*, *likeable* and *understanding*. Improving a system’s coherence, understanding of the user and its likeableness would thus be the most likely way to improve the overall impression of a dialog system.

Quality	Importance (%)
Turn-Level	
Interesting	16.15
Engaging	7.46
Specific	9.64
Relevant	18.10
Correct	13.77
Semantically Appropriate	9.90
Understandable	10.70
Fluent	14.27
Dialog-Level	
Coherent	10.95
Error Recovery	9.15
Consistent	7.92
Diverse	10.09
Topic Depth	10.51
Likeable	12.03
Understanding	11.01
Flexible	10.34
Informative	8.00
Inquisitive	9.50

Table 5: Relative importance of each dialog quality for predicting the overall impression. The most important qualities for turn-level and dialog-level are in bold.

5 Methods

The FED (fine-grained evaluation of dialog) metric is an automatic evaluation metric for dialog which (1) does not need to compare to a reference response, (2) measures eighteen fine-grained qualities of dialog, and (3) does not use training data. Capturing a diverse set of fine-grained qualities without supervision is an especially challenging problem.

The development of the FED metric is motivated by two areas of prior work: (1) pre-trained language models and their capabilities and (2) the use of follow-up utterances as a means of evaluation.

5.1 DialoGPT

Zhang et al. (2019) extend GPT-2 (Radford et al., 2018) to train DialoGPT on 147M conversation-like interactions from Reddit. As per their evaluation, DialoGPT outperforms humans at producing relevant, interesting and human-like responses.

Kocijan et al. (2019) show that pre-trained language models, specifically BERT (Devlin et al., 2018), implicitly capture world knowledge and can therefore perform common sense reasoning. By calculating which answer results in a more proba-

ble sentence according to BERT, they strongly outperform other methods on the Winograd Schema Challenge (Levesque et al., 2012).

Just as BERT has been shown to capture world knowledge, we posit that DialoGPT has implicitly captured some notion of dialog quality. The qualities of a particular dialog context (e.g., *interesting*, *relevant*, *informative*) likely inform DialoGPT’s response and, as such, must be captured by the model. If there was training data for the eighteen dialog qualities, this hypothesis could be verified by fine-tuning DialoGPT for the task of dialog evaluation. Without training data, however, the challenge is to devise an unsupervised mechanism for extracting the quality information captured by DialoGPT.

5.2 Follow-Up Utterance for Evaluation

Eskenazi et al. (2019) assess the quality of a system utterance in an interactive setting, by looking at the *following user response*. When users speak to a system, their response to a given system utterance may implicitly or explicitly provide feedback for the system. For example, if a user follows up a system utterance with “*That’s not very interesting*”, they are providing information about the quality of the system utterance.

The conversations in the FED dataset were collected in an interactive setting. Thus the use of the follow-up utterance is a valid option. Even if users consistently provided feedback, it would be difficult to interpret without training data.

5.3 Evaluating with DialoGPT

The proposed FED metric is motivated by (1) the intuition that DialoGPT has implicitly learned to reveal dialog quality and (2) that the follow-up utterance can provide valuable information about a system response. To measure the quality of a system response s , we compute the likelihood of the model generating various follow-up utterances (e.g., “*Wow! Very interesting.*”) in response to s . DialoGPT will be more likely to respond with a positive follow-up utterance if given a better (e.g., more *interesting/relevant/fluent*) preceding system utterance.

For each of the eighteen fine-grained dialog qualities, a set of positive follow-up utterances, p , and a set of negative follow-up utterances, n , is constructed. Specifically, given a dialog context c , a system response r and a function \mathcal{D} that computes the log-likelihood of DialoGPT generating a particular response, the predicted score for a dialog

quality is calculated as:

$$\sum_{i=1}^{|p|} \mathcal{D}(c + r, p_i) - \sum_{i=1}^{|n|} \mathcal{D}(c + r, n_i) \quad (1)$$

This equation can be modified to predict scores for dialog-level qualities, by simply removing the system response r from the equation.

A response is said to be *interesting* if it is more likely that DialoGPT (acting as the user) responds with a positive follow-up utterance (e.g., “*Wow! Very interesting*”) than with a negative one (e.g., “*That’s really boring*”). For each of the eighteen qualities, several positive and negative utterances were hand-written and minimally tuned on a small subset of the dataset (10 conversations). Follow-up utterances for each quality are provided in the supplementary materials.

Generally, negative follow-up utterances are more meaningful than positive ones. For example, if a system response is *irrelevant*, a follow-up utterance of “*That’s not relevant*” is reasonable. However, acknowledging the relevance of a system response is less likely. Therefore the log-likelihood produced by DialoGPT will be noisier and less informative. The number of positive utterances for each dialog quality ranges between 0 and 4, and the number of negative utterances ranges between 1 and 4. While the fine-grained qualities are computed in this manner, the overall impression scores are calculated as an average of the scores for either the turn-level or dialog-level qualities.

6 Results

6.1 Experimental Setup

The FED metric was evaluated using four variations of the pre-trained DialoGPT model. The pre-trained DialoGPT models can be either medium size: 345M or large: 762M. They are either fine-tuned from GPT-2 (Radford et al., 2018) or trained from scratch. The follow-up utterances were hand-written and minimally tuned on 10 conversations using the 762M fine-tuned model. The small (117M) DialoGPT model was not used since Zhang et al. (2019) demonstrated its poor performance.

Most of the turn-level qualities were scored using only the last system response as context. For *relevant*, *correct* and dialog-level metrics, the entire conversation was used as context.

6.2 Correlation with Human Judgement

The Spearman correlation was measured between the predicted quality scores and the mean of the annotated scores. Correlations for all the dialog qualities, and all four variations of the underlying DialoGPT model are shown in Table 6.

The best overall turn-level correlation is **0.209** and the best overall dialog-level correlation is **0.443**. To our knowledge, there are presently no other metrics that operate without a ground-truth response, thus these results cannot be directly compared to any existing metrics. However, prior work on dialog evaluation reveals roughly similar correlation. Multi-reference evaluation for dialog achieves correlations in the 0.10 - 0.27 range (Gupta et al., 2019) and ADEM has correlations in the 0.28 - 0.42 range (Lowe et al., 2017). Given neither training data nor ground-truth response, the FED metric performs competitively relative to this prior work.

6.3 Discussion

The FED metric works better for some dialog qualities than others. This is because DialoGPT was trained on Reddit. It is more likely that it has captured certain dialog qualities that Reddit exhibits. For example, it is more likely that DialoGPT learns to measure qualities like *interesting* and *engaging*, than *understandable* and *consistent*. In the Reddit training data, the former two qualities show more variation than the latter. For example, there are interesting and un-interesting utterances, however most utterances on Reddit are generally understandable. The former two qualities are also more likely to influence the system response. Conversely, the latter two qualities are unlikely to be acknowledged in the response. For example, since Reddit is a multi-participant forum and not a one-on-one conversation, inconsistencies in conversation history are unlikely to be reflected in the response. As such, it is unsurprising that this approach struggles to measure the consistency of a dialog.

An optimal generation model (e.g., a human) should exhibit compositionality and be capable of producing utterances that have never been observed. For example, even if '*That is not consistent*' has never appeared in the training data, a compositional model would be capable of generating it. This difference in performance across the different dialog qualities suggests that DialoGPT exhibits some degree of compositionality, as evidenced by its ability to compose some follow-up utterances which are

not frequently observed in the Reddit data (e.g., '*You really don't know much?*'), however it still struggles with follow-up utterances consisting of less frequently observed concepts (e.g., *consistent*, *understandable*).

DialoGPT could be used to better measure these qualities by fine-tuning on additional conversational data from a source other than Reddit or on a training set annotated with human quality judgements. However, even without additional fine-tuning, FED effectively measures many qualities.

This paper has carried out an assessment of the FED metric for three open-domain conversation agents: Meena, Mitsuku and Human. Since these three systems are different in nature and FED exhibits strong correlation with human judgements across all the systems, we believe that the performance of FED will hold for other open-domain dialog systems and will not be restricted to a particular type of model or a specific dataset. However, the FED dataset consists of only open-domain chit-chat conversations. As such, future work is needed to determine whether the FED metric will generalize to goal-oriented dialog. Since DialoGPT has not observed goal-oriented training data, it may be necessary to use self-supervised fine-tuning on the new domain (Mehri and Eskenazi, 2020).

As with all automated metrics, there is the potential to game the FED metric and obtain artificially high scores, especially by having a model produce responses that are likely to result in specific follow-up utterances. To this end, the FED metric is not a replacement for human evaluation. It is instead a means of measuring dialog quality for the purposes of validation and model tuning.

The FED metric is (1) unsupervised, (2) does not rely on a reference response and (3) can be used to assess many dialog qualities. By having DialoGPT play the role of the user and assign probabilities to follow-up utterances, we have devised a mechanism of extracting information about dialog quality without any supervision. This mechanism is versatile and could potentially be extended to other dialog qualities.

7 Conclusion

This paper introduces the FED dataset and the FED metric. The FED dataset is constructed by annotating a set of interactive conversations with eighteen fine-grained dialog qualities. The FED metric can be used to measure fine-grained qualities of dia-

Quality	345M fs	345M ft	762M fs	762M ft
Turn-Level				
Interesting	0.388	0.431	0.406	0.408
Engaging	0.268	0.285	0.278	0.318
Specific	0.260	0.326	0.270	0.267
Relevant	<i>0.028</i>	<i>-0.027</i>	<i>0.001</i>	0.152
Correct	<i>0.000</i>	<i>0.037</i>	<i>0.020</i>	0.133
Semantically Appropriate	<i>0.040</i>	0.177	0.141	0.155
Understandable	<i>0.047</i>	<i>0.048</i>	<i>0.075</i>	0.111
Fluent	0.157	0.184	0.133	0.224
Overall	0.122	<i>0.092</i>	<i>0.094</i>	0.209
Dialog-Level				
Coherent	0.195	<i>0.151</i>	<i>0.149</i>	0.251
Error Recovery	<i>0.165</i>	<i>0.128</i>	<i>0.126</i>	<i>0.165</i>
Consistent	<i>0.041</i>	<i>0.011</i>	<i>0.006</i>	<i>0.116</i>
Diverse	0.449	0.431	0.414	0.420
Topic Depth	0.522	0.479	0.470	0.476
Likeable	<i>0.047</i>	<i>0.172</i>	<i>0.224</i>	0.262
Understanding	0.237	0.174	0.192	0.306
Flexible	0.260	0.408	0.298	0.293
Informative	0.264	0.328	0.337	0.288
Inquisitive	<i>0.137</i>	<i>0.143</i>	0.298	0.163
Overall	0.401	0.359	0.355	0.443

Table 6: Spearman correlations with human judgement. All values that are not statistically significant ($p > 0.05$) are italicized. The highest correlation for each quality is shown in bold.

log without comparing to a ground-truth response. By having DialoGPT take the role of the user and calculate the likelihood of follow-up utterances, the FED metric attains moderate to strong correlation with human judgement, without the use of any training data. The FED metric is inherently versatile and generalizable, making it applicable to other dialog qualities, domains or tasks. Both the FED dataset and the code for the FED metric will be released upon acceptance of this paper.

This paper sets the groundwork for several directions of future work. (1) The FED dataset can be used to benchmark automatic evaluation metrics on eighteen fine-grained dialog qualities. (2) Building on this paper, future work could identify mechanisms that further leverage pre-trained models for dialog evaluation. (3) Future work can explore strategies for extending the FED metric beyond open-domain chit-chat conversations to goal oriented dialog. (4) The FED metric can be used to evaluate, analyze and improve dialog systems.

References

- Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.
- Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2019. Survey on evaluation methods for dialogue systems. *arXiv preprint arXiv:1905.04071*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Julian Serban, Ryan Lowe, et al. 2019. The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*.
- Maxine Eskenazi, Shikib Mehri, Evgeniia Razumovskaia, and Tiancheng Zhao. 2019. Beyond turning: Intelligent agents centered on the user. *arXiv preprint arXiv:1901.06613*.

- Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. 2015. *deltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets*. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 445–450, Beijing, China. Association for Computational Linguistics.
- Asma Ghandeharioun, Judy Hanwen Shen, Natasha Jaques, Craig Ferguson, Noah Jones, Agata Lapedriza, and Rosalind Picard. 2019. Approximating interactive human evaluation with self-play for open-domain dialog systems. In *Advances in Neural Information Processing Systems*, pages 13658–13669.
- Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, Dilek Hakkani-Tür, and Amazon Alexa AI. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. *Proc. Interspeech 2019*, pages 1891–1895.
- Fenfei Guo, Angeliki Metallinou, Chandra Khatri, Anirudh Raju, Anu Venkatesh, and Ashwin Ram. 2018. Topic-based evaluation for conversational bots. *arXiv preprint arXiv:1801.03622*.
- Prakhar Gupta, Shikib Mehri, Tiancheng Zhao, Amy Pavel, Maxine Eskenazi, and Jeffrey P Bigham. 2019. Investigating evaluation of open-domain dialogue systems with human generated multiple references. *arXiv preprint arXiv:1907.10568*.
- Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019. A surprisingly robust trick for winograd schema challenge. *arXiv preprint arXiv:1905.06290*.
- Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.
- Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.
- Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic Turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126, Vancouver, Canada. Association for Computational Linguistics.
- Shikib Mehri and Maxine Eskenazi. 2020. Usr: An unsupervised and reference free evaluation metric for dialog generation. *arXiv preprint arXiv:2005.00456*.
- Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, Yangming Li, and Ting Liu. 2019. Entity-consistent end-to-end task-oriented dialogue system with kb retriever. *arXiv preprint arXiv:1909.06762*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. URL <https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/language-understanding-paper.pdf>.
- Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*.
- Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. *arXiv preprint arXiv:1902.08654*.
- Jamin Shin, Peng Xu, Andrea Madotto, and Pascale Fung. 2019. Happybot: Generating empathetic dialogue responses by improving user experience lookahead. *arXiv preprint arXiv:1906.08487*.
- Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and William B. Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *HLT-NAACL*.
- Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Behnam Hedayatnia, Angeliki Metallinou, et al. 2018. On evaluating and comparing open domain dialog systems. *arXiv preprint arXiv:1801.03625*.
- Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella. 1997. Paradise: A framework for evaluating spoken dialogue agents. *arXiv preprint cmp-lg/9704004*.
- Sanghyun Yi, Rahul Goel, Chandra Khatri, Tagyoung Chung, Behnam Hedayatnia, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2019. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. *arXiv preprint arXiv:1904.13015*.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi.
2017. Learning discourse-level diversity for neural
dialog models using conditional variational autoen-
coders. *arXiv preprint arXiv:1703.10960*.

Towards Unified Dialogue System Evaluation: A Comprehensive Analysis of Current Evaluation Protocols

Sarah E. Finch

Department of Computer Science
Emory University
Atlanta, GA, USA
sfillwo@emory.edu

Jinho D. Choi

Department of Computer Science
Emory University
Atlanta, GA, USA
jinho.choi@emory.edu

Abstract

As conversational AI-based dialogue management has increasingly become a trending topic, the need for a standardized and reliable evaluation procedure grows even more pressing. The current state of affairs suggests various evaluation protocols to assess chat-oriented dialogue management systems, rendering it difficult to conduct fair comparative studies across different approaches and gain an insightful understanding of their values. To foster this research, a more robust evaluation protocol must be set in place. This paper presents a comprehensive synthesis of both automated and human evaluation methods on dialogue systems, identifying their shortcomings while accumulating evidence towards the most effective evaluation dimensions. A total of 20 papers from the last two years are surveyed to analyze three types of evaluation protocols: *automated*, *static*, and *interactive*. Finally, the evaluation dimensions used in these papers are compared against our *expert* evaluation on the system-user dialogue data collected from the Alexa Prize 2020.

1 Introduction

Most successful automated dialogue systems follow task-oriented dialogue management methodology, which defines an explicit goal that the system is seeking to fulfill through the conversation with the user (Gao et al., 2019). Recently, the research in chat-oriented dialogue management has experienced a substantial increase in popularity. Unlike task-oriented dialogues, where the success is generally measured as ability to complete the goal of the task, evaluation of chat-oriented dialogues is much less straightforward, since the conversational goals can be highly subjective (Huang et al., 2019).

The evaluation of chat-oriented dialogue systems has been typically accomplished through the use of automated metrics and human evaluation (Section 2). Automated evaluation requires no human

labor once the evaluation script is written (Section 3). For automated evaluation to be a reliable measurement of the dialogue system quality, however, it needs to be shown to be a close approximation of human judgements (Section 4). Unfortunately, commonly used automated metrics correlate weakly with human judgments, indicating poor utility of such metrics (Liu et al., 2016). Human evaluation has become more commonplace in recent dialogue system works; however, it presents its own challenges. For one, it is time-consuming and expensive to obtain human judgments. More critically, there is a lack of standardized protocol for such human evaluation, which makes it challenging to compare different approaches to one another.

There have been many previous attempts at standardizing dialogue system evaluations. A major limitation has been their focus on task-oriented dialogue systems, which does not translate well to chat-oriented dialogue systems (Walker et al., 1997; Malchanau et al., 2019). Previous works which have included chat-oriented evaluations have lacked comprehensive coverage over the many varieties of such evaluation procedures that are currently in use. Instead, the emphasis has rested primarily on automated metrics at the expense of detailed analysis of human evaluation (Deriu et al., 2019). At this stage in conversational AI, it is probable that automated and human metrics reveal different aspects of dialogue systems (Hashimoto et al., 2019). It would be remiss to focus on a single evaluation category when assessing the state of the field. For this reason, our work aims to fill in the gaps of previous dialogue system evaluation surveys by identifying and comparing human evaluation protocols for chat-oriented dialogue systems.

To this end, we present a comparative analysis of the evaluations used for chat-oriented dialogue systems over the past several years. Since the field of conversational AI has experienced a rapid growth

in these years, it presents a unique opportunity to observe and assess which evaluation metrics have been most widely adopted by the larger community in this period of expeditious development. We provide a detailed survey of both automated and human evaluations in order to present the most accurate depiction of the current evaluation protocols. However, our in-depth analysis is limited to that of the human evaluations due to the abundance of previous work in automated metric analysis. As such, we defer to such work as [Liu et al. \(2016\)](#), [Ghandeharioun et al. \(2019\)](#), and [Ghazarian et al. \(2019\)](#) for more detail on automated metrics.

As a part of our analysis, we also present a case study of real human-machine dialogues which explores the significance of different human evaluation metrics in terms of overall user satisfaction through an expert analysis. As a result of our work, the most commonly used evaluation metrics in contemporary literature - both automated and human - are revealed in detail and our findings towards the prevalence, impact, and applicability of human evaluation metrics are illustrated.

2 Evaluation Protocols

For a holistic understanding of current evaluation protocols on dialogue systems, we have carefully selected 20 relevant papers since 2018, primarily from top-tier venues, and synthesized their methods. These papers focus on open domain (or non-task-oriented) dialogue, and employ a variety of approaches including:¹

- Incorporation of knowledge bases [2, 4, 7, 18, 20]
- Integration of personality [8, 12]
- Handling of emotion-driven responses [10]
- Purely depending on neural-based sequence-to-sequence models [19]

Based on these papers, three main categories are found as evaluation protocols for open-domain dialogue systems: *automated*, *static*, and *interactive*.

¹Throughout the paper, the following are used to refer to the related work: 1: [Li and Sun \(2018\)](#) 2: [Liu et al. \(2018\)](#) 3: [Luo et al. \(2018\)](#) 4: [Moghe et al. \(2018\)](#) 5: [Parthasarathi and Pineau \(2018\)](#) 6: [Xu et al. \(2018\)](#) 7: [Young et al. \(2018\)](#) 8: [Zhang et al. \(2018\)](#) 9: [Du and Black \(2019\)](#) 10: [Li et al. \(2019\)](#) 11: [Lin et al. \(2019\)](#) 12: [Madotto et al. \(2019\)](#) 13: [Qiu et al. \(2019\)](#) 14: [Tian et al. \(2019\)](#) 15: [Wu et al. \(2019\)](#) 16: [Zhang et al. \(2019\)](#) 17: [Zhou et al. \(2019\)](#) 18: [Zhu et al. \(2019\)](#) 19: [Adiwardana et al. \(2020\)](#) 20: [Wang et al. \(2020\)](#).

Automated evaluation is performed systematically by a batch script such that no human effort is required once the script is written (Section 2.1). Static evaluation is done by human where the evaluator assesses a dialogue whose last utterance is generated by the dialogue system (Section 2.2). Interactive evaluation is also done by human, although the evaluator assesses the quality of the dialogue after directly interacting with the dialogue system (Section 2.3).

Table 1 shows the distributions of the three evaluation protocols. Most recent approaches adopt both automated and human evaluations, with only 2 papers not including any form of human evaluation. The most common protocol for human evaluation is static evaluation, with very few papers conducting interactive assessments of dialogue systems. No work has adopted all three types of evaluation protocols.

Method	References	#
AUT	[1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12 13, 14, 15, 16, 17, 20]	17
STA	[1, 3, 4, 7, 9, 10, 11, 12, 13, 14 15, 16, 17, 18, 19, 20]	16
INT	[8, 19]	2
AUT & STA	[1, 3, 4, 7, 9, 10, 11, 12, 13, 14 15, 16, 17, 20]	14
AUT & INT	[]	0
STA & INT	[19]	1

Table 1: Distributions of the three evaluation protocols. #: number of papers using the corresponding protocol, AUT/STA/INT: automated/static/interactive evaluation. &: approaches using both protocols.

2.1 Automated Evaluation

Automated evaluation provides an objective quantitative measurement of the dialogue systems by operationalizing various dimensions of dialogue into mathematical formulations. Depending on the specific objectives behind different systems, a few studies define novel automated metrics to capture the benefit of their proposed approaches. Automated evaluation provides the most straightforward and undemanding methods by which to evaluate dialogue systems; however, they are generally viewed as poor indicators of true dialogue quality, following results from [Liu et al. \(2016\)](#).

2.2 Static Evaluation

Static evaluation is an offline procedure where the evaluators never directly interact with the dialogue systems under review; instead, they are provided with dialogue excerpts. These excerpts are gen-

erated by first randomly sampling dialogues from a corpus consisting of human-to-human conversations, then having the systems produce responses to the sampled dialogues. The sampled dialogues together with the system responses are provided to human evaluators to assess. Because only the last utterance in these excerpts are generated by the dialogue systems, it is difficult to evaluate sequential aspects about dialogue management through static evaluation (e.g., coherence among responses generated by the same system).

2.3 Interactive Evaluation

Unlike static evaluation, interactive evaluation has the same person play the role of both the user (one who interacts with the system) and the evaluator. In this setup, the evaluator has a conversation with the dialogue system and makes the assessment at the end of the conversation. Even though this procedure is more demanding in terms of time and human effort than static evaluation, it allows the evaluator to gain a better sense of the capability of the dialogue system through explicit interaction.

3 Analysis of Automated Evaluation

Table 2 shows the 11 metrics used for automated evaluation in our survey:

- BLEU: a subset of BLEU-1 through BLEU-4 (Papineni et al., 2002)
- C: sum of entailment scores between response and persona description (Madotto et al., 2019)
- Coherence: average word embedding similarity between dialogue context and generated response (Xu et al., 2018)
- Distinct: a subset of Distinct-1, Distinct-2, and Distinct-sentence (Li et al., 2016)
- Embedding: a subset of average, extrema, and greedy embedding similarity (Liu et al., 2016)
- Entity A/R: Accuracy and recall for including the correct entities in the response (Liu et al., 2018)
- Entity Score: average number of entities per response (Young et al., 2018)
- Entropy: average character-level entropy over all responses (Mou et al., 2016)

- **Inertia**: inertia on the clusters of embeddings of responses (Du and Black, 2019)
- **Perplexity**: inverse likelihood of predicting the responses of the test set (Chen et al., 1998)
- ROUGE: a subset of ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004)

The automated metrics in Table 2 fall into the following five categories:

Ground Truth Response Similarity Most commonly used automated metrics focus on assessing how well system responses match the ground truth human responses, using word overlap (BLEU, ROUGE) or embedding similarity.

Context Coherence Embedding similarities between dialogue contexts and system responses have been used to quantitatively assess the relevance between the system responses and the preceding dialogue history (Coherence, Embedding).

Response Diversity Other widespread metrics assess the diversity of the system responses in order to determine the amount of repetition and generic content in the system responses (Distinct, Entropy, Inertia, Entity Score).

Language Model Fitness Generative models are usually evaluated in terms of how well they learn to model the language of the dialogues in their training corpus (Perplexity).

Application-Specific The other observed metrics can be considered application-specific since Entity A/R is used to measure the ability of the system to produce the correct entities in its responses and C is specifically created as a measure of the consistency between the dialogue responses and their respective persona descriptions.

4 Analysis of Human Evaluation

While automated evaluation measures dimensions of dialogue objectively, human evaluation captures the subjective assessment from the user’s point of view. Regardless of the exact method chosen, all human evaluations involve gathering external annotators who answer questions regarding the dialogues resulting from a dialogue system.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	#	
BLEU			✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	14	
C												✓										1
Coherence						✓																1
Distinct	✓		✓		✓			✓				✓	✓	✓	✓	✓						9
Embedding	✓							✓				✓	✓				✓					5
Entity A/R		✓																				1
Entity Score							✓													✓		2
Entropy																	✓					1
Inertia									✓													1
Perplexity				✓	✓	✓			✓		✓				✓	✓				✓		7
ROUGE								✓														2

Table 2: Metrics of the automated evaluation used by recent papers on open-domain dialogue systems. The top row shows the reference numbers to the 20 surveyed papers. #: number of papers using the corresponding metrics.

	1	2	3	4	7	8	10	11	12	13	14	15	17	18	19	20	#	
Appropriateness						✓									✓			2
Coherence					✓								✓					2
Consistency	✓						✓				✓							3
Context Coherence								✓										1
Correctness		✓														✓		2
Diversity											✓							1
Emotion	✓																	1
Empathy									✓									1
Engagingness							✓											1
Fluency	✓	✓	✓			✓	✓	✓	✓				✓			✓		9
Grammaticality															✓			1
Humanness					✓													1
Informativeness							✓					✓	✓		✓			4
Knowledge Rel.		✓						✓									✓	3
Logic	✓																	1
Proactivity													✓					1
Quality													✓					2
Readability											✓							1
Relevance						✓			✓		✓							3
Sensibleness																✓		1
Specificity						✓										✓		2

Table 3: Dimensions of the human evaluation used by recent dialogue system papers. The top row shows the reference numbers to the 20 survey papers. [5, 6] do not perform any human evaluation; [9, 16] perform human evaluation without reference to dimensions. #: number of papers adopting the corresponding dimensions.

4.1 Dimensions of Human Evaluation

There is high variability in the dimensions of dialogue that previous studies have used for assessing dialogue systems in both static and interactive evaluations. Table 3 provides a detailed overview of the dimensions used by each of the surveyed papers when evaluating their work. There are a total of 21 uniquely-worded dimensions found; 11 of them appear in only a single paper. The resulting matrix provides clear evidence of the inconsistencies in human evaluation methods, as its sparsity is indicative of low overlap among those methods. The long tail distribution of the evaluation metrics makes it difficult for cross-work comparisons without a substantial study to align the disparate evaluation of one work with another.

Although the evaluation dimensions appear to be distinct on the surface, several of them appear to be similar in meaning. To analyze the level of overlap among the seemingly distinct evaluation dimensions, we compile the definitions and instructions shared by each of the papers regarding their evaluation dimensions and rating scales. Based on manual analysis, we are able to group dimensions together that are indeed evaluating the same aspect of dialogue as one another, even though the authors mention them by different names. Table 4 provides the dimension groupings that are identified on the basis of their respective definitions.

Definitions in Table 4a aim to address the grammaticality of system responses, including words like *grammar*, *understandable*, and *accurate*. As

Fluency	Whether the response from the listener is understandable (Lin et al., 2019)
	Whether the response is fluent and natural (Li et al., 2019)
	Whether each sentence has correct grammar (Luo et al., 2018)
	Fluency measures if the produced response itself is fluent (Wu et al., 2019):
Consistency	Whether the reply is fluent and grammatical (Li and Sun, 2018)
Readability	Whether the utterance is grammatically formed (Qiu et al., 2019)
Grammaticality	Whether the response is fluent and grammatical (Zhu et al., 2019)

(a) Grammatical Capability.

Relevance	Whether the responses of the listener seem appropriate to the conversation (Lin et al., 2019)
	Whether the response is appropriate/relevant in the current context language (Moghe et al., 2018)
	Whether the reply is relevant to the query (Qiu et al., 2019)
Appropriateness	Whether the response is appropriate in grammar, topic, and logic (Young et al., 2018)
Coherence	Whether the generated response is relevant to the input (Luo et al., 2018)
Context Coherence	Whether the whole dialogue is fluent (does not contain irrelevant or illogical responses) (Wu et al., 2019)
Logic	Whether the post and the reply are logically matched (Li and Sun, 2018)
Sensibleness	Whether the response makes sense given the context (Adiwardana et al., 2020)

(b) Turn Coherence.

Informativeness	Whether the response provides new information and knowledge in addition to the post (Young et al., 2018)
	Whether the response has unique words and multi-topic clauses (Tian et al., 2019)
	Whether the response has meaningful information relevant to its message (Zhu et al., 2019)
	Whether the model makes full use of knowledge in the response (Wu et al., 2019)
Specificity	Whether the model produced movie-specific responses or generic responses (Moghe et al., 2018)
Diversity	Whether the response is specific to the context (Adiwardana et al., 2020)

(c) Response Informativeness.

Table 4: Proposed reductions of dialogue evaluation dimensions into non-overlapping components

a result, the four dimensions recorded in this table can be viewed as lexical variations of the same underlying Grammaticality dimension. Similarly, definitions in Table 4b highlight keywords like *appropriate*, *relevant*, and *on-topic*, thus providing evidence that each of those dimensions are instances of the Relevance dimension. Finally, Table 4c has a high occurrence of information and diversity-focused definitions, and we can reduce the dimensions shown there to the single Informativeness dimension.

Other than these highly overlapping dimensions, Quality (Tian et al., 2019; Zhou et al., 2019) and Humanness (Moghe et al., 2018) can both be considered as the single Quality dimension, since they are used to elicit an overall quality assessment of the dialogue system responses. Similarly, Emotion (Li and Sun, 2018) and Empathy (Lin et al., 2019) can be reduced into the Emotional Understanding dimension that captures both the comprehension and production of emotional responses. The remaining two dialogue dimensions assess a unique quality of dialogue and are useful as independent dialogue dimensions:

- Engagingness: whether the response includes interesting content (Zhang et al., 2018)

- Proactivity: whether the response introduces new topics without breaking coherence (Wu et al., 2019)

Finally, two evaluation dimensions are specifically used for a subset of dialogue systems that incorporate knowledge:

- Correctness: was the response accurate based on the real-world knowledge (Liu et al., 2018; Wang et al., 2020)
- Knowledge Relevance: was the knowledge shared in the response appropriate to the context (Liu et al., 2018; Wang et al., 2020)

Knowledge Relevance is very similar to the previously discussed Relevance dimension, although it is specifically targeting an assessment of the appropriateness of the knowledge being used. Even more niche, the Correctness dimension is unique to knowledge-focused systems that seek to present only true factual information to the user; thus, such a dimension may not be useful in other contexts. Due to their targeted nature, these two dimensions may fall outside of the scope of a general, comprehensive, unified evaluation of dialogue systems, and instead be used for a targeted subgroup.

Dimension	Definition
Grammaticality	Responses are free of grammatical and semantic errors
Relevance	Responses are on-topic with the immediate dialogue history
Informativeness	Responses produce unique and non-generic information that is specific to the dialogue context
Emotional Understanding	Responses indicate an understanding of the user’s current emotional state and provide an appropriate emotional reaction based on the current dialogue context
Engagingness	Responses are engaging to user and fulfill the particular conversational goals implied by the user
Consistency	Responses do not produce information that contradicts other information known about the system
Proactivity	Responses actively and appropriately move the conversation along different topics
Quality	The overall quality of and satisfaction with the dialogue

Table 5: The final set of our proposed dialogue dimensions for human evaluation.

In total, after merging similar dimensions and discarding non-generalizable dimensions, a total of eight dimensions have been identified that share little to no definitional overlap and are reasonably applicable to all dialogue systems. Table 5 shows the finalized set of dialogue evaluation dimensions.

4.2 Diversities in Evaluation Metrics

Aside from the discrepancies in dialogue dimensions used for evaluation among different works, the actual procedure of evaluating these dialogue dimensions varies even further, particularly for static evaluations. A majority of work instructs human annotators to rate the dialogue system responses on a set of dialogue dimensions using numeric scales, where the scales being used are often different even between works that employ the same dialogue dimensions. For instance, one of the most commonly used dimension is the Fluency of the dialogue, with 9 out of the 16 papers in Table 3 have adopted this as an evaluation dimension. Between those 9 studies, Fluency ratings include scales of:

- 0~2: [Wu et al. \(2019\)](#); [Li et al. \(2019\)](#)
- 0~3: [Wang et al. \(2020\)](#); [Liu et al. \(2018\)](#)
- 1~5: [Moghe et al. \(2018\)](#); [Zhang et al. \(2018\)](#); [Lin et al. \(2019\)](#); [Madotto et al. \(2019\)](#)
- 1~10: [Luo et al. \(2018\)](#)

Furthermore, some studies use a preference metric for static evaluation in addition to - or even instead of - the numerical ratings ([Lin et al., 2019](#); [Young et al., 2018](#); [Du and Black, 2019](#); [Zhang et al., 2019](#)). In this case, human annotators are asked to select the most compelling response among many generated by multiple dialogue systems or even humans. Thus, preference metrics provide estimated ranking scores among different systems by measuring the percentage of times each system is preferred over the others.

Unlike the diversity in static evaluation, for the two papers, [Zhang et al. \(2018\)](#) and [Adiwardana et al. \(2020\)](#), employing interactive evaluation, only numerical ratings on specific dialogue dimensions are used as evaluation methods; other methods such as preference metrics are not used in either case.

4.3 Static vs Interactive Evaluations

Establishing the necessary assessment metrics is only one consideration to achieve an accurate dialogue evaluation. The other major consideration is the procedure underlying the evaluation. This section discusses the two human evaluation protocols, static and interactive evaluations, that have previously been used by many dialogue systems. Although both evaluation protocols overcome the deficiencies brought forth by automated evaluation through human judgment, interactive evaluation is hypothesized to be a more reliable assessment strategy than static one. What static evaluation offers above interactive evaluation is a lower cost in terms of time and labor. By removing the human annotator from the task of interacting with the dialogue system, and instead having them review a dialogue excerpt, the amount of work required is reduced.

However, this is simultaneously a point in favor of static evaluation, but also a factor as to why it is less reliable. As [Ghandeharioun et al. \(2019\)](#) suggest, chat-oriented dialogues have a less defined conversational goal which can best be summarized as being able to hold a “natural social interaction with humans”. The success - or failure - at this can only be evaluated by the targeted recipient of the conversation; namely, the user that the system is interacting with. External annotators, at best, can estimate the user’s satisfaction with the conversation based on their own projected opinions, which is not necessarily the most accurate assessment.

In addition, static evaluation is commonly conducted by producing a single system response in

OQ	GR	RE	IN	EU	EN	CO	PR
1	5.00 (± 0.00)	1.94 (± 0.98)	2.86 (± 1.29)	1.00 (± 0.00)	2.33 (± 0.89)	4.94 (± 0.23)	1.64 (± 0.87)
2	4.70 (± 0.47)	2.85 (± 0.88)	3.25 (± 1.25)	1.15 (± 0.37)	3.15 (± 0.75)	4.90 (± 0.31)	2.15 (± 0.59)
3	4.62 (± 0.51)	3.46 (± 0.52)	2.92 (± 0.86)	1.08 (± 0.28)	2.92 (± 0.49)	4.77 (± 0.44)	2.38 (± 0.65)
4	4.71 (± 0.46)	3.89 (± 0.42)	4.25 (± 0.70)	1.11 (± 0.31)	3.86 (± 0.36)	4.82 (± 0.39)	2.93 (± 0.54)
5	4.33 (± 0.58)	4.33 (± 0.58)	3.67 (± 0.58)	1.33 (± 0.58)	4.00 (± 0.00)	5.00 (± 0.00)	3.00 (± 0.00)

(a) The OQ column shows the overall quality ratings from our expert and the other columns show the average ratings from the expert on the corresponding dialogue dimensions.

OQ	GR	RE	IN	EU	EN	CO	PR
1	4.85 (± 0.37)	2.20 (± 1.20)	2.95 (± 1.28)	1.00 (± 0.00)	2.60 (± 1.05)	4.85 (± 0.37)	1.95 (± 0.94)
2	4.80 (± 0.41)	3.05 (± 1.10)	3.95 (± 1.19)	1.25 (± 0.44)	3.30 (± 0.92)	5.00 (± 0.00)	2.10 (± 0.79)
3	4.85 (± 0.37)	2.75 (± 1.07)	2.50 (± 0.95)	1.00 (± 0.00)	2.60 (± 0.75)	4.90 (± 0.31)	2.05 (± 0.89)
4	4.65 (± 0.49)	3.40 (± 0.82)	3.30 (± 0.92)	1.10 (± 0.31)	3.25 (± 0.79)	4.85 (± 0.37)	2.25 (± 0.72)
5	4.80 (± 0.41)	3.30 (± 1.13)	4.10 (± 0.97)	1.05 (± 0.22)	3.50 (± 0.76)	4.80 (± 0.41)	2.85 (± 0.75)

(b) The OQ column shows the overall quality ratings from the Alexa Prize and the other columns show the average ratings from the expert on the corresponding dialogue dimensions.

Table 6: The average ratings by our expert on each of the dialogue dimensions in Table 5 with respect to the overall ratings from the expert and the Alexa Prize. OQ: Quality, GR: Grammaticality, RE: Relevance, IN: Informativeness, EU: Emotional Understanding, EN: Engagingness, CO: Consistency, PR: Proactivity.

a fixed dialogue context. This fails to reveal certain system deficiencies, such as repetitiveness, inconsistency, and lack of long-term memory of the information shared in the conversation. It also prevents an assessment of the system’s error-handling or misunderstanding recovery capabilities from being encountered. All of these aspects are necessary to truly assess the quality of dialogues that a given dialogue system can produce. Without this information, only a biased perspective can be achieved, and the evaluation will not reflect the true capability of the system if it were to be used in practice.

5 Case Study: Alexa Prize 2020

This section presents a case study of the significance of the proposed dialogue dimensions in Table 5 using real human-machine dialogues. For this analysis, 100 rated conversations were taken from the Alexa Prize Socialbot Grand Challenge 3², which is a university competition to create innovative open-domain chatbots (Ram et al., 2018). During the competition, conversations are rated in terms of Overall Quality on a scale of 1 (worst) to 5 (best) under the interactive evaluation protocol. For this case study, we sampled conversations with an equal distribution between all ratings, where every conversation has at least three turns to ensure sufficient content.

Because only the Overall Quality dimension is provided from the interactive evaluation, we also conducted an expert analysis on the same conversations in order to explore the implications of

the other previously identified dialogue dimensions. To this end, one of the authors - who has over three years of experience in dialogue system research - manually rated the conversations on each of the dialogue dimensions in Table 5.

It is worth mentioning that the following findings are taken as only a preliminary analysis, strongly considering the low agreement between the expert and interactive evaluations on OQ, which will be discussed shortly (Section 5.2). This disparity between the expert and human user evaluations renders it difficult to convey a convincing conclusion regarding the significance of the evaluation dimensions. However, we hope this work begins the momentum to investigate the importance of such evaluation dimensions in overall human perception of dialogue quality.

5.1 Quality vs. Other Dialogue Dimensions

Table 6 shows the average rating and its standard deviation on each of the 7 dialogue dimensions (GR, RE, IN, EU, EN, CO, PR) across the overall quality ratings (OQ). All ratings on those 7 dimensions are assessed by our expert. OQ ratings are provided by the expert for Tables 6a and the human users from the Alexa Prize for Table 6b.

Relevance & Proactivity The clearest positive relationship to OQ is observed from RE and PR, especially from the expert evaluation although it can be seen in the interactive evaluation as well. This suggests that these dimensions are pertinent to the human perception of dialogue quality, and that this relationship is even more apparent when eval-

²<https://developer.amazon.com/alexaprize>

ators are given the opportunity to review previous dialogue turns when determining OQ.

Informativeness & Engagingness The relationship between IN and EN to OQ is not as obvious as the previous two dimensions, RE and PR, although an indication of a positive relationship is observed.

Grammaticality Due to the manual curation of responses in our Alexa Prize chatbot, we have tight control over the grammaticality of our responses; thus, the overall variance in GR is low. Interestingly, we do notice that there is a slight inverse relationship between GR and OQ. Although this may seem counter-intuitive, the likely explanation is that conversations with higher OQ tend to be longer so that they comprise a greater number of topics and, as more topics are introduced, the chance for an (accidentally) ungrammatical response to be revealed is higher. Nonetheless, it appears that ungrammaticality is not a strict deterrent on OQ.

Emotional Understanding & Consistency The effect of EU and CO on OQ is inconclusive from the presented analysis. This is attributed to the low variation in these dimensions of our chatbot, as we can enforce the consistency of responses and do not aim to tackle emotional understanding.

5.2 Expert vs. Interactive Evaluations

The inter-annotator agreement between the OQ ratings of the expert and the users from the Alexa Prize is provided in Table 7. The agreement is measured for both fine-grained ratings that consider all scales (1 - 5) and coarse-grained ratings that consider only two scales (low: 1 - 2, high: 3 - 5). Although the inter-annotator agreement is higher for the coarse-grained ratings, it is apparent that the agreement scores are dramatically low for both.

Rating Type	Agreement
Fine-grained	0.13
Coarse-grained	0.22

Table 7: Cohen’s Kappa scores on the overall quality ratings between the expert and interactive evaluation.

Table 8 shows that the expert evaluation tends to be more punishing overall, with a much fewer number of conversations receiving a 5.0 rating. Indeed, 56% of the conversations from the expert evaluation would be categorized as a low rating, whereas the interactive evaluation has only 40%. Even so, the low agreement indicates that the quality as-

sessments across the two evaluation protocols are highly variable across the same conversations.

OQ	1	2	3	4	5	Σ
Interactive	20	20	20	20	20	100
Expert	36	20	13	28	3	100

Table 8: Comparison of the rating distribution between expert and interactive evaluation

This provides preliminary support for the hypothesis in Section 4 that external evaluators are unable to accurately infer the same impression of a conversation as that of the user who is actually participating in the conversation. Although there are potential methods which aim to mitigate this effect - such as agglomerate ratings across more than one external annotator - the underlying cause of such variance may be attributed to the poor suitability of external evaluations for dialogue system evaluation as a whole, but further work is required.

6 Conclusion and Future Work

In this paper, we provide an extensive background and the current states on the three types of dialogue system evaluation protocols, automated, static, and interactive. Our analysis shows that static evaluation is the dominating human evaluation used in the most recent dialogue system works, although it has several concerning limitations, some of which are exemplified through our case study. We propose a set of eight dialogue dimensions that encapsulate the evaluations of previous studies without redundancy. As a result of our case study, we find preliminary evidence that the dimensions of relevance, proactivity, informativeness, and engagingness are likely to be contributing factors to the overall perception of dialogue quality.

Our future work will build upon these findings to develop a thorough understanding of the necessary dialogue dimensions for comprehensive interactive evaluation of dialogue systems. Through an analysis based on large-scale user studies, we look to propose an evaluation protocol that captures the human judgement of dialogue quality through precise formulation of evaluation dimensions, in order to enable targeted dialogue system advancements.

Acknowledgments

We gratefully acknowledge the support of the Alexa Prize Socialbot Grand Challenge 3. Any contents in this material are those of the authors and do not necessarily reflect the views of the Alexa Prize.

References

- Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. *Towards a Human-like Open-Domain Chatbot*. *arXiv preprint arXiv:2001.09977*.
- Stanley F Chen, Douglas Beeferman, and Roni Rosenfeld. 1998. Evaluation metrics for language models. In *DARPA Broadcast News Transcription and Understanding Workshop (BNTUW)*.
- Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2019. Survey on evaluation methods for dialogue systems. *arXiv preprint arXiv:1905.04071*.
- Wenchao Du and Alan W Black. 2019. *Boosting Dialog Response Generation*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 38–43, Florence, Italy. Association for Computational Linguistics.
- Jianfeng Gao, Michel Galley, Lihong Li, et al. 2019. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298.
- Asma Ghandeharioun, Judy Hanwen Shen, Natasha Jaques, Craig Ferguson, Noah Jones, Agata Lapedriza, and Rosalind Picard. 2019. *Approximating Interactive Human Evaluation with Self-Play for Open-Domain Dialog Systems*. In *Advances in Neural Information Processing Systems 32*, pages 13658–13669. Curran Associates, Inc.
- Sarik Ghazarian, Johnny Wei, Aram Galstyan, and Nanyun Peng. 2019. *Better Automatic Evaluation of Open-Domain Dialogue Systems with Contextualized Embeddings*. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 82–89, Minneapolis, Minnesota. Association for Computational Linguistics.
- Tatsunori Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1689–1701.
- Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2019. Challenges in building intelligent open-domain dialog systems. *arXiv preprint arXiv:1905.05709*.
- Jingyuan Li and Xiao Sun. 2018. *A Syntactically Constrained Bidirectional-Asynchronous Approach for Emotional Conversation Generation*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 678–683, Brussels, Belgium. Association for Computational Linguistics.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.
- Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019. *Incremental Transformer with Deliberation Decoder for Document Grounded Conversations*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21, Florence, Italy. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. *ROUGE: A Package for Automatic Evaluation of Summaries*. In *Proceedings of the Workshop on Text Summarization Branches Out*, pages 56–60, Barcelona, Spain. Association for Computational Linguistics.
- Zhaojiang Lin, Andrea Madotto, Jamin Shin, Peng Xu, and Pascale Fung. 2019. *MoEL: Mixture of Empathetic Listeners*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 121–132, Hong Kong, China. Association for Computational Linguistics.
- Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. *How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation*. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.
- Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. *Knowledge Diffusion for Neural Dialogue Generation*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1498, Melbourne, Australia. Association for Computational Linguistics.
- Liangchen Luo, Jingjing Xu, Junyang Lin, Qi Zeng, and Xu Sun. 2018. *An Auto-Encoder Matching Model for Learning Utterance-Level Semantic Dependency in Dialogue Generation*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 702–707, Brussels, Belgium. Association for Computational Linguistics.
- Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. *Personalizing Dialogue Agents via Meta-Learning*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459, Florence, Italy. Association for Computational Linguistics.

- Andrei Malchanau, Volha Petukhova, and Harry Bunt. 2019. **Multimodal Dialogue System Evaluation: A Case Study Applying Usability Standards**. In *9th International Workshop on Spoken Dialogue System Technology*, volume 579, pages 145–159. Springer Singapore, Singapore.
- Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. **Towards Exploiting Background Knowledge for Building Conversation Systems**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332, Brussels, Belgium. Association for Computational Linguistics.
- Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3349–3358.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. **Bleu: a Method for Automatic Evaluation of Machine Translation**. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Prasanna Parthasarathi and Joelle Pineau. 2018. **Extending Neural Generative Conversational Model using External Knowledge Sources**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 690–695, Brussels, Belgium. Association for Computational Linguistics.
- Lisong Qiu, Juntao Li, Wei Bi, Dongyan Zhao, and Rui Yan. 2019. **Are Training Samples Correlated? Learning to Generate Dialogue Responses with Multiple References**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3826–3835, Florence, Italy. Association for Computational Linguistics.
- Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*.
- Zhiliang Tian, Wei Bi, Xiaopeng Li, and Nevin L. Zhang. 2019. **Learning to Abstract for Memory-augmented Conversational Response Generation**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3816–3825, Florence, Italy. Association for Computational Linguistics.
- Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. **PARADISE: A framework for evaluating spoken dialogue agents**. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 271–280, Madrid, Spain. Association for Computational Linguistics.
- Jian Wang, Junhao Liu, Wei Bi, Xiaojiang Liu, Kejing He, Ruiyang Xu, and Min Yang. 2020. **Improving Knowledge-aware Dialogue Generation via Knowledge Base Question Answering**. *arXiv preprint arXiv:1912.07491*.
- Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. **Proactive Human-Machine Conversation with Explicit Conversation Goal**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, Florence, Italy. Association for Computational Linguistics.
- Xinnuo Xu, Ondřej Dušek, Ioannis Konstas, and Verena Rieser. 2018. **Better Conversations by Modeling, Filtering, and Optimizing for Coherence and Diversity**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3981–3991, Brussels, Belgium. Association for Computational Linguistics.
- Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. **Augmenting End-to-End Dialogue Systems With Commonsense Knowledge**. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 4970–4977.
- Hainan Zhang, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. 2019. **ReCoSa: Detecting the Relevant Contexts with Self-Attention for Multi-turn Dialogue Generation**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3721–3730, Florence, Italy. Association for Computational Linguistics.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. **Personalizing Dialogue Agents: I have a dog, do you have pets too?**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.
- Kun Zhou, Kai Zhang, Yu Wu, Shujie Liu, and Jingsong Yu. 2019. **Unsupervised Context Rewriting for Open Domain Conversation**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1834–1844, Hong Kong, China. Association for Computational Linguistics.
- Qingfu Zhu, Lei Cui, Wei-Nan Zhang, Furu Wei, and Ting Liu. 2019. **Retrieval-Enhanced Adversarial Training for Neural Response Generation**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3763–3773, Florence, Italy. Association for Computational Linguistics.

Human-Human Health Coaching via Text Messages: Corpus, Annotation, and Analysis

Itika Gupta,¹ Barbara Di Eugenio,¹ Brian Ziebart,¹ Aiswarya Baiju,¹ Bing Liu,¹ Ben S. Gerber,² Lisa K. Sharp,³ Nadia Nabulsi,³ Mary H. Smart³

¹Department of Computer Science

²Department of Medicine

³Department of Pharmacy Systems, Outcomes, and Policy

University of Illinois at Chicago, Chicago, Illinois

{igupta5, bdieugen, bziebart, abaiju2, liub}@uic.edu
{bgerber, sharpl, nnabul2, msmart5}@uic.edu

Abstract

Our goal is to develop and deploy a virtual assistant health coach that can help patients set realistic physical activity goals and live a more active lifestyle. Since there is no publicly shared dataset of health coaching dialogues, the first phase of our research focused on data collection. We hired a certified health coach and 28 patients to collect the first round of human-human health coaching interaction which took place via text messages. This resulted in 2853 messages. The data collection phase was followed by conversation analysis to gain insight into the way information exchange takes place between a health coach and a patient. This was formalized using two annotation schemas: one that focuses on the goals the patient is setting and another that models the higher-level structure of the interactions. In this paper, we discuss these schemas and briefly talk about their application for automatically extracting activity goals and annotating the second round of data, collected with different health coaches and patients. Given the resource-intensive nature of data annotation, successfully annotating a new dataset automatically is key to answer the need for high quality, large datasets.

1 Introduction

A sedentary lifestyle significantly increases the risk of numerous diseases such as type 2 diabetes, cardiovascular disease, and depression (Booth et al., 2017). Unfortunately, physical inactivity has progressively increased over the past several decades. It can be attributed to using modes of transportation for short distances, labor-saving devices, and less active occupations among various other reasons. However, the underlying problem is a lack of motivation. Successfully implementing healthy behaviors require significant motivation that most people, individually, find difficult to initiate and

maintain (Cerin et al., 2010; Poncela-Casasnovas et al., 2015). Health coaching (HC) has been identified as a successful method for facilitating health behavior changes by having a professional provide evidence-based interventions, support for setting realistic goals, and encouragement for goal adherence (Kivelä et al., 2014). But HC has its limitations such as it is expensive, time-intensive, and not available around the clock.

Therefore, we aim to build a dialogue-based virtual assistant health coach that will converse with the patients via text messages and help them to set Specific, Measurable, Attainable, Realistic and Time-bound (S.M.A.R.T.) goals (Doran, 1981). The SMART goal-setting approach has been rigorously adopted to set realistic and manageable goals in different fields such as health behavior change and software engineering. It has been shown that goal setting and action planning help patients adopt healthy behaviors and manage chronic diseases (Bodenheimer et al., 2007; Handley et al., 2006). Also, text messages have been shown to help patients follow healthy behaviors as they provide a continuous means of education, support, and motivation (Chow et al., 2015); currently, a majority of the population owns a cellphone (96% are cellphone users and 81% are smartphone users¹).

Most goal-oriented dialogue systems assume that a user has a predefined goal that needs to be accomplished using the system. However, that is not the case in HC dialogues. Instead of the usual information-seeking dialogues, where the user requests information from the system, HC dialogues are collaborative where both the coach and the patient negotiate a goal that best suits the patient's lifestyle and seems realistic based on their previous activity patterns. An excerpt from dataset 1 is shown in Figure 1. The patient starts with an

¹<https://www.pewresearch.org/internet/fact-sheet/mobile/>

(1) Patient: Good morning, my goal is to aim for 30,000 steps, 40 flights, 12000 calories and 12 miles by the end of Friday this week
(2) Coach: Wow that's a lot.
(3) Coach: Last week you did 38 flights. Do you know how many step you got?
(4) Coach: Ok i just calculated roughly 23k for Mon to fri. You had more over the weekend.
(5) Coach: Those are great personal goals to have. Let's focus on the walking goals for the purpose of the study. So how likely do you think you will be able to accomplish your goal of 30K steps and 40 flights?
(6) Patient: Well considering it will be measured Monday through Friday I guess I should reduce my goals.ill aim for 20,000 steps and 30 flights. I feel I will be more
(7) Patient: likely to accomplish this goal without any problems.

Figure 1: Example of a conversation between the health coach and a patient

ambitious goal in (1), and the coach helps to make it more realistic through (2)-(5). As the conversation takes place over the text messages, a dialogue system will also need to take care of abbreviations and typing errors such as ‘goals.ill’ in (6).

Moreover, most existing dialogue datasets do not involve any follow-up conversations. For instance, once a flight is booked, the system doesn’t follow-up on how the trip was or if the user would like to modify the booking. However, it is a crucial step in HC conversations as patients tend to change their goals on encountering a barrier. Lastly, HC conversations happen over multiple days. Some days no messages are exchanged and some days more than 10 messages get exchanged. Most publicly available datasets assume that the task will be finished in one sitting. Due to this collaborative negotiation setting over multiple days in our corpus, goal information is spread throughout the dialogue.

Motivated by these complexities, we decided to annotate our data for two types of information: (1) the SMART goal attributes in the dialogues to track patients’ goals, and (2) different stages and phases that model the conversation flow in HC dialogues. For our domain, SMART goal attributes are the slot-values pertaining to a patient’s goal. Stages and phases are more abstract, but otherwise analogous to tasks and sub-tasks as defined in task-oriented dialogue systems ([Chotimongkol and Rudnicky, 2008](#)). We believe the SMART annotation schema that we designed can be applied to any task where SMART goal setting is being used and not just physical activity. Similarly, the stages-phases annotation schema can be used to model the flow of any collaborative decision making counseling dialogue. In this paper, we will discuss the two rounds of data collection process, the subsequent analysis of the dialogues, which includes developing schemas and annotating the data, and application of models trained on these annotations.

Our contributions can be summarized as follows:

- We describe the data collection methodology for health coaching dialogues via text messages that span over multiple days. We undertook two rounds of data collection; we discuss what we learned in round 1 and what this led us to change in round 2. We will refer to the first round of data as *dataset 1* and the second round of data as *dataset 2* throughout the paper.
- We believe we are the first to formalize the SMART goal-setting approach, which we did based on dataset 1 using two annotation schemas. We demonstrate that this approach results in reliable annotator agreement.
- We show that supervised classification models trained on dataset 1 can be used to automatically extract goals and reliably annotate dataset 2 for SMART tags (macro F-score = 0.81) even though the latter was collected with 3 different health coaches and 30 different patients.
- We will release dataset 2 to the community, since we collected consent from the patients in this regard². Dataset 2 will be available upon request along with the annotation manual. Given the nature of the dataset, out of an abundance of respect for our patients, the text data will not be made public online.

2 Related Work

One cannot build a good domain-specific dialogue system without having any insights into how users will interact with the system. Therefore, first we need data that represents at least some range of actions that are found in human-human or human-machine conversations in the given domain. Initiatives such as the Dialogue State Tracking Challenge (DSTC) started in 2013 to provide a common testbed for different tasks related to domain-specific dialogue systems such as dialogue state

²Unfortunately, the activity data collected via Fitbit cannot be shared, since consent did not include permission for such data; dataset 1 cannot be shared, because of lack of consent.

tracking, dialogue act prediction, and response generation; labeled datasets for each of these tasks were provided (Williams et al., 2013). However, most of these datasets focused on traveling and restaurant booking domains (Henderson et al., 2014). Moreover, for data collection, predefined scenarios are given to the users and thus, the users' responses are not as spontaneous as they would be in a real-life situation (Asri et al., 2017; Budzianowski et al., 2018). Unfortunately, there are no such publicly available datasets for dialogue systems in the health domain.

The idea of automated conversational agents to promote healthy behaviors has recently gained considerable interest. Researchers such as Watson et al. (2012) and Both et al. (2010) respectively worked on promoting physical activity adherence and supporting psychotherapy for adults using automated systems. But internally most of these systems rely on a predefined set of input/output mappings, focus more on general goal setting, and do not provide follow-up during goal accomplishment.

Researchers have also focused on computational analysis of conversations in the health domain. Pérez-Rosas et al. (2018) collected Motivational Interviewing (MI) based counseling interviews from public sources such as YouTube and built models to predict the overall counseling quality using linguistic features. Before the YouTube data, the authors also worked on data collected in clinical settings, graduate student training and such, but didn't release it due to privacy reasons (Pérez-Rosas et al., 2016). The authors used the well established Motivational Interviewing Treatment Integrity (MITI) coding system to annotate the data and score how well or poorly a clinician used MI (Moyers et al., 2016). The MITI coding system was also used by Guntakandla and Nielsen (2018) to annotate reflections in the health behavior change therapy conversations. Since MI based interventions focus on understanding patient's attitudes towards the problem and persuading them to change, the MITI coding system supports assessing clinicians based on how well they bring forth patient's experiences, cultivate change talk, provide education, persuade them through logical arguments, and such. However, specific goal setting is not the main focus of these interviews and is rarely discussed.

A framework for health counseling dialogue systems closest to ours is by Bickmore et al. (2011). Their task model comprises opening, small talk,

review tasks, assess, counseling, assign task, pre-closing, and closing. Conversely, our stages-phases schema looks at the fine-grained decomposition of review-tasks, counseling, and assign task, which Bickmore et al. (2011) did not do. As far as we know, no other work models HC dialogues collected in a SMART goal setting, focusing on slot-values and higher-level conversation flow.

3 SMART Goal Setting

Based on the domain, practitioners modify the definition of SMART components to fit the task at hand. For physical activity, we define them as follows:

- **Specific (S):** Create a clear goal that is as specific as possible and focuses on a particular activity or task such as cycling, walking, or taking stairs.
- **Measurable (M):** Quantify the goal to know when the goal has been accomplished.
- **Attainable (A):** The goal should be attainable given the current situation such as workload and family responsibilities. The person should feel confident towards accomplishing the goal.
- **Realistic (R):** Set goals that are not too easy, but at the same time are not too hard. The goal should appear like a challenge but still be realistic. In other words, it should be more challenging than the current average, but not too far off.
- **Time-Bound (T):** Set an upper-bound time by which you want to achieve the goal. It is the higher level measurable component that is not set regularly but instead is an overall time frame.

An example of a well-defined SMART goal is, *I will walk 5000 steps three days a week on Monday, Wednesday, and Friday for 2 weeks*, where walk is a specific activity; 5000 steps and 3 days are measurable quantities; 2 weeks is the total time frame. As concerns attainability and realism, they are not immediately available from this goal statement and will depend on the person's circumstances. On the other hand, *I will start walking more* is a poorly defined, vague, and unquantified goal and is not likely to lead to success.

4 Data Collection and Analysis

Dataset 1: We recruited 28 patients between the ages of 21 to 65 years who were interested in increasing their physical activity at our university's internal medicine clinic. A health coach, trained in SMART goal setting, conversed with the patients to set goals every week for four weeks via

	1	2	3	4	5	6	7	8
Dataset 1								
C	19.6	13.8	12.3	11.8		-NA-		
P	15.3	12.0	10.7	11.0		-NA-		
T	34.9	25.8	23.0	22.8		-NA-		
Dataset 2								
C	14.9	14.2	11.1	11.3	8.7	9.5	9.2	9.2
P	11.0	10.9	7.8	7.1	5.6	6.9	6.6	6.7
T	25.9	25.1	18.9	18.4	14.3	16.4	15.8	15.9

Table 1: Average number of messages per week

T: total, C: Coach, P: Patient

text messages. Each week wasn't necessarily 7 days as sometimes patients took longer to set a goal, which made some weeks shorter like 5-6 days and some longer like 8-9 days. Patients used their smartphones and texting service to communicate with the coach. The coach used a web application named Mytapp, developed by Dr. Ben Gerber, to send texts to the patients. The application has been used to conduct other text-based health monitoring studies (Stolley et al., 2015; Kitsiou et al., 2017). Mytapp is a two-way text messaging application that was designed to help promote healthy behaviors and manage chronic diseases. The main benefit of using it over a normal texting service is the privacy of data. All data is encrypted and exchanged using transport layer security. The messages were saved in a secured database and the application stored minimum information about the patients.

The patients were also given Fitbits to monitor their progress. The coach monitored patients' progress using the Mytapp application, as it can fetch the most up-to-date activity data from a patient's Fitbit account and show it at one place along with text messages. This reduces the workload for the coach as at any point in time during the study the coach had at least 3 patients and would have had to login into their respective accounts to access the Fitbit data without the application. The coach needed all this information to help patients set realistic goals based on previous weeks' performance.

The HC conversations involved setting a specific, measurable and realistic goal, and solving any barriers to goal attainment. The coach sent reminders based on patients' preferences and provided motivational feedback on their progress. Out of 28 patients, only one did not finish the study due to health problems. Therefore, we only considered 27 patients' data for analysis and building models. Dataset 1 comprises 2853 messages, where 54% of messages were sent by the coach and 46% by the patients. This tells us that both the coach and

the patients were equally involved. An excerpt was shown earlier in Figure 1.

Lessons from dataset 1 collection: During the initial face-to-face recruitment process at the university clinic, patients were given information about the study and the concept of SMART goal setting was explained to them. To help them understand it clearly, the goal for the first week was sometimes discussed during that initial interaction. Hence, we found that portions of the initial goal setting conversation may have been missing from the text messages, including: the patient's goal for the first week, discussion that led to that goal, and any time preferences for the text messages. Therefore, during dataset 2 collection, we asked the recruiters to take notes about what was discussed face-to-face, and asked health coaches to reiterate the first goal in text messages even if it was already known. In cases where patients didn't have information about their current activity level, a goal of one mile (2000 steps) a day was suggested. The recruiters also helped patients with setting up Fitbit trackers, downloading the Fitbit app, and linking the two together during the initial recruitment process (same as dataset 1). However, based on dataset 1 collection, recruiters had a better understanding of the issues that might arise with Fitbit and also met the patients again during the study (if possible) to fix the issues. Lastly, during the dataset 2 collection, we also collected audio-recorded feedback at the end of the study if the patients came back to the clinic, else feedback was taken over a phone call and notes were recorded.

Dataset 2: We recruited three different individuals trained in SMART goal setting to be health coaches. We also recruited 30 different patients and conducted the study for eight weeks instead of four to analyze changes in messaging behavior over a longer period. The same Mytapp application was used to text the patients and Fitbits were given to the patients. Out of 30 patients, one patient withdrew after 5 weeks, one lost their Fitbit after 2 weeks, and one set goals for only 2 weeks and then almost stopped responding. Since the latter two were in the study for fewer than 4 weeks, we only consider the data from 28 patients. We also removed all the messages discussing an appointment time for the exit interview, which comprises more than 600 messages. This resulted in 4134 messages among which 58% were sent by coaches and 42% by the patients. Dataset 1 only included about 30

Stage	Phase	Description	Phase Boundary Examples
Goal Setting	Identification	during the beginning of the week when the coach asks the patients about their goal or when the patients inform their goal to the coach	Coach: Now what goal could you make that would allow you to do more walking?
	Refining	when the coach asks (or the patient informs) the specifics of the goal such as time, location, frequency to make the goal more effective	Coach: what time do you plan to do so I can set up a reminder?
	Anticipate Barrier	when the coach asks the patients (or the patients specify) their confidence in achieving the goal (range 1-10) or if they see any upcoming barriers	Coach: Do you think the weather will make it hard for you to take 50 min walks everyday this week?
	Solve Barrier	when the coach tries to help patients overcome a barrier or increase their attainability score to 10 without modifying the quantity	Coach: what do you think will make it easy to accomplish/achieve your goal?
	Negotiation	when the patient chooses a goal that the coach thinks might be too much/less or vice-versa	Coach: another 8 . What if you were to try for 8000 steps again this week would the answer be a 10?
Goal Implementation	Refining	same as the previous stage; here it usually follows solve barrier or goal negotiation phase to make the goal more specific	Coach: Have you decided when you would like to get your walk in?
	Anticipate Barrier	similar to the previous stage, but here it indicates the barrier that has been encountered	Patient: Good morning [NAME]. I probably won't be able to make my goal this week. I'm at a professional development all day today and there are no stairs in this building
	Solve Barrier	same as the previous stage	Coach: Do you want to try to make your goal over the weekend?
	Negotiation	when the patient is unable to accomplish the goal or wants to do more, the coach or the patient asks to modify the goal	Patient: Please change my safety goal to three days per week.
	Follow up	when the coach asks the patient (or patients themselves inform) about their progress and if they can accomplish the goal	Coach: Good afternoon! How is your goal for this week going so far?

Table 2: Stages and phases schema description with examples

messages in total from 2 patients regarding appointment. So we didn't eliminate them.

Table 1 shows the average number of messages exchanged weekly, where a week corresponds to the patient's goal. There is a decrease in the number of messages over the weeks. This is because during the first week the coach sometimes redefines what a SMART goal is and also explicitly asks the patients to specify details such as which day, what time, and how much. However, as the study progresses, the answers to some of these questions such as time and days are implicitly understood to be the same as in the previous weeks if not stated otherwise and only the amount of activity is modified. Dataset 2 was collected two years after dataset 1 and hence the schemas and models were built using dataset 1 exclusively without any bias from dataset 2.

5 Annotation of the Coaching Dialogues

In this section, we will look at the two types of annotations: SMART goal annotations and stages-phases annotations. Since no work exists that has used SMART criteria to set physical activity goals via SMS, we designed the schemas that were in-

spired by the literature on goal setting (Bodenheimer et al., 2007; Bovend'Eerdt et al., 2009). We used the General Architecture for Text Engineering (GATE) tool for annotations (Cunningham, 2002).

Stages and Phases Annotation Schema: 15 patient-coach conversations from dataset 1 were used to design stages-phases schema. This annotation aims to understand how the conversation unfolds in HC dialogues. Stages and phases respectively help to capture the coaching tasks and sub-tasks being performed throughout the communication dialogue. The annotation schema along with descriptions is shown in Table 2. The higher tier is composed of stages; Goal Setting (GS) and Goal Implementation (GI). Stages are composed of phases. The GS stage consists of identification, refining, negotiation, anticipate barrier, and solve barrier. The GI stage consists of the same phases plus an additional follow up phase and minus the identification phase. We annotated the first message that indicated a change in a phase and all the messages after that are assumed to belong to that phase until there is a change in phase. Each message belongs to only one stage-phase. A snippet of

Stage: Goal Setting

Phase: Goal Identification

Coach: What would you like to set as your SMART goal this week?

Patient: Smart goal 12k steps a day?

Phase: Goal Negotiation

Coach: Ok, just something to think about... You got 12K steps 3 out of 7 days in the last week. That was Saturday, Sunday and Monday. How many days out the week do you want to do 12K step? Everyday?

Patient: Let's do 15K

Coach: That's more

Patient: 12k TU ,W, TH

Coach: Are you sure? If you think 12K everyday is realistic for you , go for it!

Patient: It's a challenge I'll try

Coach: Let's keep it at Tue, Wed. and Thurs then.

Patient: Ok

Phase: Solve Barrier

Coach: what do you think will make it easy to accomplish/achieve your goal?

Patient: Use stairs more and less elevator

Phase: Anticipate Barrier

Coach: On a scale of 1-10 with 10 being very sure and 1 not at all sure. How sure are you that you will accomplish your goal?

Patient: 8

Coach: What do you think will make it difficult?

Patient: Not being able to walk during my lunch hours because it's busy at work. So Time.

Phase: Goal Negotiation

Coach: I see maybe you should pick weekend days. That's when you have been most active according to fitbit

Coach: Last Sat and Sunday you got well over 12K steps

Coach: or maybe cut down on the amount of steps on those days. How can you change your goal to make that a 10 on the scale?

Patient: Ok. 12k on weekends

Coach: Sounds great good luck!!

Stage: Goal Implementation

Phase: Follow up

Coach: Good morning! How is your goal for this week going so far?

Patient: Good morning. It's going great

Figure 2: Example showing usage of stages and phases annotation schema

Stage	Phase	Message Count	Boundary Count
Goal Setting	Identification	408	109
	Refining	344	85
	Anticipate Barrier	363	82
	Solve Barrier	158	52
	Negotiation	92	19
Goal Implementation	Refining	16	4
	Anticipate Barrier	8	4
	Solve Barrier	25	7
	Negotiation	23	6
	Follow up	1348	120

Table 3: Stage-phase tags. Number of: messages in given stage-phase ('Message count'); dialogue transitions into given stage-phase ('Boundary count').

an annotated conversation is shown in Figure 2.

Two annotators annotated four previously unseen patients' data for stages and phases (447 messages). Inter Annotator Agreement (IAA) was measured using Cohen's kappa coefficient (κ) (Cohen, 1960); we obtained an excellent $\kappa=0.93$. This may

	gs_I	gs_R	gs_N	gs_AB	gs_SB	gi_R	gi_N	gi_AB	gi_SB	gi_F	Stop
Start	1	0	0	0	0	0	0	0	0	0	0
gs_I	0	0.63	0.04	0.17	0.08	0	0	0	0	0.08	0
gs_R	0.01	0	0.06	0.38	0.2	0	0	0	0	0.35	0
gs_N	0	0.05	0	0.21	0.16	0	0	0	0	0.58	0
gs_AB	0	0.05	0.1	0	0.28	0	0	0	0	0.57	0
gs_SB	0.02	0.21	0.04	0.54	0	0	0	0	0	0.19	0
gi_R	0	0	0	0	0	0	0.25	0	0	0.75	0
gi_N	0	0	0	0	0	0.33	0	0	0	0.67	0
gi_AB	0	0	0	0	0	0	0.5	0	0.25	0.25	0
gi_SB	0	0	0	0	0	0.14	0	0	0	0.71	0.14
gi_F	0	0	0	0	0	0.01	0.02	0.03	0.05	0	0.88

Figure 3: Transition probabilities from one stage-phase to another [gs: goal setting, gi: goal implementation, I: identification, R: refining, N: negotiation, AB: anticipate barrier, SB: solve barrier, F: follow-up]

be partially due to the stages-phases being bound to occur in a particular sequence: our HC conversations follow a particular structure, which involves phases such as identification, refining, and negotiation. Therefore, we analyzed the HC conversations as concerns likely transitions, and their frequencies.

First, Table 3 shows the counts for stage-phase

Tag	Feature	Description	Slot Example	Intent Example
Specificity	activity	the activity that will be done by the patient	Patient: Ok. I'll walking the stairs in the mornings from 8 to 10 Monday - Friday	
	time	the time of the day when the patient will be doing the activity	Patient: Ok. I'll walking the stairs in the mornings from 8 to 10 Monday - Friday	Coach: like how many days next week and at what time of day?
	location	the location where the patient will be doing the activity	Patient: I can also plan to walk the stairs at home . After work	Coach: Could you maybe get your steps done in the house?
Measurability	quantity (amount/ distance/ duration)	quantifies the activity in some way to show what patients are planning to accomplish. It can be number of steps or stairs, distance or duration	Patient: Yes, I'm going for 6000 step (amount) Patient: I will walk 3 blocks (distance) Patient: I do 40 min of walk (duration)	Coach: How many would you like to try for?
	days (name/ number)	the number of days or the name of the days the patient will be working on the chosen activity	Patient: Ok. I'll walking the stairs in the mornings from 8 to 10 Monday - Friday (days-name) Patient: I will try 3 days. (days-number)	Coach: Will you walk 4 block on the same days Mon, Wed and Fri? (days-name) Coach: like how many days next week and at what time of day? (days-number)
	repetition	the number of times the activity will be done in the same day	Patient: I will attempt to spend 15 mins 3times a day walking up and down two flights of stairs. 8am	
Attainability	score	specifies how confident a patient is about accomplishing the goal on a scale of 1-10	Patient: 8	Coach: On a scale of 1-10 with 10 being very sure and 1 not at all sure. How sure are you that you will accomplish your goal?
Realism		helps to indicate statements that judge the realism of a goal for the patient. It is usually based on their previous performance		Coach: Sounds like a very doable goal you are averaging over 9k steps during the week-days, now

Table 4: SMART annotation schema description with examples

Coach: What goal could you make that would allow you to do more walking?
Patient: Maybe <u>walk</u> (<i>S.activity</i>) more <u>in the evening after work</u> (<i>S.time</i>).
Coach: Ok sounds good. <u>How many days after work</u> (<i>S.time</i>) <u>would you like to walk</u> (<i>S.activity</i>)?
<small>M.days.number.intent</small>
Coach: <u>And which days would be best?</u>
<small>M.days.name.intent</small>
Patient: <u>2 days</u> (<i>M.days.number</i>). <u>Thursday</u> (<i>M.days.name</i>), maybe <u>Tuesday</u> (<i>M.days.name.update</i>)
Coach: <u>Think about how much walking</u> (<i>S.activity</i>) <u>you like to do for example 2 block</u> (<i>M.quantity.distance.other</i>)
<small>M.quantity.intent</small>
Patient: At least <u>around the block</u> (<i>M.quantity.distance</i>) to start.
Coach: <u>On a scale of 1 – 10 with 10 being very sure. How sure are you that you will accomplish your goal?</u>
<small>A.intent</small>
Patient: <u>5</u> (<i>A.score</i>)

Figure 4: Example showing usage of SMART goal annotation schema

annotations in dataset 1 minus the first two to three introduction messages about the study in each conversation as they were the same. Other than *follow up*, all other phases in GI stage rarely occur.

Focusing on transitions now, a priori, 121 different transitions are possible in a given week, as we have 10 unique stage-phase categories plus the beginning and end of the week (start, stop). However, only 39 unique transitions occur in our dataset,

given that a week always starts with the goal setting stage, which in turn starts with the goal identification phase. On further analysis, we found that only 13 of those 39 transitions have a probability above 0.3, as shown in Figure 3.

SMART Tag Annotation Schema: Similar to stage-phase annotations, 15 patient-coach conversations were used to design the SMART goal annotation schema. The schema is described in Table 4

Level	S	M	A	R
Message	0.967	0.965	0.907	0.694
Word	0.878	0.895	0.515	0.549

Table 5: Kappa on SMART goal annotation schema

Tag	Feature	Slot Value	Intent
Specificity	Activity	671	0
	Time	131	31
	Location	41	1
Measurability	Quantity	627	30
	Days	303	63
	Repetition	69	0
Attainability		70	261
Realism		N/A	70

Table 6: Counts for SMART tags in the dataset 1

along with examples. We didn’t annotate for Timeliness as a new goal was set every week, and hence by default, its value is one week. Each annotation can either be categorized as a slot value or an intention. A slot value is a word or group of words that capture a particular piece of information, for example, ‘walk’ is a slot value for *specific activity*; the intention is an utterance that tries to gain information about a slot. Each SMART annotation category can have other optional tags such as *previous* to annotate an attribute related to the previous week’s goal, *accomplished* or *remaining* to annotate the progress of the patient, *update* to add another slot value to an existing one, and *other* for anything which doesn’t belong to the previous or current week. Figure 4 shows the use of the SMART annotation schema.

Two annotators annotated four previously unseen patients’ data for SMART goal attributes. Results for IAA measured using kappa (κ) is shown in Table 5. We measured κ on two levels: message and word. At the message level, we consider an agreement if both the annotators labeled at least one word in the message with the given tag (not necessarily the same word). At the word-level, we consider it an agreement if both annotators labeled the same word with the given tag.

In total, 447 messages were annotated for IAA. There were 128 messages with Specificity (S) tag, 120 with Measurability (M) tag, 45 with Attainability (A) tag and 13 with Realism (R) tag. We achieved $\kappa \approx 0.9$ for {S, M} and $\kappa \approx 0.5$ for {A, R}. This is mostly because {S, M} tags have a higher number of occurrences in the data as compared to {A, R} which are hard to distinguish from each other and have very few occurrences. It should

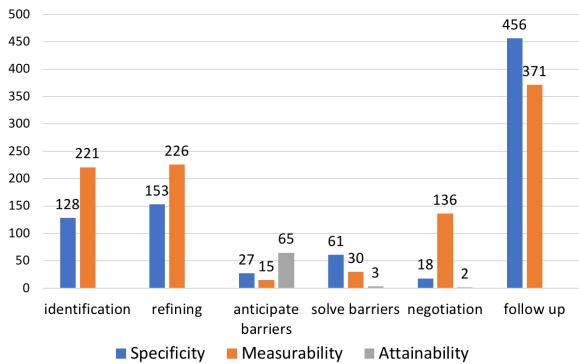


Figure 5: SMART tag counts per phase

also be noted that for {S, M} word-level annotation is more important whereas for {A, R} message level annotation makes more sense. Table 6 shows the counts for SMART categories in dataset 1. One can notice that the percentage of {R} is fairly small as compared to the {S, M, A} tags. It is not surprising as the coach only questions the realism of the goal if he thinks it is either too difficult/easy based on the patient’s past performances.

6 Development on Dataset 1

Dataset 1 has been our foundation to develop the computational models we are interested in, namely SMART tag and phase prediction. Before building these models, we wanted to see if SMART tags and phases exhibit any sort of relationship that can be leveraged as features. We plotted the number of SMART tags in each phase and obtained the graph shown in Figure 5. SMART tags are unevenly distributed across phases, with identification, refining and follow up containing the majority of SMART tags. Therefore, we experimented with SMART tags as a feature in the phase prediction model and vice versa, and found that SMART tags helped to predict phases better, than phases help predict SMART tags (Gupta et al., 2019).

We achieved an average (macro) F1 score of 0.80 on SMART tag prediction using Structured Perceptron with feature combination of the current and surrounding words, pre-trained Google word embeddings³, and SpaCy⁴ named entity recognizer output. Similarly, we achieved an average (macro) F1 score of 0.71 on phase prediction using Conditional Random Fields with feature combination of unigrams, distance of the message from the top in a given week, and human-annotated SMART tag

³<https://code.google.com/archive/p/word2vec/>

⁴<https://spacy.io/>

(1) Coach: Okay, so your goal this week is to reach 17,500 steps (<i>M_quantity_amount</i>) one day (<i>M_days_number</i>) this week (Monday through Sunday) (<i>M_days_name</i>), correct? (Human and Automated)
(2) Coach: Your goal is to reach 10,000 steps (<i>M_quantity_amount</i>) any day (<i>M_days_number</i>) this week by Friday (<i>M_days_name</i>)! You said that you give your confidence a 9 (<i>A_score</i>) on a 10 point scale. You can do this! (Human)
(2) Coach: Your goal is to reach 10,000 steps (<i>M_quantity_amount</i>) any day this week by Friday (<i>M_days_name</i>)! You said that you give your confidence a 9 (<i>A_score</i>) on a 10 point scale. You can do this! (Automated)

Figure 6: Automated annotation output (dataset 2).

counts. Importantly, an almost similar performance (F1 score = 0.69) was achieved using automatically predicted SMART tags.

Unfortunately, use of deep learning is not suitable due to our very small dataset; only 2853 messages in total. One can also notice rare occurrences of classes such as anticipate barriers, solve barriers, and negotiation in Figure 5.

7 Applications of Models Developed on Dataset 1

So far the models developed in Section 6 have been used for two applications: annotating dataset 2 for both SMART tags and phases, and goal extraction.

Goal extraction on dataset 1: Goal extraction can help health coaches to recall a goal discussed during the conversation and save their time. Since SMART tags helped predict phases better, we built a pipeline where SMART tags were predicted first, then they were used as one of the features in phase prediction. After SMART tag and phase prediction, we extracted the SMART tags as long as they were not from the follow-up phase to avoid extracting accomplished and remaining *measurable quantity*. 65% of the goals we extracted correctly identified at least 8 out of 10 SMART attributes of the gold standard goal. Detailed results for goal extraction and the two models are available in Gupta et al. (2020). We are currently evaluating our goal extraction model on dataset 2 with the help of health coaches and automatic evaluation.

Dataset 2 annotation: We used the same pipeline for annotating dataset 2, except we changed Google word embeddings to pre-trained ELMo word representations for SMART tag prediction (Peters et al., 2018). To measure performance, we manually annotated three randomly chosen pa-

tients’ data, one from each coach. We achieved an F1 score of 0.81 (macro) and 0.98 (weighted) on SMART tag annotations and 0.37 (macro) and 0.61 (weighted) on phase annotations. The results for SMART tag prediction on dataset 2 is equal to what we achieved on dataset 1. This means that SMART tag annotations are transferable even if the dialogues are between different coaches and patients. A sample output for SMART tags is shown in Figure 6. Our model correctly annotated (1), but missed *M_days_number* in (2). More specifically, for the three patients that we automatically annotated, only 113 words (2%) were incorrectly labelled or had a missing label; 390 words (6%) were correctly labelled with a SMART tag; and 5959 words (92%) were correctly labelled with ‘none’ tag.

Because performance on automatic phase annotation was not as high as we had hoped, we will adopt a semi-automatic approach, with a round of manual edits following automatic annotation of phases. We see semi-automatic annotation as crucial, especially given that state-of-the-art deep learning models require large labeled training data. Semi-automatic annotation can still save thousands of hours of manual labor.

8 Conclusions and Future Work

We envision a virtual assistant health coach that can help people to increase their physical activity by motivating them to set SMART goals. To this end, we collected a health coaching dialogue dataset and developed two annotation schemas, one that captures the slot-values of a SMART goal and the other that captures the higher-level conversation flow of the health coaching dialogues. We briefly discussed the models built using the two annotations and their application for automatic goal extraction. We also collected a second round of dataset and showed that it can be reliably annotated using the models built on the first dataset. Our immediate next steps are to perform extrinsic evaluation of the goal extraction pipeline with the help of our health coaches and integrate it into the Mytapp application used by the health coaches for round three of the data collection.

9 Acknowledgements

This work is supported by the National Science Foundation through awards IIS 1650900 and 1838770.

References

- Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A corpus for adding memory to goal-oriented dialogue systems. *arXiv preprint arXiv:1704.00057*.
- Timothy W Bickmore, Daniel Schulman, and Candace L Sidner. 2011. A reusable framework for health counseling dialogue systems based on a behavioral medicine ontology. *Journal of Biomedical Informatics*, 44(2):183–197.
- Thomas Bodenheimer, Connie Davis, and Halsted Holman. 2007. Helping patients adopt healthier behaviors. *Clinical Diabetes*, 25(2):66–70.
- Frank W Booth, Christian K Roberts, John P Thyfault, Gregory N Ruegsegger, and Ryan G Toedebusch. 2017. Role of inactivity in chronic diseases: evolutionary insight and pathophysiological mechanisms. *Physiological Reviews*, 97(4):1351–1402.
- Fiemke Both, Pim Cuijpers, Mark Hoogendoorn, Michel CA Klein, A Fred, J Filipe, and H Gamboa. 2010. Towards fully automated psychotherapy for adults: BAS-behavioral activation scheduling via web and mobile phone.
- Thamar JH Bovend'Eerdt, Rachel E Botell, and Derrick T Wade. 2009. Writing smart rehabilitation goals and achieving goal attainment scaling: a practical guide. *Clinical Rehabilitation*, 23(4):352–361.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadhan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.
- Ester Cerin, Evie Leslie, Takemi Sugiyama, and Neville Owen. 2010. Perceived barriers to leisure-time physical activity in adults: an ecological perspective. *Journal of Physical Activity and Health*, 7(4):451–459.
- Ananlada Chotimongkol and Alexander I Rudnicky. 2008. Acquiring domain-specific dialog information from task-oriented human-human interaction through an unsupervised learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 955–964. Association for Computational Linguistics.
- Clara K Chow, Julie Redfern, Graham S Hillis, Jay Thakkar, Karla Santo, Maree L Hackett, Stephen Jan, Nicholas Graves, Laura de Keizer, Tony Barry, et al. 2015. Effect of lifestyle-focused text messaging on risk factor modification in patients with coronary heart disease: a randomized clinical trial. *JAMA*, 314(12):1255–1263.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.
- Hamish Cunningham. 2002. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254.
- George T Doran. 1981. There's a SMART way to write management's goals and objectives. *Management Review*, 70(11):35–36.
- Nishitha Guntakandla and Rodney Nielsen. 2018. Annotating reflections for health behavior change therapy. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Itika Gupta, Barbara Di Eugenio, Brian Ziebart, Bing Liu, Ben Gerber, and Lisa Sharp. 2019. Modeling health coaching dialogues for behavioral goal extraction. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1188–1190. IEEE.
- Itika Gupta, Barbara Di Eugenio, Brian Ziebart, Bing Liu, Ben Gerber, and Lisa Sharp. 2020. Goal summarization for human-human health coaching dialogues. In *Florida Artificial Intelligence Research Society Conference*.
- Margaret Handley, Kate MacGregor, Dean Schillinger, Claire Sharifi, Sharon Wong, and Thomas Bodenheimer. 2006. Using action plans to help primary care patients adopt healthy behaviors: a descriptive study. *The Journal of the American Board of Family Medicine*, 19(3):224–231.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272.
- Spyros Kitsiou, Manu Thomas, G Elisabeta Marai, Nicos Maglaveras, George Kondos, Ross Arena, and Ben Gerber. 2017. Development of an innovative mhealth platform for remote physical activity monitoring and health coaching of cardiac rehabilitation patients. In *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 133–136. IEEE.
- Kirsi Kivelä, Satu Elo, Helvi Kyngäs, and Maria Kääriäinen. 2014. The effects of health coaching on adult patients with chronic diseases: a systematic review. *Patient Education and Counseling*, 97(2):147–157.
- Theresa B Moyers, Lauren N Rowell, Jennifer K Manuel, Denise Ernst, and Jon M Houck. 2016. The motivational interviewing treatment integrity code (miti 4): rationale, preliminary reliability and validity. *Journal of Substance Abuse Treatment*, 65:36–42.

Verónica Pérez-Rosas, Rada Mihalcea, Kenneth Resnicow, Satinder Singh, and Lawrence An. 2016. Building a motivational interviewing dataset. In *Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology*, pages 42–51.

Verónica Pérez-Rosas, Xuetong Sun, Christy Li, Yuchen Wang, Kenneth Resnicow, and Rada Mihalcea. 2018. Analyzing the quality of counseling conversations: the tell-tale signs of high-quality counseling. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Julia Poncela-Casasnovas, Bonnie Spring, Daniel McClary, Arlen C Moller, Rufaro Mukogo, Christine A Pellegrini, Michael J Coons, Miriam Davidson, Satyam Mukherjee, and Luis A Nunes Amaral. 2015. Social embeddedness in an online weight management programme is linked to greater weight loss. *Journal of The Royal Society Interface*, 12(104):20140686.

Melinda R Stolley, Lisa K Sharp, Giamila Fantuzzi, Claudia Arroyo, Patricia Sheean, Linda Schiffer, Richard Campbell, and Ben Gerber. 2015. Study design and protocol for moving forward: a weight loss intervention trial for african-american breast cancer survivors. *BMC Cancer*, 15(1):1018.

Alice Watson, Timothy Bickmore, Abby Cange, Ambar Kulshreshtha, and Joseph Kvedar. 2012. An internet-based virtual coach to promote physical activity adherence in overweight adults: randomized controlled trial. *Journal of Medical Internet Research*, 14(1).

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413.

Agent-Based Dynamic Collaboration Support in a Smart Office Space

Yansen Wang, R. Charles Murray, Haogang Bao, and Carolyn P. Rosé

Language Technologies Institute, Carnegie Mellon University

5000 Forbes Avenue, Pittsburgh PA, 15213

yansenwa, rcmurray, haogangb, cp3a@andrew.cmu.edu

Abstract

For the past 15 years, in computer-supported collaborative learning applications, conversational agents have been used to structure group interactions in online chat-based environments. A series of experimental studies has provided an empirical foundation for the design of chat-based conversational agents that significantly improve learning over no-support control conditions and static-support control conditions. In this demo, we expand upon this foundation, bringing conversational agents to structure group interaction into physical spaces, with the specific goal of facilitating collaboration and learning in workplace scenarios.

1 Introduction

AI-Enhanced human learning is a broad area of research with a history at least 50 years long (Alevin and Kay, 2016), with Carbonell’s SCHOLAR system being among the earliest systems (Carbonell, 1970). However, while great strides to introduce technologies to enhance both individual and collaborative learning have been made in relatively structured environments such as the lab and the classroom over the decades of research since that time, less progress has been made in more unstructured environments such as the workplace, where the stakes are far higher and social and political pressures play a more substantial role. This demo presents an apparatus for support of collaboration and learning in workplace scenarios using a Virtual Human facilitator interacting face-to-face through speech and gesture.

Large scale quantitative research, including experimental studies and carefully controlled quasi-experimental corpus studies, are the basis for learning generalizable principles (i.e., causal models) that underlie data-driven design of effective AI-enabled systems that support human learning. In recent decades, process data such as click logs,

discourse data, biometric sensors, and images are used to understand the process of human learning more deeply (Lang et al., 2017). Models trained over this process data are also used to enable real-time monitoring and support of learning processes even as groups learn through multi-party discussion (Adamson et al., 2014; Rosé and Ferschke, 2016). Thus, the ability to draw causal inferences to motivate effective interventions and the ability to trigger personalized, just-in-time support go hand-in-hand towards development of AI-enhanced learning experiences.

Much of the prior research on workplace learning is qualitative work, which focuses on deep understanding of individual contexts rather than producing generalizable principles through intervention studies. Thus, there is a dearth of empirical research that can rigorously motivate design of effective AI-enabled interventions to support workplace learning, and the data from such research is unable to support model-enabled real-time sensing technology that would facilitate just-in-time support for learning in the workplace. In response, we have constructed a “Smart Office Space” in which to run lab studies with simulated work conditions in order to discover causal mechanisms that can form the foundation for design.

2 Smart Office Space

2.1 Technical Description

As a resource for exploring how to introduce analytics and just-in-time support for collaborative learning during work, we have assembled a “Smart Office Space” which has been instrumented for behavioral sensing (See Figures 1 and 2). It is designed as a foundation for simulating workplace conditions for collaborative and individual desk work. Figure 1 displays the layout of the room while Figure 2 describes the architecture of the

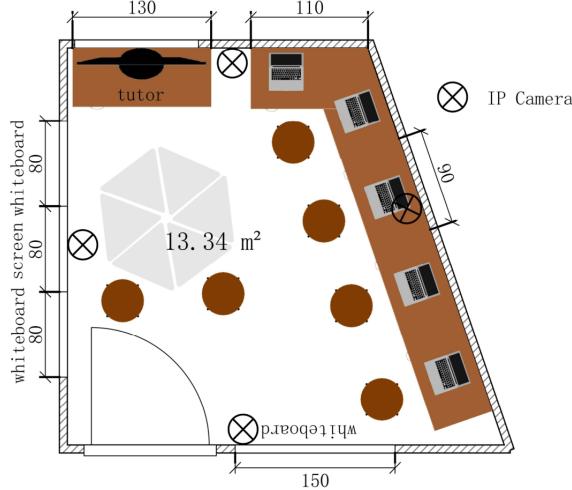


Figure 1: Room layout.

software infrastructure for monitoring and support.

The foundation for the dialogue-based support offered within the Smart Office Space is the Bazaar toolkit (Adamson et al., 2014), which has been used extensively as support for online collaborative learning groups. In this past work, Bazaar agents use what is referred to as an academically productive talk(APT)-based approach, which uses reasoning-focused prompts that encourage participants to articulate and elaborate their own lines of reasoning, and to challenge and extend the reasoning of their teammates in a group discussion. In order for students to learn and contribute to group discussions, it is important for students to articulate their reasoning and build on each other’s reasoning. This allows them to identify gaps in their knowledge and to observe how others think differently and might possess knowledge that they are missing. In this way, they have the opportunity to construct knowledge together as a group. The Bazaar toolkit has extensive authoring capabilities that enable a wide range of activities to be authored for virtually any topic area. Dozens of studies of group learning have been conducted with an online, text-based version of Bazaar. Here we place it within the Smart Office Space to communicate, not with text, but with speech and gesture within a physical space.

The room has been instrumented with a variety of sensors including four Lorex 4K cameras with microphones, a Kinect camera with a microphone array, an Intel RealSense depth-sensing camera, and an AWS DeepLens camera. Key software components include the Microsoft Platform for Situated Intelligence (PSI) (Bohus et al., 2017) for coordination across datastreams, CMU Sphinx (Lamere

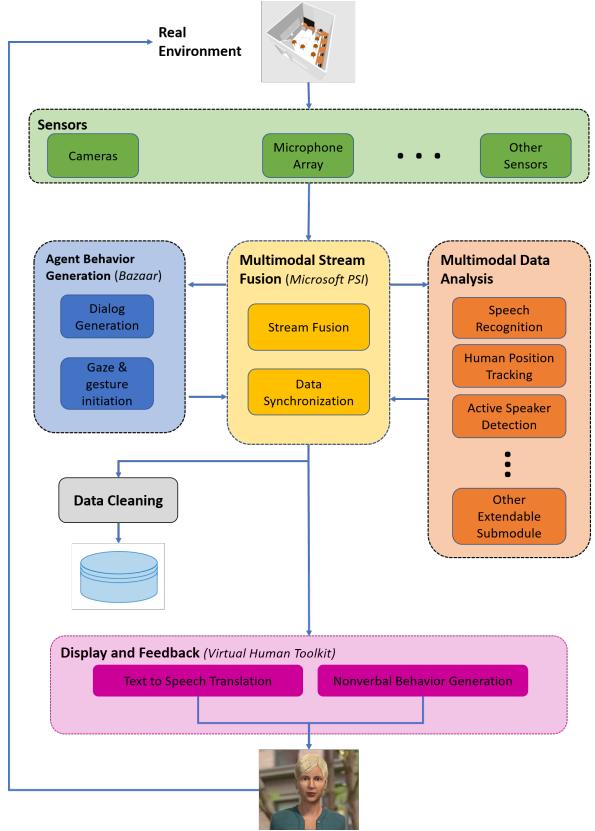


Figure 2: The software infrastructure.

et al., 2003) and the Azure Speech Recognizer for speech recognition, the USC Institute for Creative Technologies Virtual Human Toolkit (VHT) to present an embodied conversational agent (Hartholt et al., 2013), OpenFace for face recognition (Amos et al., 2016), OpenPose for sensing body movement and positioning (Cao et al., 2017), and Bazaar for sensing collaboration-relevant events (such as ideas that have not yet been elaborated or that no one has responded to or built on yet) and triggering support for collaboration in response (such as prompts that direct participants to consider and respond to the contribution of another participant) (Adamson et al., 2014).

2.1.1 Information Flow

Information flow for operating the Smart Office Space is displayed in Figure 2. As we develop the Smart Office Space, we are first focusing on using the audio and video data provided by the Lorex cameras and the Kinect microphone array to communicate with users via VHT. The data captured by the cameras and the microphone array are sent in separate streams to PSI. PSI passes the streams to audio and video recognizers. As the recognizers detect events, they pass event messages

back to PSI: Visual information is translated to semantic text describing body position and facial expressions; location information is translated to polar coordinates; and speech is translated to text. Recognition of the various audio and visual events may occur at different speeds, so PSI may receive event messages out of order. PSI therefore synchronizes event messages by their originating time, then passes on the translated events as appropriate: User location changes are passed directly to VHT to update agent gaze direction with low latency; speech translations along with user locations and visual events like a raised hand are passed to Bazaar; and some events are discarded, such as recognition that the agent itself is speaking. Messages between PSI and Bazaar use an internally developed multimodal message format that associates a user identifier with any combination of the following easily-expandable list of user attributes: location, speech text, body position, facial expression, and any detected emotion. Bazaar uses the information it receives from PSI to decide when and how to respond to events in the room, passing response messages through PSI, which coordinates verbal and nonverbal communication, to VHT for communication with users as a virtual agent.

2.1.2 Multimodal Stream Fusion

To handle multiple data streams, PSI ([Bohus et al., 2017](#)) provides a runtime environment for parallel, coordinated computation across data streams along with a set of tools for visualization, data processing and machine learning. We run PSI on Windows 10. PSI associates timestamps with the data it receives from video and audio streams and includes these timestamps as it passes the data on to the appropriate video and audio recognizers. When the recognizers detect events, they include the originating timestamps with the event messages that they return to PSI. PSI uses these timestamps to synchronize messages received on different streams, enabling it to identify both simultaneous events and the correct order of event sequences. PSI’s messages to Bazaar combine synchronous audio and video events. For instance, PSI might combine video recognition that a user is speaking, audio and visual recognition of the user’s location, and audio recognition of the user’s words in a single message to Bazaar. In addition, PSI logs all data that it receives for playback, analysis, and offline machine learning.

2.1.3 Multimodal Data Analysis

We incorporate multiple video and audio recognizers to process the video and audio streams received through PSI. Video recognizers run on a Linux GPU server for faster processing of neural network models. We use OpenFace ([Amos et al., 2016](#)) to find and recognize people facing any of the four cameras, including recognizing whether two face inputs are the same person. To detect body position and key body points, we use OpenPose([Cao et al., 2017](#)). OpenPose forwards its body points for the nose and neck to a location detector which maps these to lines in real space to triangulate users’ locations. For location verification, we are currently using a Kinect microphone array and we plan to try adding inputs from an Intel RealSense depth-sensing camera. For audio speech-to-text recognition, we are currently testing two packages integrated with PSI: CMU Sphinx ([Lamere et al., 2003](#)) and Microsoft Azure Speech to Text ¹.

2.1.4 Agent Behavior Generation

Bazaar receives event updates from PSI and uses this information to decide exactly how and when to respond to events. For instance, when a user enters the room, PSI sends Bazaar a message specifying a newly created internal user identifier along with the user’s location within the room, specified in terms of polar coordinates. Bazaar saves this information as the beginning of its user model. As additional information is acquired about the user – including spoken words as text, body position, facial expression, and apparent emotion – PSI sends event updates and Bazaar updates its user model accordingly. Bazaar’s responses can be tailored to the context. For example, if Bazaar wants to respond to an assertion by prompting the user to explain her reasoning, it can identify the user by associating the location of the speech source with the user’s saved location, call up the user’s name, and respond to the user by name while gazing in her direction.

2.1.5 Communication to Users

To communicate to users both verbally and non-verbally, Bazaar sends messages through PSI to VHT ([Hartholt et al., 2013](#)). VHT’s display to the user can be designed to represent a 3-dimensional setting with one or more actors that communicate to

¹<https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/>



Figure 3: Virtual tutor

users using speech, facial expressions, gaze directions, body position, and gestures. Speech is sent to VHT as text while non-verbal behavior is specified using the Behavior Markup Language (BML) realization library, “Smartbody”(Feng et al., 2012). At this stage, we communicate nonverbally using facial expressions, gaze direction, and simple arm and hand gestures. Facial expressions are specified in terms of lips, brows, and eyes, while gaze direction is realized through coordinated rotation of the shoulders, neck, head and eyes. We use these non-verbal cues to present some common non-verbal expressions – neutral, listening, confused, angry, happy, and amazed – and to gaze at individual users. For instance, if user Ron has offered an idea and Joan has not contributed to the ongoing group discussion in a while, the VHT may turn towards Joan and say, “Joan, can you build on the idea that Ron has offered?” Using the Smart Office Space, we are working towards collecting multiple datasets in collaboration with industry partners who help inform the characteristics of workplace scenarios for our studies including support for maintaining social distancing during intensive collaborative learning.

3 Demo Session

The video presentation of the demo for the online demo session will display scenarios in which groups of individuals work together on a task, with the VHT providing guidance for task structuring and collaborative work processes. What makes the demo unique among other applications of in person multi-party dialogue is the use of the virtual human as a group learning facilitator, enabled through the Bazaar architecture.

Acknowledgments

This work was funded in part by NSF grants IIS 1822831 and 1917955 and funding from Microsoft.

References

- David Adamson, Gregory Dyke, Hyeju Jang, and Carolyn Penstein Rosé. 2014. Towards an agile approach to adapting dynamic collaboration support to student needs. *International Journal of Artificial Intelligence in Education*, 24(1):92–124.
- Vincent Aleven and Judy Kay. 2016. *International Journal of AI in Education, 25th Anniversary Edition, volume 26(1-2)*. Springer.
- Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. 2016. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science.
- Dan Bohus, Sean Andrist, and Mihai Jalobeanu. 2017. Rapid development of multimodal interactive systems: A demonstration of platform for situated intelligence. In *ICMI 2017 Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 493–494.
- Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299.
- Jaime R Carbonell. 1970. Ai in cai: An artificial-intelligence approach to computer-assisted instruction. *IEEE transactions on man-machine systems*, 11(4):190–202.
- Andrew Feng, Yazhou Huang, Yuyu Xu, and Ari Shapiro. 2012. Automating the transfer of a generic set of behaviors onto a virtual character. In *International Conference on Motion in Games*, pages 134–145. Springer.
- Arno Hartholt, David Traum, Stacy C Marsella, Ari Shapiro, Giota Stratou, Anton Leuski, Louis-Philippe Morency, and Jonathan Gratch. 2013. All together now. In *International Workshop on Intelligent Virtual Agents*, pages 368–381. Springer.
- Paul Lamere, Philip Kwok, Evandro Gouveia, Bhiksha Raj, Rita Singh, William Walker, Manfred Warmuth, and Peter Wolf. 2003. The cmu sphinx-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, volume 1, pages 2–5.
- Charles Lang, George Siemens, Alyssa Wise, and Dragana Gasevic. 2017. *Handbook of learning analytics*. SOLAR, Society for Learning Analytics and Research.
- Carolyn Penstein Rosé and Oliver Ferschke. 2016. Technology support for discussion based learning: From computer supported collaborative learning to the future of massive open online courses. *International Journal of Artificial Intelligence in Education*, 26(2):660–678.

Emora STDM: A Versatile Framework for Innovative Dialogue System Development

James D. Finch

Department of Computer Science
Emory University
Atlanta, GA, USA
jdfinch@emory.edu

Jinho D. Choi

Department of Computer Science
Emory University
Atlanta, GA, USA
jinho.choi@emory.edu

Abstract

This demo paper presents Emora STDM (State Transition Dialogue Manager), a dialogue system development framework that provides novel workflows for rapid prototyping of chat-based dialogue managers as well as collaborative development of complex interactions. Our framework caters to a wide range of expertise levels by supporting interoperability between two popular approaches, state machine and information state, to dialogue management. Our Natural Language Expression package allows seamless integration of pattern matching, custom NLP modules, and database querying, that makes the workflows much more efficient. As a user study, we adopt this framework to an interdisciplinary undergraduate course where students with both technical and non-technical backgrounds are able to develop creative dialogue managers in a short period of time.

1 Introduction

Constructing a functional end-to-end dialogue system is typically an extensive development process. Depending on the goals, such development often involves defining models for natural language understanding and generation (Section 3), and also creating dialogue management logic to control conversation flow. Training a deep learning-based end-to-end model is a cost-effective way to develop a dialogue agent when the goal is a system conforming to behaviors present in training data; however, substantial development effort must be spent as the developer demands broaden to incorporate features that are not well-represented in available data.

We present Emora STDM (State Transition Dialogue Manager), henceforth E-STDM, a dialogue system development framework that offers a high degree of customizability to experts while preserving a workflow intuitive to non-experts. E-STDM caters to a wide range of technical backgrounds by

supporting the interoperability between two popular dialogue management approaches, state machine and information state (Section 4). Our framework makes it easy for not only rapid prototyping of open-domain and task-oriented dialogue systems, but also efficient development of complex dialogue managers that tightly integrate pattern matching, NLP models, and custom logic such as database queries. (Section 5).

2 Related Work

A variety of dialogue development frameworks have emerged to expedite the process of dialogue system creation. These frameworks cater to various use cases and levels of developer expertise. Popular commercial-oriented frameworks are primarily intended for non-experts and have workflows supporting rapid prototyping (Bocklisch et al., 2017). They often allow developers to customize natural language understanding (NLU) modules and perform dialogue management using state machines.

Some frameworks require more expertise, but offer better developer control, by following the information state formulation of dialogue management (Ultes et al., 2017; Jang et al., 2019; Kiefer et al., 2019). According to this formulation, dialogues are driven by iterative application of logical implication rules (Larsson and Traum, 2000). This design provides support for complex interactions, but sacrifices the intuitiveness and development speed of dialogue management based on state machines.

Other frameworks (e.g., ChatScript, botml) rely on custom programming languages to design conversation flow. The custom syntax they specify is based on pattern matching. Although requiring expertise, rapid prototyping in these frameworks is possible with a high degree of developer’s control. However, dialogue management focuses primarily on shallow pattern-response pairs, making complex interactions more difficult to model.

ID	Framework	Type	License	SM	IS	PM	IC	EF	ON	ET	CM
1	Emora STDM	Library	Apache 2.0	✓	✓	✓	✓	✓	✓	✓	✓
2	AIML	Language	GNU 3.0			✓					
3	RiveScript	Language	MIT			✓		✓			✓
4	ChatScript	Language	MIT	✓		✓		✓	✓		
5	botml	Language	MIT	✓		✓		✓			
6	OpenDial	Tool	MIT		✓	✓		✓			
7	PyDial	Tool	Apache 2.0		✓			✓	✓		✓
8	VOnDA	Tool	CC BY-NC 4.0		✓	✓		✓	✓		
9	Botpress	Tool	Commercial	✓			✓	✓		✓	
10	RASA	Tool	Commercial	✓			✓	✓		✓	
11	DialogFlow	API	Commercial	✓			✓			✓	

Table 1: Comparison of features supported by various dialogue system development frameworks. SM: state machine, IS: information state, PM: pattern matching for natural language, IC: developer-trained intent classification, EF: external function calls, ON: ontology, ET: error tracking, CM: combine independent dialogue systems.

Table 1 shows a comparison of E-STDM to existing frameworks. E-STDM is most similar to PyOpen-Dial and botml, which support pattern matching for NLU and tight integration of external function calls. Unlike any existing framework, however, E-STDM explicitly supports both state machine and information state paradigms for dialogue management, and also provides NLU that seamlessly integrates pattern matching and custom modules.¹

3 NATEX: Natural Language Expression

To address the challenge of understanding user input in natural language, we introduce the NATural language EXpression, NATEX, that defines a comprehensive grammar to match patterns in user input by dynamically compiling to regular expressions. This dynamic compilation enables abstracting away unnecessary verbosity of regular expression syntax, and provides a mechanism to embed function calls to arbitrary Python code. We highlight the following key features of NATEX.

String Matching It offers an elegant syntax for string matching. The following NATEX matches user input such as ‘I watched avengers’ or ‘I saw Star Wars’ and returns the variable \$MOVIE with the values ‘Avengers’ and ‘Star wars’, respectively:

```
[I {watched, saw}
$MOVIE={Avengers, Star Wars}]
```

The direct translation of this NATEX to a regular expression would be as follows:

```
. *? \b I \b
. *? (?: \b(?:watched) \b| \b(?:saw) \b)
. *? (?: P<MOVIE> (?: \b(?:avengers) \b|
\ b(?:star wars) \b)) . *?
```

As shown, NATEX is much more succinct and interpretable than its counterpart regular expression.

¹Emora STDM is available as an open source project at github.com/emora-chat/emora_stdm.

Function Call It supports external function calls. The following NATEX makes a call to the function #MDB in Python that returns a set of movie titles:

```
[I {watched, saw} $MOVIE=#MDB()]
```

This function can be implemented in various ways (e.g., database querying, named entity recognition), and its NATEX call matches substrings in the user input to any element of the returned set. Note that not all elements are compiled into the resulting regular expression; only ones that are matched to the user input are compiled so the regular expression can be processed as efficiently as possible.

Ontology It supports ontology editing and querying as the built-in NATEX function called #ONT. An ontology can be easily built and loaded in JSON. #ONT(movie) in the example below searches for the movie node in a customizable ontology represented by a directed acyclic graph and returns a set of movie titles from the subgraph of movie:

```
[I {watched, saw} $MOVIE=#ONT(movie)]
```

Response Generation It can be used to generate system responses by randomly selecting one production of each disjunction (surrounded by {}) in a top-down fashion. The following NATEX can generate “I watched lots of action movies lately” or “I watched lots of drama movies recently”, and assign the values of ‘action’ and ‘drama’ to the variable \$GENRE respectively:

```
I watched lots of $GENRE={action,
horror, drama} movies {recently, lately}
```

Error Checking Our NATEX compiler uses the Lark parser to automatically detect syntax errors.² Additionally, several types of error checking are performed before runtime such as:

²<https://github.com/lark-parser/lark>

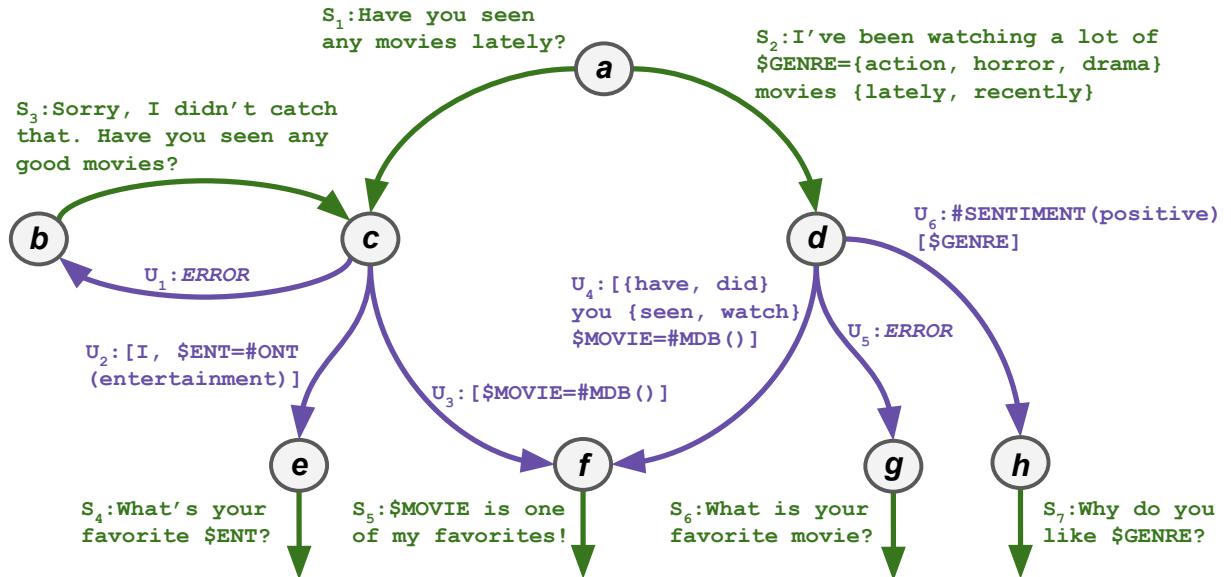


Figure 1: A dialogue graph using a state machine approach with NATEX to dialogue management.

- Call to a non-existing function.
- Exceptions raised by any function.
- Function returning mismatched type.
- Reference to a non-existing variable.

4 Dialogue Management

4.1 Dialogue State Machine

The primary component responsible for dialogue management within E-STDM is a state machine. In our framework, state transitions alternate between the user and the system to track turn taking, and are defined by NATEX (Figure 1). Any transition performed with a variable-capturing NATEX will store a variable-value pair in a dedicated table in memory, which persists globally for future reference.

User turns are modeled by transitions according to which NATEX matches the user input. To resolve cases where multiple NATEX yield matches, transitions can be defined with priority values. Similarly, developers can specify a catch-all “error transition” (ERROR in Figure 1) to handle cases where no transition’s NATEX returns a match. The user input resulting in an error transition is automatically logged to improve the ultimate design of the state machine.

System turns are modeled by randomly selecting an outgoing system transition. Random selection promotes uniqueness among dialogue pathways, but can be restricted by specifying explicit priority values. To avoid redundancy when returning to a previously visited state, E-STDM prefers system transitions that have not been taken recently.

The simplicity of this dialogue management formulation allows for rapid development of contextually aware interactions. The following demonstrates the streamlined JSON syntax for specifying transitions S_1, S_3, U_1, U_2 , and U_3 in Figure 1.

```
{
  "Have you seen any movies lately?": {
    "state": "c",
    "[I, \$ENT=\#\text{ONT}(\text{entertainment})]": {
      "What's your favorite \$ENT?": {...}
    },
    "[\$MOVIE=\#\text{MDB}())": {
      "\$MOVIE is one of my ...": {...}
    }
  },
  "error": {
    "Sorry, I didn't catch ...": "c"
  }
}
```

4.2 Information State Update Rules

Despite its simplicity, state machine-based dialogue management often produces sparsely connected state graphs that are overly rigid for complex interactions (Larsson and Traum, 2000). E-STDM thus allows developers to specify information state update rules to take advantage of the power of information state-based dialogue management.

Information state update rules contain two parts, a precondition and a postcondition. Each user turn before E-STDM’s state machine takes a transition, the entire set of update rules is iteratively evaluated with the user input until either a candidate system response is generated or no rule’s precondition is satisfied. In the following example, satisfying precondition $[\text{I have } \$\text{USER_PET}=\#\text{PET}()]$ triggers postcondition $\#\text{ASSIGN}(\$\text{USER_LIKE}=\$USER_PET)$

```

to assign $USER_PET to $USER_LIKE, allowing rule
#IF(..) I like $USER_LIKE .. to trigger in turn:

{
    "[I have $USER_PET=#PET() ]"
    : "#ASSIGN($USER_LIKE=$USER_PET)",
    "[${USER_FAVOR=#PET() is my favorite}"
    : "#ASSIGN($USER_LIKE=$USER_FAVOR)",
    "#IF($USER_LIKE != None)"
    : "I like $USER_LIKE too! (0.5)"
}

```

When a precondition is satisfied, the postcondition is applied through the language generation (Sec. 3). If a real-number priority is provided in parentheses at the end of any NATEX, the generated string becomes a candidate system response. A priority value higher than any outgoing system transition in the dialogue state machine results in the candidate becoming the chosen one; thus, no dialogue state machine transition is taken. Often however, a developer can choose to omit the priority value to use NATEX purely as a state updating mechanism.

This formalism allows flexible interoperability between state machine-based and information state-based dialogue management. Given E-STDM, developers have the latitude to develop a system entirely within one of the two approaches, although we believe a mixed approach lends the best balance of development speed and dialogue sophistication.

4.3 Combining Dialogue Modules

E-STDM has explicit support for a team-oriented workflow, where independent dialogue modules can be easily combined into one composite system. Combining multiple modules requires specification of a unique namespace per module to enforce encapsulation of both errors and identifiers. The following is an example Python script combining dialogue systems df1 and df2 under namespaces DF1 and DF2, respectively:

```

df1 = DialogueFlow('start_1')
df1.add_transitions('df1.json')
df2 = DialogueFlow('start_2')
df2.add_transitions('df2.json')

cdf = CompositeDialogueFlow('start')
cdf.add_module(df1, 'DF1')
cdf.add_module(df2, 'DF2')
cdf.add_user_transition(
    'DF1.stateX', 'DF2.stateY',
    "[{film, movie}]")

```

Moreover, inter-component transitions can be made between any two dialogue states to seamlessly combine modules together and allow smooth topic transitions for better user experience.

5 Educational Use of Emora STDM

As an application case study, we present the use of E-STDM in an educational setting. E-STDM is deployed in an interdisciplinary undergraduate course called *Computational Linguistics*,³ where dialogue system development within E-STDM is a part of the requirements. Students in this course come with varying levels of programming ability; many with little to no imperative programming experience.

Students are tasked with the development of chat-based dialogue systems that can engage a user in 10+ turn conversations. At the time of writing, students have completed two assignments involving dialogue system creation. Students are grouped in teams, with at least one student with prior coding experience per team. Teams are free to select a domain, such as video games, sports, or technology, and are given two weeks for development.

We make the unmodified version of dialogue systems from these students publicly available.⁴ The successful use of E-STDM by novice programmers demonstrates the utility of this framework, in terms of its usability and potential as an educational tool.

Acknowledgments

We gratefully acknowledge Sarah E. Finch for her support in developing E-STDM as well as assessing the course assignments (Section 5).

References

- T. Bocklisch, J. Faulkner, N. Pawłowski, and A. Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv:1712.05181*.
- Y. Jang, J. Lee, J. Park, K. Lee, P. Lison, and K. Kim. 2019. PyOpenDial: A Python-based Domain-Independent Toolkit for Developing Spoken Dialogue Systems with Probabilistic Rules. In *Proceedings of EMNLP System Demonstrations*.
- B. Kiefer, A. Welker, and C. Biwer. 2019. Vonda: A framework for ontology-based dialogue management. *arXiv:1910.00340*.
- S. Larsson and D. R. Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *NLE*, 6(3 & 4):323–340.
- S. Ultes, Rojas B., Lina M., P. Su, D. Vandyke, D. Kim, I. Casanueva, P. Budzianowski, N. Mrkšić, T. Wen, M. Gašić, and S. Young. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL System Demonstrations*.

³github.com/emory-courses/cs329

⁴github.com/emora-chat/emora_stdm_zoo

Boosting Naturalness of Language in Task-oriented Dialogues via Adversarial Training

Chenguang Zhu

Microsoft Speech and Dialogue Research Group, Redmond, WA, USA

chezhu@microsoft.com

Abstract

The natural language generation (NLG) module in a task-oriented dialogue system produces user-facing utterances conveying required information. Thus, it is critical for the generated response to be natural and fluent. We propose to integrate adversarial training to produce more human-like responses. The model uses Straight-Through Gumbel-Softmax estimator for gradient computation. We also propose a two-stage training scheme to boost performance. Empirical results show that the adversarial training can effectively improve the quality of language generation in both automatic and human evaluations. For example, in the RNN-LG Restaurant dataset, our model AdvNLG outperforms the previous state-of-the-art result by 3.6% in BLEU.

1 Introduction

In task-oriented dialogues, the computer system communicates with the user in the form of a conversation and accomplishes various tasks such as hotel booking, flight reservation and retailing. In this process, the system needs to accurately convert the desired information, a.k.a. *meaning representation*, to a natural utterance and convey it to the users (Table 1). The quality of response directly impacts the user’s impression of the system. Thus, there are numerous previous studies in the area of natural language generation (NLG) for task-oriented dialogues, ranging from template-based models (Cheyer and Guzzoni, 2014; Langkilde and Knight, 1998) to corpus-based methods (Dušek and Jurčíček, 2016; Tran and Nguyen, 2017; Wen et al., 2015; Zhu et al., 2019).

However, one issue yet to be solved is that the system responses often lack the fluency and naturalness of human dialogs. In many cases, the system responses are not natural, violating inherent human language usage patterns. For instance,

Input	name[Wildwood], eatType[restaurant], food[Indian], area[riverside], familyFriendly[no], near[Raja Indian Cuisine]
with adv.	Wildwood is an Indian restaurant in the riverside area near Raja Indian Cuisine . It is not family friendly.
w/o adv.	Wildwood is a restaurant providing Indian food. It is located in the riverside . It is near Raja Indian Cuisine .

Table 1: Example of generated utterances from meaning representation input. Our model learns to put two pieces of location information in one sentence via adversarial training.

in the last row of Table 1, two pieces of location information for the same entity *restaurant* should not be stated in two separate sentences. In another example in Table 4, the positive review *child friendly* and the negative review *low rating* should not appear in the same sentence connected by the conjunction *and*. These nuances in language usage do impact user’s impression of the dialogue system, making the system response rigid and less natural.

To solve this problem, several methods use reinforcement learning (RL) to boost the naturalness of generated responses (Ranzato et al., 2015; Li et al., 2016). However, the Monte-Carlo sampling process in RL is known to have high variance which can make the training process unstable. Li et al. (2015) proposes to use maximum mutual information (MMI) to boost the diversity of language, but this criterion makes exact decoding intractable.

On the other hand, the adversarial training for natural language generation has shown to be promising as the system needs to produce responses indiscernible from human utterances (Rajeswar et al., 2017; Wu et al., 2017; Nie et al., 2018). Apart from the generator, there is a dis-

criminator network which aims to classify system responses from human results. The generator is trained to fool the discriminator, resulting in a min-max game between the two components which boosts the quality of generated utterances (Goodfellow et al., 2014). Due to the discreteness of language, most previous work on adversarial training in NLG apply reinforcement learning, suffering from high-variance problem (Yu et al., 2017; Li et al., 2017; Ke et al., 2019).

In this work, we apply adversarial training to utterance generation in task-oriented dialogues and propose the model AdvNLG. Instead of using RL, we follow Yang et al. (2018) to leverage the Straight-Through Gumbel-Softmax estimator (Jang et al., 2016) for gradient computation. In the forward pass, the generator uses the argmax operation on vocabulary distribution to select an utterance and sends it to the discriminator. But during backpropagation, the Gumbel-Softmax distribution is used to let gradients flow back to the generator. We also find that pretraining the generator for a warm start is very helpful for improving the performance.

To evaluate our model, we conduct experiments on public datasets E2ENLG (Novikova et al., 2017) and RNN-LG (Wen et al., 2016). Our model achieves strong performance and obtains new state-of-the-art results on four datasets. For example, in Restaurant dataset, it improves the best result by 3.6% in BLEU. Human evaluation corroborates the effectiveness of our model, showing that the adversarial training against human responses can make the generated language more accurate and natural.

2 Problem Formulation

The goal of natural language generation module in task-oriented dialogues is to produce system utterances directly issued to the end users (Young, 2000). The generated utterances need to carry necessary information determined by upstream dialogue modules, including the *dialogue act* (DA) and *meaning representation* (MR).

The dialogue act specifies the type of system response (e.g. *inform*, *request* and *confirm*), while the meaning representation contains rich information that the system needs to convey to or request from the user in the form of slot-value pairs. Each slot indicates the information category and each value represents the information content.

Therefore, the training data for the supervised NLG task is $\{x_i = (d_i, r_i), y_i\}_{i=1}^n$, where d_i is the dialogue act, $r_i = \{(s_1, v_1), \dots, (s_t, v_t)\}$ is the set of MR slot-value pairs, and y_i is the human-labeled response.

NLG models typically use *delexicalization* during training and inference, replacing slots and values in the utterance with a special token $\langle SLOT\ NAME \rangle$. In this way, the system does not need to generate the proper nouns. Finally, the model substitutes these special tokens with corresponding values when delivering to users.

3 Model

3.1 Generator Model

We use the sequence-to-sequence encoder-decoder architecture (Sutskever et al., 2014) for the response generator G . The input to the encoder is a single sequence x of length m via concatenating dialogue act d and slots and values in the meaning representation r . The target utterance y has n tokens, y_1, \dots, y_n . Following Zhu et al. (2019), we delexicalize both sequences and surround each sequence with $\langle BOS \rangle$ and $\langle EOS \rangle$ tokens.

Both the encoder and decoder use GRU (Cho et al., 2014) for contextual embedder, and they share the embedding matrix E to map each token to a fixed-length vector. The final hidden state of the encoder RNN is used as the initial state of the decoder RNN. Moreover, the decoder employs a dot-product attention mechanism (Bahdanau et al., 2014) over the encoder states to get a context vector c at each decoding step.

This context vector c is concatenated with the embedding of the current token and fed into the GRU to predict the next token. The result $p_t = p(y_t|y_1, \dots, y_{t-1}; x)$ is the probability distribution of the next token over all tokens in dictionary V .

We use cross entropy as the generator’s loss function. Suppose the one-hot ground-truth token vector at the t -th step is y_t , then the loss is:

$$\mathcal{L}_{Gen}(\theta) = - \sum_{t=1}^n y_t^T \log(p_t) \quad (1)$$

3.2 Adversarial Training

The goal of the adversarial training is to use a discriminator to differentiate between the utterance y' from generator and the ground-truth utterance y .

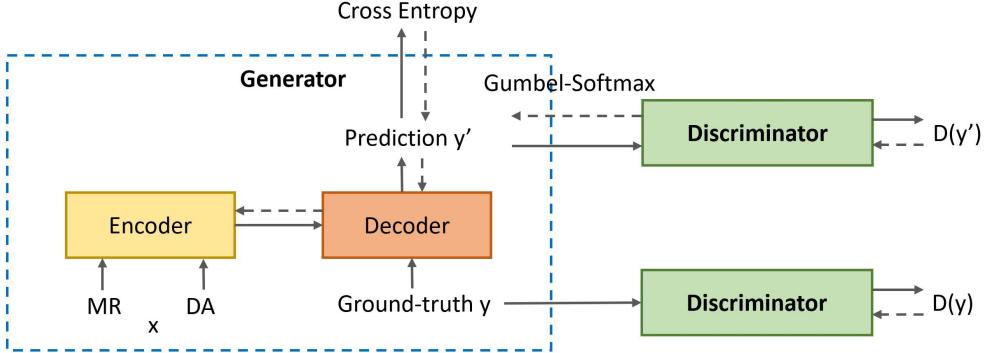


Figure 1: AdvNLG model with generator and discriminator. The dashed arrow is the direction of gradient flow.

We leverage the improved version of generative adversarial network (GAN), Wasserstein-GAN (WGAN) (Arjovsky et al., 2017), in our framework. WGAN designs a min-max game between the generator G and the discriminator D :

$$\min_G \max_D E_{y \sim P_{data}(y)}[D(y)] - E_{y' \sim G(x)}[D(y')] \quad (2)$$

where $G(x)$ denotes the probability distribution computed by the generator G given input x . The discriminator function D is a scoring function on utterances.

The goal of the generator is to obtain y' as similar as possible to y to fool the discriminator D (the outer-loop min), while D learns to successfully classify generated output y' from the ground-truth y (the inner-loop max), via the scoring function D .

3.2.1 Discriminator Model

For the discriminator, we reuse the embedding matrix \mathbf{E} as the embedder, followed by a bidirectional GRU layer. The last GRU hidden state \mathbf{h} is passed through a batch normalization layer and a linear layer to get the final score $D(y)$:

$$\mathbf{r} = \text{BatchNorm}(\mathbf{h}) \quad (3)$$

$$D(y) = \mathbf{W}_3 \mathbf{r} + \mathbf{b}_3, \quad (4)$$

where \mathbf{W}_3 and \mathbf{b}_3 are trainable parameters.

3.2.2 Training

Gradient computation. One problem with adversarial training in language generation is that the token sequence y' sampled from G is discrete, making it impossible to back-propagate gradients from the min-max objective to the generator.

Several previous methods leverage reinforcement learning for gradient computation (Yu et al., 2017; Li et al., 2017). However, the related sampling process can introduce high variance dur-

ing training. Therefore, we employ the Straight-Through Gumbel-Softmax estimator (Jang et al., 2016; Baziotis et al., 2019). In detail, during the forward pass, at the t -th step, the argmax of the generated word distribution \mathbf{p}_t is taken, i.e. greedy sampling. But for gradient computation, the Gumbel-Softmax distribution is used as a differentiable alternative to the argmax operation:

$$\mathbf{p}'_{t,i} = \frac{\exp(\log(\mathbf{p}_{t,i}) + g_i)/\tau}{\sum_{j=1}^{|V|} \exp(\log(\mathbf{p}_{t,j}) + g_j)/\tau}, \quad (5)$$

where $g_1, \dots, g_{|V|}$ are i.i.d samples drawn from the Gumbel distribution $G(0, 1)$ and τ represents the softmax temperature. Jang et al. (2016) shows that the Gumbel-Softmax distribution converges to the one-hot distribution as $\tau \rightarrow 0$ and to the uniform distribution as $\tau \rightarrow \infty$. We set $\tau = 0.1$ in all the experiments.

Two-stage Training. We find that the adversarial training does not work well if we optimize both the cross entropy (Eq. 1) and the min-max objective (Eq. 2) from the beginning. However, after we warm up the generator model with only cross entropy loss for several epochs, and then train with the discriminator under both the cross entropy and adversarial objective, the performance is consistently boosted. We argue that during early stages, the generator cannot produce meaningful output, making the discriminator easy to overfit. It's then hard for generator to learn to fool the adversary.

We summarize our model AdvNLG and gradient computation process in Fig. 1.

4 Experiments

We conduct empirical tests on a number of benchmarks for task-oriented dialogues over a variety of domains such as restaurant booking, hotel booking and retail. The datasets include the E2E-NLG

task (Novikova et al., 2017) with 51.4K samples, and the TV, Laptop, Hotel and Restaurant datasets from RNN-LG (Wen et al., 2016), with 14.1K, 26.5K, 8.7K and 8.5K samples respectively. We use BLEU-4 (Papineni et al., 2002) for the automatic metric, computed by the official evaluation scripts from E2E-NLG and RNN-LG.

4.1 Baselines

The baseline systems include TGen (Dušek and Jurčíček, 2016), SC-LSTM (Wen et al., 2015), RALSTM (Tran and Nguyen, 2017), Slug (Juraska et al., 2018), S2S+aug (Nie et al., 2019) and NLG-LM (Zhu et al., 2019). We also implement adversarial training using reinforcement learning in the same way as Li et al. (2017), denoted by RL. The generator in RL is warmed up in the same way as AdvNLG.

4.2 Training Details

In all experiments, the learning rate is 1e-3, the batch size is 20 and the beam width in inference is 10. According to WGAN, the discriminator’s parameters are clipped at 0.1. We use RMSprop (Ruder, 2016) as the optimizer. Teacher forcing is used for training the generator, which means that the decoder is exposed to the previous ground-truth token. In warm-up phase, we train the generator for 2 epochs. In E2E-NLG dataset, the generator is updated 5 times before the discriminator is updated once, which is typical in GAN training (Wu et al., 2017). The hyper-parameters above are chosen based on performance on the dev set. Other hyper-parameters like dropout rate, dictionary dimension and RNN hidden size are the same with Table 3 in Zhu et al. (2019).

For baseline models, we implemented NLG-LM (Zhu et al., 2019) and reproduced its results. We obtain the prediction results of Slug (Juraska et al., 2018) from its open-source website.

4.3 Results

As shown in Table 2, our model AdvNLG achieves new state-of-the-art results on TV, Laptop, Hotel and Restaurant datasets, improving previous best results by 0.8%, 3.8%, 0.6% and 3.6%. Statistical tests show that this advantage is statistically significant with p-values smaller than 0.05. Our model also obtains results on par with NLG-LM on E2ENLG. We show some prediction examples in Table 4. Generally, with adversarial training, the generated output can group information from

Model	E	TV	L	H	R
TGen	0.659	/	/	/	/
Slug	0.662	0.529	0.524	/	/
SCLSTM	/	0.527	0.512	0.848	0.752
RALSTM	/	0.541	0.525	0.898	0.779
S2S+aug	0.665	/	/	/	/
NLG-LM	0.684	0.617	0.586	0.939	0.795
AdvNLG	0.683	0.625*	0.624*	0.945*	0.831*
RL	0.674	0.605	0.606	0.932	0.796
-Adv.	0.671	0.618	0.564	0.931	0.753
-2 stages	0.662	0.621	0.557	0.932	0.782

Table 2: BLEU scores on E2ENLG (E), TV, Laptop (L), Hotel (H) and Restaurant (R) testset. *: means the result is statistically significant with p-value<0.05. -Adv. means we only train the generator, with cross entropy loss. -2 stages means that both the generator and discriminator are trained together from scratch.

the same category together, while placing positive and negative aspects (e.g. family-friendly and expensive) in different sentences.

Ablation Study. The bottom section of Table 2 shows that adversarial training can boost performance by 0.7% to 7.8%. Our proposed two-stage training is also very beneficial. If both generator and discriminator are trained from scratch, the result drops significantly. RL-based adversarial training achieves mixed results. On TV dataset, it even hurts the performance. We attribute this to the high variance and instability in training.

Model	Naturalness	Accuracy
Slug	2.51 (0.48)	2.89 (0.36)
NLG-LM	2.52 (0.46)	2.84 (0.41)
AdvNLG	2.84* (0.27)	2.97* (0.17)
-Adv.	2.45 (0.53)	2.63 (0.58)

Table 3: Average human evaluation ratings (1-3, 3 is best) for naturalness and accuracy of output generated by different models. Standard deviation is shown in parenthesis. *: the p-value is smaller than 0.01.

4.4 Human Evaluation

We randomly sample 100 data-text pairs from the test set of E2ENLG. We then ask 3 labelers to judge the accuracy and naturalness of the utterances generated by Slug, NLG-LM, AdvNLG with and without adversarial training. The accuracy measures how precise the utterance expresses

the dialogue act and meaning representation. The naturalness is measured by how likely the labeller thinks the utterance is spoken by a real human. In addition to the model output, each labeler is also given the meaning representation and the ground truth. The labelers need to give an integer rating from 1 to 3 (3 being the best) for each criterion.

Table 3 shows that our AdvNLG model has an apparent lead in both naturalness and accuracy, and the paired t-test shows that the result is statistically significant with p-value smaller than 0.01. And our ablation model -Adv. achieves the lowest score, proving that adversarial training can boost both naturalness and accuracy.

5 Conclusion

In this paper, we propose adversarial training using the Straight-Through Gumbel-Softmax estimator in NLG for task-oriented dialogues. We also propose a two-stage training scheme to further boost the gain in performance. Experimental results show that our model, AdvNLG, consistently outperforms state-of-the-art models in both automatic and human evaluations.

In the future, we plan to apply this method to other conditional generation tasks, e.g. produce a natural utterance containing a given list of keywords.

Acknowledgement

We thank the anonymous reviewers for their valuable comments. We thank William Hinthon for proof-reading the paper.

References

- Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. Seq³: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. *arXiv preprint arXiv:1904.03651*.
- Adam Cheyer and Didier Guzzoni. 2014. Method and apparatus for building an intelligent automated assistant. US Patent 8,677,377.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *arXiv preprint arXiv:1606.05491*.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. pages 2672–2680.
- Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.
- Juraj Juraska, Panagiotis Karagiannis, Kevin K Bowden, and Marilyn A Walker. 2018. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. *arXiv preprint arXiv:1805.06553*.
- Pei Ke, Fei Huang, Minlie Huang, and Xiaoyan Zhu. 2019. Araml: A stable adversarial training framework for text generation. *arXiv preprint arXiv:1908.07195*.
- Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 704–710. Association for Computational Linguistics.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.
- Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.
- Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.
- Feng Nie, Jin-ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. A simple recipe towards reducing hallucination in neural surface realisation. pages 2673–2679.
- Weili Nie, Nina Narodytska, and Ankit Patel. 2018. Relgan: Relational generative adversarial networks for text generation.
- Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. 2017. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*.

Marc’ Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Van-Khanh Tran and Le-Minh Nguyen. 2017. Natural language generation for spoken dialogue system using rnn encoder-decoder networks. *arXiv preprint arXiv:1706.00139*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. *arXiv preprint arXiv:1603.01232*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2017. Adversarial neural machine translation. *arXiv preprint arXiv:1704.06933*.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. pages 7287–7298.

Steve J Young. 2000. Probabilistic methods in spoken–dialogue systems. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 358(1769):1389–1402.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2019. Multi-task learning for natural language generation in task-oriented dialogue.

MR	name[Wildwood], eatType[restaurant], food[Indian], area[riverside], familyFriendly[no], near[Raja Indian Cuisine]
Ref.	Located in the riverside area near the Raja Indian Cuisine, Wildwood offers Indian food and a restaurant. It is not family friendly.
AdvNLG	Wildwood is an Indian restaurant in the riverside area near Raja Indian Cuisine . It is not family friendly.
-Adv.	Wildwood is a restaurant providing Indian food. It is located in the riverside . It is near Raja Indian Cuisine .
NLG-LM	Wildwood is a restaurant providing Indian food. It is located in the riverside . It is near Raja Indian Cuisine .
Comment	Only AdvNLG places the two pieces of location information “riverside” and “near Raja Indian Cuisine” together, which is aligned with human language patterns.
MR	name[The Cricketers], eatType[restaurant], food[English], priceRange[high], customer rating[1 out of 5], area[city centre], familyFriendly[yes], near[Café Rouge]
Ref.	The Cricketers, an English restaurant located near Café Rouge in the city centre, offers food at high price range. Although it has a customer rating of 1 out of 5, it also is children friendly.
AdvNLG	The Cricketers is a child friendly English restaurant in the city centre near Café Rouge. It has a high price range and a customer rating of 1 out of 5 .
-Adv.	The Cricketers is a restaurant located in the city centre near Café Rouge. It is a high priced restaurant that serves English food. It is rated 1 out of 5 and is children friendly .
NLG-LM	The Cricketers is a high priced English restaurant located in the city centre near Café Rouge. It has a customer rating of 1 out of 5 and is child friendly .
Comment	AdvNLG model naturally put the negative aspects like “high price” and “rating 1 out of 5” together with conjunction “and”, whereas both -Adv. and NLG-LM juxtapose negative aspect (low customer rating) and positive aspect (kid-friendly) in one sentence, which appears contradictory.
MR	name[The Plough], eatType[restaurant], food[Chinese], priceRange[cheap], area[riverside], familyFriendly[yes], near[Raja Indian Cuisine]
Ref.	The Plough is a cheap Chinese restaurant located riverside by Raja Indian Cuisine. It is a family friendly establishment.
AdvNLG	The Plough is a cheap Chinese restaurant in the riverside area near Raja Indian Cuisine . It is family friendly.
-Adv.	The Plough is a cheap family friendly restaurant that serves Chinese food . It is located in the riverside area near Raja Indian Cuisine.
NLG-LM	The Plough is a restaurant providing Chinese food in the cheap price range. It is located in the riverside . It is near Raja Indian Cuisine .
Comment	AdvNLG places “Chinese” immediately before “restaurant”, and this is in line with the human reference. And NLG-LM model has two less connected sentences at the end.

Table 4: Example of predictions on E2E-NLG by reference, NLG-LM model and our model AdvNLG with and without adversarial training. As E2E-NLG only has *inform* dialogue act, we show the meaning representation (MR).

A Sequence-to-sequence Approach for Numerical Slot-filling Dialog Systems

Hongjie Shi

Megagon Labs, Tokyo, Japan, Recruit Co., Ltd.

shi.hongjie@megagon.ai

Abstract

Dialog systems capable of filling slots with numerical values have wide applicability to many task-oriented applications. In this paper, we perform a particular case study on the `number_of_guests` slot-filling in hotel reservation domain, and propose two methods to improve current dialog system model on 1. numerical reasoning performance by training the model to predict arithmetic expressions, and 2. multi-turn question generation by introducing additional context slots. Furthermore, because the proposed methods are all based on an end-to-end trainable sequence-to-sequence (seq2seq) neural model, it is possible to achieve further performance improvement on growing dialog logs in the future.

1 Introduction

Task-oriented dialog systems which assist users to complete tasks like hotel reservation, are drawing great attentions among both research and industry. Compared to conventional pipelined system, recently emerging end-to-end trainable dialog systems are showing many favorable characteristics — because of the neural models that directly learn from chatlogs of human-to-human conversation employed, such systems hold the promise of low data preparation cost, flexible response generation and the ability to evolve with new data.

In this work, we are going to explore the possibility to bring the end-to-end trainable dialog system model to the hotel reservation chatbot application, where we encounter two new problems: 1. numerical slots-filling and 2. multi-turn dialog management. To the best of our knowledge, both of them can not be fully solved using currently available end-to-end systems. In this paper, we will focus on these two problems and propose possible workarounds which can lead to satisfactory results.

2 Problem description

The hotel reservation application requires a dialog system to fill three slots with integer — number of adults (`slot:num_adult`), number of primary school children (aged 6-12) (`slot:num_c6_12`), and number of preschool children (aged 0-5) (`slot:num_c0_5`). These numerical slots are necessary because the applicable room plan (number of beds, quantity of amenities) and the pricing (food cost etc.) vary on the number of adults and children. This numerical slot-filling problem is also widely applicable to other domains such as restaurant reservation or flight booking, with slightly different slot configurations.

The challenges of building such dialog system mainly lie in two aspects. First challenge is the difficulty in the numerical slot value inference. Unlike most task-oriented dialog systems or datasets such as [Wen et al. \(2016\)](#); [Henderson et al. \(2013\)](#), where the slot filling can be either solved as a named entity extraction problem or a multi-label classification problem, the numerical slot-filling requires additional reasoning and calculation. For examples the simple expression “*My wife and me*” means 2 adults, and “*4 including 1 baby*” implies 3 adults. And moreover, the numerical inference sometimes involves with multi-turn dialog context, which brings to the second challenge.

Second challenge is the multi-turn dialog management. Many previous task-oriented dialog systems are designed in a turn-wise manner e.g. [Lei et al. \(2018\)](#) — the systems ask the question for particular slot in each turn and expect user to give explicit answer within that turn. If no exact slot value can be extracted from the response, the system will simply repeat the same question. This behavior is unfavorable for the numerical slot-filling, because of the likely ambiguity in the user responses. For example, no target slot value can be determined

from the user response “*4 people including 2 kids*”, while human agent may ask drill-down questions such as “*How old are the children*” to address this ambiguity. To achieve this human-level conversation, a dialog system capable of managing multi-turn strategy such as asking drill-down questions, is desirable.

3 Present methods

Several end-to-end model architectures have been proposed for task-oriented dialog system. [Wen et al. \(2016\)](#) proposed a modularly connected neural networks to enable end-to-end training. Later [Lei et al. \(2018\)](#) simplified this architecture to a single sequence-to-sequence model (SEQUICITY), which not only reduced the training cost but also improved the performance. More recently more advanced model like HaGAN has been applied to end-to-end learning ([Fang et al., 2019](#)). [Wu et al. \(2019\)](#) also explored the possibility of applying recent large pre-trained language model such as BERT and GPT-2 to the task-oriented dialog system.

After survey and review on different models, we consider the SQUICITY framework a particularly good point to start because of its simplicity and extendability. Its key idea is to encode the dialog states (slot values) into a text format which can be concatenated to the target utterances, so that any seq2seq models can handle both slot-filling and language generation at the same time. In this way, the model complexity and the training procedure are greatly simplified (refer to original paper for more details). Recently published T5 model ([Rafel et al., 2019](#)) also demonstrates the promising performance and the wide applicability of such text-to-text format training. Therefore we consider this SEQUICITY framework using seq2seq model has great potential and lower maintenance cost for commercial applications, and in this paper we chose this framework as our base model.

4 Proposed methods

4.1 Slot-filling with numerical reasoning

In order to enable the seq2seq model to perform numerical reasoning, we train the model to predict arithmetic expressions instead of numeric values. For example for the utterance “*three men and two women*”, we modify the target output to be ‘3 +

2’¹ instead of the numeric value ‘5’ during training. This encourages the seq2seq model to simply copy values from the input sentence², rather than manipulate the number directly. This method is also inspired by recent state-of-the-art models from the Discrete Reasoning Over Passages (DROP) dataset ([Dua et al., 2019](#)), where most models are trained to predict numerical spans and math operations respectively ([Ran et al., 2019; Andor et al., 2019](#)). Intuitively, by doing so, we can achieve better generalization performance because it can easily handle unseen combinations of different numbers and math operations.

4.2 Multi-turn dialog management for ambiguous user utterances

The original SQUICITY model only takes one single turn of previous utterance and slot values as model input for the response generation. This mechanism reduces the training cost, however, may hinder the model from learning multi-turn dialog strategy. For example in the following dialog:

Agent:	How many people is the reservation for?
User:	Four people including two kids. (1) (total_num:4 num_child:2)
Agent:	...
User:	...
Agent:	How old are the two children? (2)
User:	One is 5 and the other is 8. (num_adult:2 num_c6_12:1 num_c0_5:1)

It will not be possible for system to ask questions like (2) without being aware of earlier user utterance (1). To address this problem, we use additional slots to track down all necessary dialog context. We call them *context slots*. In this particular example, we use two context slots — `total_num` slot with value of 4 and `num_child` slot with value of 2 to track the information mentioned in user utterance (1). We treat these context slots just like other numerical slots — they will be carried on to the next turn’s input until the goal is achieved, so that the model can refer to them at any position of the dialog. With the help of context slots, the dialog system can generate context-aware questions with less effort, and also is able to learn multi-turn dialog strategy from less data.

¹This output consists of three tokens, which are ‘3’, ‘+’ and ‘1’

²Same as original SEQUICITY paper, we choose CopyNet ([Gu et al., 2016](#)) as our seq2seq architecture, so that some of the output tokens can be simply copied from the input sequence

5 Experiment and results

5.1 Slot-filling with numerical reasoning

To collect training and evaluation dataset with arithmetic expression, crowd sourcing service was used. We ask crowd workers to compose utterances using numbers given in the instruction, while avoiding directly including the answer in the sentence, so that each collected utterance requires numerical reasoning for inference. Samples of collected data are shown below³ (with target slot values shown in the brackets):

Agent: How many people is the reservation for?
User: One adult and one middle school child. (<u>num_adult</u> :1+1 <u>num_c6_12</u> :0 <u>num_c0_5</u> :0)
Agent: How many people is the reservation for?
User: Four including one elementary school child. (<u>num_adult</u> :4-1 <u>num_c6_12</u> :1 <u>num_c0_5</u> :0)
Agent: Is the reservation for 4 adults?
User: No, we have one 8-year-old child and one 3-year-old child. (<u>num_adult</u> :4-1-1 <u>num_c6_12</u> :1 <u>num_c0_5</u> :1)
Agent: Are there 3 adults and 1 preschool child?
User: Oh, we have one more preschool child. (<u>num_adult</u> :3 <u>num_c6_12</u> :0 <u>num_c0_5</u> :1+1)

All utterances are trained with the arithmetic expressions as shown underlined above (more training details can be found in Appendix A). We also compared the proposed method to training the model with the numerical value directly. The result is summarized in the table below:

training / test data # ⁴	Numerical values F1	Arithmetic expressions F1
899 / 1230	0.48	0.89
1829 / 1230	0.72	0.93
3666 / 1230	0.90	0.94

Our result shows that the proposed method (predicting algorithmic expression) outperforms predicting numeric value by a huge margin when the training data size is small. However, by increasing the training data size, the performance gap between two methods can be greatly reduced. These results can be interpreted as the following two reasons. 1. Algorithmic expression prediction have superior generalization performance for small size of training data, because it can easily handle unseen combination of numbers. On the other hand predicting with numeric value requires the model to

³ All data used in this paper are collected and trained in Japanese. The examples showing here are the English translations.

⁴ All test data are identical. And F1 scores are the weighted average over all three slots.

also learn to manipulate numbers directly , therefore it may need more instances to train. 2. The algorithmic expressions appeared in the dataset is quite simple with limited range and variations. It is possible to train a seq2seq model with pretrained word embedding to be able to do simple calculation. This observation is consist with one recent paper, which also reported the good performance of neural models on addition calculation within the training range (Wallace et al., 2019).

In the real application, we can combine these two models to furthermore boost the performance in a ensemble learning way. And also, when two models give completely different answers, we can also tune the dialog system to confirm with user.

5.2 Multi-turn dialog management for ambiguous user utterances

To achieve dialog management resemble to real human-human conversation, we collected around 900 hotel reservation dialogs from pairs of workers who played agent or user roles. Each dialog covers all topics in hotel reservation, including location, price range, preference and so on. We then analyzed all sub dialog segments concerning total_num/num_child slot from each dialog, and extracted 7 representative drill-down questions as listed below:

- Are all people above middle school students?
- Are there any children in the group?
- Are there any children who are primary school students or below?
- Are all people adults?
- How old is the child?
- If there is any child in the group, could you please tell me their ages?
- Is it for <total_num> adults?

In order to collect more variations of possible user utterances which are applicable to these questions, again we used crowd sourcing service and asked workers to fill in the blank of the dialog below:

Agent: How many people is the reservation for?
User: _____(a)_____
Agent: <one of the questions shown in above list>
User: _____(b)_____
Agent: Alright, so it is <n1> adults, <n2> child (6-12) and <n3> child (0-5).

Example of collected dialogs:

Agent: How many people is the reservation for?
User: For 6 people including kids.
Agent: How old are the children?
User: We have one primary school child, two preschool children.
Agent: Alright, so it is 3 adults, 1 child (6-12) and 2 children (0-5).

Agent utterances	User utterances	Predicted slots	Generated response
How many people is the reservation for? お泊りの人数はお決まりでしょうか？ Are there any children in the family? ご家族の中にお子様はいらっしゃいますか？	Four members of my family. 家族4名です。 No there isn't. いません。	total_num 4 num_adult 4 num_c6_12 0 num_c0_5 0	Are there any children in the group? お子様はいらっしゃいますか? <DIALOG_END>
How many people is the reservation for? お泊りの人数はお決まりでしょうか？ Could you tell me the age of your child? お子様のご年齢を教えていただけますか？	2 adults and 1 child please. 大人2人子供1人でお願いします。 Four years old! 4歳です！	num_adult 2 num_child 1 num_adult 2 num_c6_12 0 num_c0_5 1	How old is the child? お子様のご年齢をお伺いできますか? <DIALOG_END>
How many people is the reservation for? お泊りの人数はお決まりでしょうか？ Are all people adults? 皆様大人の方でいらっしゃいますか？	For 3 people. 3名です。 2 adults and 1 child. 大人2名と子供1名です。	total_num 3 num_adult 2 num_c6_12 0 num_c0_5 1	Are all people adults? 皆様大人の方でいらっしゃいますか? <DIALOG_END>
How old is the child? お子様のご年齢をお伺いできますか？	10 years old. 10歳です。	N/A	N/A

Table 1: Samples of test dialogs with model outputs, including the slot values and next response. The model is trained and predicted in Japanese. Red bold texts are the results considered to be incorrect.

With these data, our model is ready to train for the multi-turn dialog strategy. During the training, turn (a) is trained with context slots (Sect. 4.2) and next-turn agent question. Turn (b) is trained with target slot values consistent with the last agent utterance, and a special token <DIALOG-END>.

To evaluate the model, we extracted 20 dialog segments that contain drill-down questions from actual human-human dialogs as hold-out test dataset, and only train the model with 2000 crowdsourcing collected dialogs. Samples of generated responses from the test data can be found in Table 1. Human evaluation shows that 80% of model generated responses are reasonable (more results including comparison with baseline can be found in Appendix B), however compared to actual human dialogs, the responses generated by model tend to be less diverse. In particular the question which quotes user context — “*Is it for <total_num> adults?*”, rarely appears in the model output, even though it has same number of training data as the other questions. The lack of variation in the generated output is also a common issue that has been studied in previous general-purpose and task-oriented dialog models (Shao et al., 2017; Rajendran et al., 2018).

5.2.1 Agent utterance normalization

Another problem we encountered when training with large data is that, multiple correct next utterances corresponding to the same dialog state may exist in the training corpus. For example, followed by the same user response “*We have three people*”, some of dialog contains next utterance “*Are there any children in the group?*”, while others contain different utterance like “*Is it for 3 adults?*”. This may cause the training difficult to converge, and therefore results in lower train and test accuracy.

To address this problem, Rajendran et al. (2018) proposed a method which uses a combination of supervised learning and reinforcement learning. However for the model we used, we found this reinforcement learning approach unstable and very sensitive to the heuristic determined rewards. After several unsuccessful trials, we decide to simply normalize all agent utterances based on the appearance frequency, so that for each unique dialog state (or user response), only one possible next utterance exists in the training corpus. By doing so, we are able to achieve almost 100% train accuracy and better test accuracy than previous results. This method works but is less sophisticated compared to the reinforcement learning. We will continuously explore alternatives to improve it in the future.

6 Conclusion and future work

In this paper, we proposed two methods for improving the original end-to-end dialog system on numerical slot-filling. By training the model to predict arithmetic expressions, the dialog system can perform numeric reasoning more robustly, and with newly included context slots, the dialog system is able to generate multi-turn questions for ambiguous user responses.

Future work may include extending the current seq2seq network to more recent large-scale pre-trained models such as RoBERTa, as suggested in Talmor et al. (2019), for a better performance in reasoning task. And also the proposed multi-turn dialog management approach should be extensively tested on other slots and domains.

Acknowledgments

I would like to thank Dr. Hidekazu Tamaki for the help of data collection, and Prof. Yuki Arase for helpful research advice.

References

- Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving bert a calculator: Finding operations and arguments with reading comprehension. *arXiv preprint arXiv:1909.00109*.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.
- Ting Fang, Tingting Qiao, and Duanqing Xu. 2019. Hagan: Hierarchical attentive adversarial learning for task-oriented dialogue system. In *International Conference on Neural Information Processing*, pages 98–109. Springer.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.
- Matthew Henderson, Blaise Thomson, and Jason Williams. 2013. Dialog state tracking challenge 2 & 3.
- Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Janarthanan Rajendran, Jatin Ganhotra, Satinder Singh, and Lazaros Polymenakos. 2018. Learning end-to-end goal-oriented dialog with multiple answers. *arXiv preprint arXiv:1808.09996*.
- Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. Numnet: Machine reading comprehension with numerical reasoning. *arXiv preprint arXiv:1910.06701*.
- Louis Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. *arXiv preprint arXiv:1701.03185*.
- Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2019. olmpics—on what language model pre-training captures. *arXiv preprint arXiv:1912.13283*.
- Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. *arXiv preprint arXiv:1909.07940*.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.
- Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2019. Alternating recurrent dialog model with large-scale pre-trained language models. *arXiv preprint arXiv:1910.03756*.

Agent utterances	User utterances	Predicted slots	Generated response
How many people is the reservation for? お泊りの人数はお決まりでしょうか?	There are four. 4人なんですが。	num_c6_12 0 num_c0_5 0	Are there any children in the group? お子様はいらっしゃいますか? <DIALOG_END>
Are there any children in the group? お子様はいらっしゃいますか?	There is one child. 子供は一人です。	N/A	N/A
Can you tell me how old the child is? お子様のご年齢も教えていただけますか?	5 years old. 5歳です。	num_adult 4	Are there any children in the group? お子様はいらっしゃいますか? <DIALOG_END>
Can you tell me how old the child is? お子様のご年齢も教えていただけますか?	5 years old. 5歳です。	num_adult 4 num_c6_12 1 num_c0_5 0	N/A
How many people is the reservation for? お泊りの人数はお決まりでしょうか?	There are four. 4人なんですが。	total_num 4	Are there any children in the group? お子様はいらっしゃいますか?
Are there any children in the group? お子様はいらっしゃいますか?	There is one child. 子供は一人です。	total_num 4 num_child 1	How old is the child? お子様のご年齢をお伺いできますか? <DIALOG_END>
Can you tell me how old the child is? お子様のご年齢も教えていただけますか?	5 years old. 5歳です。	num_adult 3 num_c6_12 0 num_c0_5 1	

Table 2: Samples of evaluation results of baseline #1, baseline #2 and proposed method in order from the top. Red bold texts are the results considered to be incorrect.

A Details of experiments for slot-filling with numerical reasoning

The SEQUICITY framework processes dialog output in two stages: in the first stage, it decodes a text form of slot values, which is called *belief span* (*bspan*) in the original paper; in the second stage, it decodes a machine response conditioning on the belief span decoded in the first stage. The processed input and output of each stage are summarized as below:

1st stage Inputs: Outputs:	$bspan_{t-1} \oplus agent_{t-1} \oplus user_t$ $bspan_t$
2nd stage Inputs: Outputs:	$bspan_{t-1} \oplus agent_{t-1} \oplus user_t \oplus bspan_t$ $agent_t$

Table 3: Two-stage process used in SEQUICITY framework.

where $t - 1$ represents the previous turn, t the current turn and \oplus the concatenation operator. Since in the experiment 5.1 we only examine the model performance on slot-filling, only the first stage above is used. And also, because all data we collected in 5.1 only contain single turn (with no dialog history), $bspan_{t-1}$ is set empty during training and test. Sample of encoded $bspan_t$ is shown below:

slots: num_adult:4-1 num_c6_12:1 num_c0_5:0
bspan: <slot1>_4_-1-<slot2>_1-<slot3>_0

where $<\text{slot1}>$, $<\text{slot2}>$ and $<\text{slot3}>$ are the special tokens for indicating num_adult, num_c6_12 and num_c0_5 slot respectively.

B Compare multi-turn dialog performance with baseline

To compare the proposed method (context slots) with original SEQUICITY framework, we performed additional comparison experiments. Here we modified collected multi-turn dialog's context slots in two ways so that it can be applied with original model: 1. simply delete total_num and num_child slots; 2. move total_num and num_child values to num_adult and num_c6_12 slots if unfilled.

Both of the baseline methods degrade in performance because: 1. deleting context slots causes missing out context information; 2. filling values with other slots causes indistinguishable value interpretation. Same as before, we evaluate 20 human-human dialog sections with both baseline methods. Compared to 80% success rate achieved by the proposed method, baseline #1 and #2 can only reach 20% and 70% respectively. Furthermore, we find that baseline #1 fails in almost all cases, while baseline #2 tends to wrongly generate <DIALOG_END> signal where drill-down question is necessary for more than one turn. As the example shown in Table 2, baseline #2 is able to generate correct drill-down question for the first turn, while fails on the second turn. This is partially because that the model can only access the dialog history by previous belief span (as explained in Table 3), which is inaccurate in this case due to lack of context slots.

Beyond Domain APIs: Task-oriented Conversational Modeling with Unstructured Knowledge Access

Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan,
Behnam Hedayatnia, Yang Liu, Dilek Hakkani-Tur

Amazon Alexa AI, Sunnyvale, CA, USA

{seokhwk,mihaeric,karthgop,behnam,yangliu,hakanit}@amazon.com

Abstract

Most prior work on task-oriented dialogue systems are restricted to a limited coverage of domain APIs, while users oftentimes have domain related requests that are not covered by the APIs. In this paper, we propose to expand coverage of task-oriented dialogue systems by incorporating external unstructured knowledge sources. We define three sub-tasks: knowledge-seeking turn detection, knowledge selection, and knowledge-grounded response generation, which can be modeled individually or jointly. We introduce an augmented version of MultiWOZ 2.1, which includes new out-of-API-coverage turns and responses grounded on external knowledge sources. We present baselines for each sub-task using both conventional and neural approaches. Our experimental results demonstrate the need for further research in this direction to enable more informative conversational systems.

1 Introduction

Traditionally, task-oriented dialogue systems have focused on providing information and performing actions that can be handled only by given databases or APIs. However, in addition to task-focused requests, users also have needs that go beyond what is provided by the backend resources. For example, while most virtual assistants can help users book a hotel, a restaurant or movie tickets, they fall short of answering potential follow-up questions users may have, such as: where to park vehicles; whether they are allowed to bring pets or children to the reserved place; or what the cancellation policy is. No API/DB entry is usually available to handle such requests. On the other hand, relevant domain knowledge is already available on web pages in the form of descriptions, FAQs and customer reviews for many of these out-of-coverage scenarios. Since current dialogue systems don't

incorporate these external knowledge sources into task-oriented conversational modeling, users need to visit the websites by themselves to find out any additional information beyond API/DB coverage, making conversational interactions inefficient.

In this work, we propose a new conversational modeling task towards frictionless task-oriented scenarios, where the flow of the conversation does not break when users have requests that are out of the coverage of APIs/DB but potentially are already available in external knowledge sources. Inspired by recent studies on knowledge-grounded conversational modeling (Zhou et al., 2018; Dinan et al., 2018; Galley et al., 2019; Gopalakrishnan et al., 2019), our proposed task aims to develop end-to-end dialogue systems to understand relevant domain knowledge, and generate system responses with the selected knowledge. Different from previous work on social conversations (Ritter et al., 2011; Vinyals and Le, 2015; Serban et al., 2017), this task addresses task-oriented conversations grounded on fine-grained domain-level or entity-level knowledge sources related to given dialogue contexts.

Figure 1 shows an example conversation with unstructured knowledge access. The user utterances at turns $t = \{3, 7\}$ and $t = \{11, 15\}$ request the policy details about bringing pets and making payments, respectively, which are out of the coverage of the structured domain APIs. On the other hand, the relevant knowledge contents can be found from the external sources as in the rightmost column which includes the QA snippets from the FAQ lists for each corresponding entity within domains such as train, hotel, or restaurant. With access to these unstructured external knowledge sources, the agent managed to continue the conversation with no friction by responding adequately at the turns $t = \{4, 8, 12, 16\}$.

t	Speaker	Utterance (u_t)	Relevant Knowledge Snippets from FAQs
1	User	I need a train from King's Lynn to Cambridge arriving by 17:45 on Sunday.	
2	Agent	I have train TR6003 leaving cambridge at 5:11 and arriving at 5:58 on sunday	
3	User	I also need to bring my dog. Do they allow pets?	Q: Can I bring my pets to trains? A: We happily welcome dogs and cats up to 20 pounds.
4	Agent	Yes, you can travel with your dog up to 20 pounds.	
5	User	That sounds great. I also need help finding a place to stay in the South.	
6	Agent	I have 2 options available. The Lensfield Hotel and Gonville Hotel.	
7	User	Do either of them allow to stay with my dog?	Q1: Can I bring my dog to Gonville Hotel? A1: Pets are permitted. But charges may be applicable.
8	Agent	You could stay with your dog at Gonville with a fee. Would you like me to book it?	Q2: Can I bring my dog? A2: Pets are not allowed at the Lensfield Hotel.
9	User	Maybe later. How about a place to eat nearby? Chinese food would be great.	
10	Agent	The peking restaurant is a nice place. Do you need reservations?	
11	User	Before that, could you confirm that this restaurant accepts AMEX?	Q: What type of payments are accepted?
12	Agent	Unfortunately, the Peking restaurant accepts cash only. Would it work?	A: Peking Restaurant accepts cash only.
13	User	Okay. Can you book a table for 4 at 18:30 on Monday, please?	
14	Agent	Booking was successful. Do you have any other question?	
15	User	What about the hotel? Can I use my credit card there?	Q: What credit cards are accepted by Gonville Hotel?
16	Agent	Yes, Gonville Hotel accepts all major credit cards including AMEX.	A: You can use AMEX, Visa and Mastercard.

Figure 1: Examples of task-oriented conversations with unstructured knowledge access

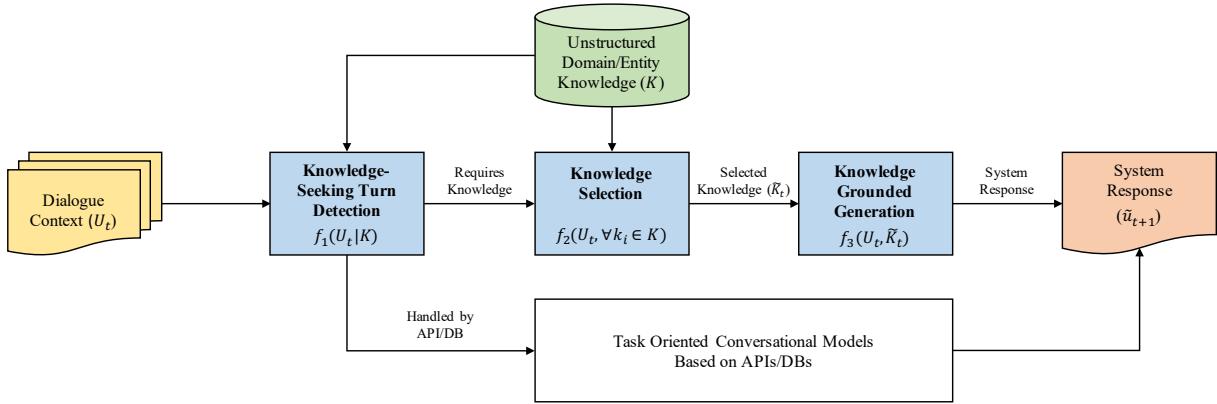


Figure 2: A baseline architecture for task-oriented conversational modeling grounded on unstructured knowledge

2 Related Work

Task-oriented dialogue systems aim to enable users to complete tasks by interacting with an automated agent in natural language (Young et al., 2013). These systems typically convert user utterances to a semantic representation (such as domain, intent, and slots (Tur and De Mori, 2011)) based on what is used by the backend resources (such as APIs) that accomplish the tasks. At each turn, the dialogue system decides the next action to take based on the estimated dialogue state as well as any results or responses from the backend resources (Levin et al., 2000; Singh et al., 2002; Williams and Young, 2007). The next action, which is typically in the form of a semantic frame formed of dialogue acts, arguments and values, is converted to a natural language response to the user by natural language generation (Perera and Nand, 2017).

On the other hand, social conversational systems typically follow an end-to-end approach, and aim to generate target responses based on the previous conversation context (Ritter et al., 2011; Vinyals and Le, 2015; Serban et al., 2017). Ghazvininejad

et al. (2018) proposed an extension to these models that grounds the responses on unstructured, textual knowledge, by using end-to-end memory networks where an attention over the knowledge relevant to the conversation context is estimated. Along similar lines, Liu et al. (2018) used pattern matching, named entity recognition and linking to find facts relevant to the current dialogue and other related entities from a knowledge base. Zhou et al. (2018) proposed both static and dynamic graph attention mechanisms for knowledge selection and response generation, respectively, using knowledge graphs. More recently, Dinan et al. (2018) and Gopalakrishnan et al. (2019) both have publicly released large conversational data sets, where knowledge sentences related to each conversation turn are annotated. Our proposed task, data, and baseline models in this work differ from these studies in the following aspects: we target task-oriented conversations with more clear goals and explicit dialogue states than social conversations; and we aim to incorporate task-specific domain knowledge instead of commonsense knowledge.

The other line of related work is machine reading comprehension which aims to answer questions given unstructured text (Richardson et al., 2013; Hermann et al., 2015; Rajpurkar et al., 2016) and has later been extended to conversational question answering (Choi et al., 2018; Reddy et al., 2019). In our work, the document required to generate a response needs to be identified according to the conversation context. The responses are also different in that, rather than plain answers to factual questions, we aim to form factually accurate responses that seamlessly blend into the conversation.

3 Problem Definition

We define an unstructured knowledge-grounded task-oriented conversational modeling task based on a simple baseline architecture (Figure 2) which decouples turns that could be handled by existing task-oriented conversational models with no extra knowledge and turns that require external knowledge resources. In this work, we assume that a conventional API-based system already exists and focus on the new knowledge access branch which takes a dialogue context $U_t = \{u_{t-w+1}, \dots, u_{t-1}, u_t\}$ and knowledge snippets $K = \{k_1, \dots, k_n\}$, where u_i is the i -th utterance in a given dialogue, t is the time-step of the current user utterance to be processed, w is the dialogue context window size.

Our proposed task aims to generate a context-appropriate system response \tilde{u}_{t+1} grounded on a set of relevant knowledge snippets $\tilde{K} \subset K$. The remainder of this section presents the detailed formulations of the following three sub-tasks: ‘Knowledge-seeking Turn Detection’, ‘Knowledge Selection’, and ‘Knowledge-grounded Response Generation’.

3.1 Knowledge-seeking Turn Detection

For each given turn at t , a system first needs to decide whether to continue an existing API-based scenario or trigger the knowledge access branch. We call this task *Knowledge-seeking Turn Detection*. This problem is defined as a binary classification task formulated as follows:

$$f_1(U_t|K) = \begin{cases} 1 & \text{if } \exists k \in K \text{ satisfies } u_t, \\ 0 & \text{otherwise,} \end{cases}$$

which we assume that every turn can be handled by either branch in this work. For the examples in Figure 1, $f_1(U_t|K) = 1$ for the knowledge-seeking

turns at $t = \{3, 7, 11, 15\}$, while $f_1(U_t|K) = 0$ for the other user turns at $t = \{1, 5, 9, 13\}$.

3.2 Knowledge Selection

Once a given user turn at t is determined as a knowledge-seeking turn by $f_1(U_t|K)$, it moves forward with *Knowledge Selection* to sort out the relevant knowledge snippets. This task takes each pair of U_t and $k_i \in K$ and predicts whether they are relevant or not as follows:

$$f_2(U_t, k_i) = \begin{cases} 1 & \text{if } k_i \in K \text{ is relevant to } U_t, \\ 0 & \text{otherwise.} \end{cases}$$

Different from other information retrieval problems taking only a short single query, this knowledge selection task must be highly aware of the dialogue context. For example, u_3 and u_7 themselves in Figure 1 share the same question type with similar surface form, but the relevant knowledge snippets would vary depending on their dialogue states across different domains. Even within a single domain, fine-grained dialogue context needs to be taken into account to select proper knowledge snippets corresponding to a specific entity, for example, ‘Peking Restaurant’ and ‘Gonville Hotel’ for u_{11} and u_{15} against any other restaurants and hotels, respectively.

Since more than one knowledge snippet can be relevant to a single turn, as for u_7 in Figure 1, we form a task output \tilde{K} including all the positive knowledge snippets from $f_2(U_t, k)$, as follows:

$$\tilde{K}_t = \{k_i | k_i \in K \wedge f_2(U_t, k_i) = 1\} \subset K.$$

3.3 Knowledge-grounded Generation

Finally, a system response \tilde{u}_{t+1} is generated based on both dialogue context U_t and the selected knowledge snippets \tilde{K}_t , as follows:

$$f_3(U_t, \tilde{K}_t) = \tilde{u}_{t+1}.$$

Each generated response is supposed to provide the user with the requested information grounded on the properly selected knowledge sources. In addition, the response should be naturally connected to the previous turns. The knowledge-grounded responses in Figure 1 focus not only on delivery of the information by knowledge access, but also maintain natural conversation. For example, the responses at $t = \{4, 8\}$ paraphrase written sentences into a colloquial style, the responses at $t = \{4, 16\}$ acknowledge before giving a statements, the responses at $t = \{8, 12\}$ ask a follow-up question to the user.

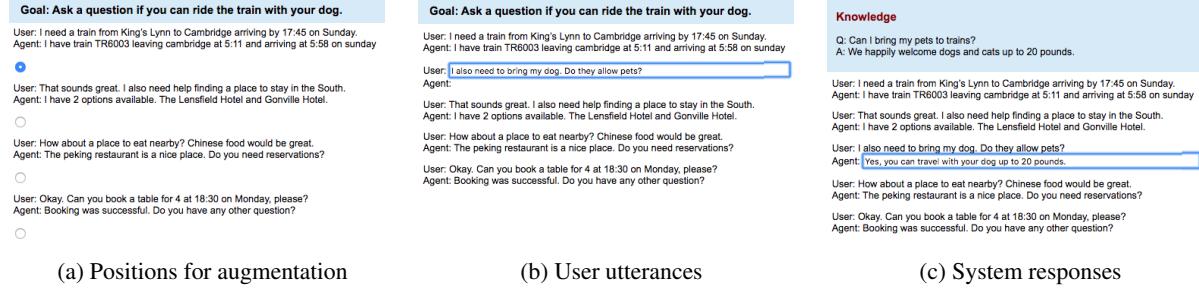


Figure 3: Crowdsourcing user interfaces for MultiWOZ data augmentation with knowledge access turns

4 Data

To address the proposed research problems, we collected an augmented version of MultiWOZ 2.1 (Budzianowski et al., 2018; Eric et al., 2019) with out-of-API-coverage turns grounded on external knowledge sources beyond the original database entries. This was incrementally done by the following three crowdsourcing tasks.

First, crowd workers were given a dialogue sampled from the original MultiWOZ 2.1 conversations and asked to indicate an appropriate position to insert a new turn about a selected subject from external knowledge categories (Figure 3a). This task aims to collect user behaviors about when to ask a knowledge-seeking question for a given subject. It corresponds to the knowledge-seeking turn detection sub-task in Section 3.1.

Then, they were asked to write down a new user utterance at each selected position in the first task to discuss about a given corresponding subject (Figure 3b), which is for both knowledge-seeking turn detection (Section 3.1) and knowledge selection (Section 3.2) sub-tasks. In order to collect various expressions, a single task with the same dialogue context and knowledge category was assigned to multiple crowd workers in parallel.

Finally, we collected the agent’s response to each question collected in the previous step. In this task (Figure 3c), crowd workers were given external knowledge sources for each category and asked to convert them into a system response which is more colloquial and coherent to both the question and dialogue context. This task aims at knowledge-grounded response generation (Section 3.3).

Our proposed pipeline for data collection has the following advantages over Wizard-of-Oz (WoZ) approaches. First, it is more efficient and scalable, since every task can be done by a single crowd worker independently from others, while WoZ requires to pair up two crowd workers in real time.

Split	# dialogues	# augmented turns	# utterances
Train	8,438	7,169	127,894
Valid	1,000	923	16,594
Test	1,000	980	16,704
Total	10,438	9,072	161,192

Table 1: Statistics of the data divided into training, validation, and test purposes. The total number of utterances includes both the original and augmented turns.

Domain	Domain-level		Entity-level
	# snippets	# entities	# snippets
Hotel	24	27	477
Restaurant	8	81	401
Train	20	-	-
Taxi	8	-	-
Total	60	108	878

Table 2: Statistics of domain-/entity-level knowledge snippets collected from FAQ webpages

This aspect enables us to have more control in the whole process compared to the end-to-end data collection entirely by crowd workers from scratch. Furthermore, the intermediate outcomes from each phase can be utilized to build conversational models with no additional annotation.

Table 1 shows the statistics of the collected data sets. A total of 9,072 utterance pairs are newly collected in addition to the original MultiWOZ dialogues, each of which is linked to corresponding knowledge snippets among 938 question-answer pairs (Table 2) collected from the FAQ webpages about the domains and the entities in MultiWOZ databases. Figure 4 shows the length distribution of the augmented utterances. Similar to the original MultiWOZ (Budzianowski et al., 2018), the agent responses are longer than the user utterances, which have 12.45 and 9.85 tokens on average spoken by agents and users, respectively. Figure 5 presents the distribution of trigram prefixes of the augmented user utterances with various types of follow-up questions that go beyond the coverage of domain APIs.

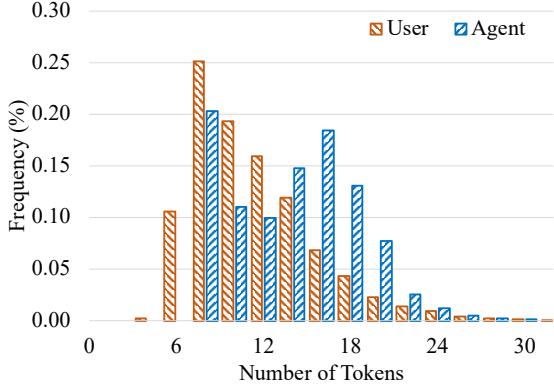


Figure 4: Distribution of number of tokens of the augmented utterances

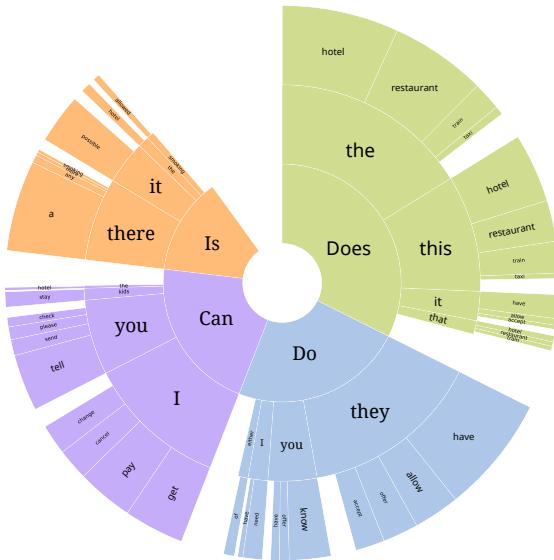


Figure 5: Distribution of trigram prefixes of the augmented user utterances

5 Methods

In this section, we present baseline methods for the problems defined in Section 3. Specifically, we introduce both a non-machine learning approach and a neural baseline model for each sub-task.

5.1 Knowledge-seeking Turn Detection

For the knowledge-seeking turn detection, we compare two baselines with unsupervised anomaly detection and supervised classification methods.

5.1.1 Unsupervised Anomaly Detection

In the first baseline, we consider the task as an anomaly detection problem that aims to identify the turns that are out of the coverage of conventional API-based requests. Given the assumption that there is no knowledge-seeking turn available in

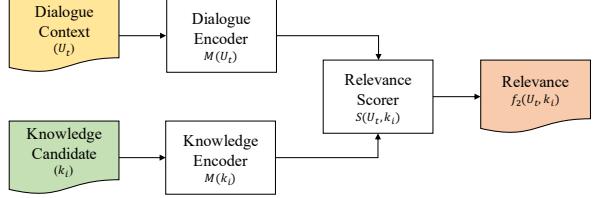


Figure 6: Retrieval baseline for knowledge selection

most task-oriented dialogue data, we applied an unsupervised anomaly detection algorithm, Local Outlier Factor (LOF) (Breunig et al., 2000). The algorithm compares the local densities between a given input instance and its nearest neighbors. If the input has a significantly lower density than the neighbors, it is considered an anomaly.

We built a knowledge-seeking turn detector with the LOF implementation in PyOD (Zhao et al., 2019) with its default configurations. The system includes all the user utterances in the original MultiWOZ 2.1 training set. Every utterance in both training and test sets was encoded by the uncased pre-trained BERT (Devlin et al., 2019) model.

5.1.2 Neural Utterance Classification

If training data is available for the knowledge-seeking turn detection, the most straightforward solution will be training a binary classifier in a supervised manner. In this experiment, we fine-tuned the uncased pre-trained BERT (Devlin et al., 2019) model on the training data in Section 4. The model takes each single user utterance u_t as an input and generates the utterance representation as the final layer output for $[CLS]$ which is a special token in the beginning of the input sequence. We added a single layer feedforward network on top of the utterance embeddings, which was trained with binary cross-entropy loss for three epochs. We used a mini-batch size of 128 with truncated utterances up to 256 tokens.

5.2 Knowledge Selection

In our experiments, we consider two variants of the knowledge selector: unsupervised knowledge-retrieval baselines and supervised neural Transformer architectures.

5.2.1 Unsupervised Knowledge Retrieval

First, we propose the unsupervised knowledge selection baselines using information retrieval (IR) algorithms (Figure 6). Let us denote an encoder function M mapping the concatenation of all the

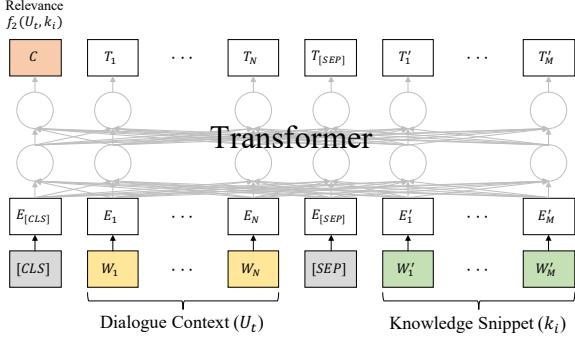


Figure 7: BERT-based knowledge selection baseline

sentences in a query or a document to a fixed-dimensional weight vector. In this work, we take the dialogue context U_t as a query and each knowledge snippet k_i as a candidate document. When scoring entity-level knowledge, we also add the name of the entity to each document k_i being scored as this helps differentiate among potentially ambiguous knowledge contents that may be applicable to multiple entities.

Our IR model then computes the following cosine similarity score per knowledge snippet:

$$S(U_t, k_i) = \cos(M(U_t), M(k_i)),$$

where we finally take the most relevant document as a selected knowledge in the following fashion:

$$f_2(U_t, k_i) = \begin{cases} 1 & \text{if } i = \operatorname{argmax}_j S(U_t, k_j), \\ 0 & \text{otherwise.} \end{cases}$$

We use two types of standard IR baselines: a TF-IDF (Manning et al., 2008) and a BM25 (Robertson and Zaragoza, 2009) system. We also consider another IR baseline that employs an uncased pretrained BERT model as a static utterance encoder. In this baseline, we encode U_t and each k_i separately and then compute the cosine similarity between the pooled utterance outputs.

5.2.2 Neural Relevance Classification

We also employ a BERT-based (Devlin et al., 2019) neural model as a baseline knowledge selection system. In particular, we train a binary classification model (Figure 7) over a pair of encoded texts as is done in prior Transformer sentence relationship models (Nogueira and Cho, 2019). The model takes the concatenation of the utterances in U_t and the sentences in k_i as an input instance. We use the final layer output C at the same position to the $[CLS]$ token as input to a single layer feedforward

network to obtain a probability s_i that the k_i is relevant to the given dialogue context U_t .

We finetune a pretrained BERT model using a binary cross-entropy loss as follows:

$$L = - \sum_{i \in I_{pos}} \log(s_i) - \sum_{i \in I_{neg}} \log(1 - s_i),$$

where I_{pos} refers to the set of knowledges that are relevant for the given dialogue context and I_{neg} refers to those that are not.

During training of the knowledge classifier, we experimented with sampling methods of negative knowledge candidates to be paired with a given dialogue context. For dialogues annotated with domain-level knowledge, we chose negative candidates by sampling other documents in the same domain as the annotation. For entity-level knowledge dialogues, we chose negative candidates by sampling other documents from the same entity as the provided annotation. We built models in which the number of negative candidates for each positive example was varied from 1 to 13 in increments of 4 and found the best-performing model used 5 negative candidates for each positive candidate.

5.3 Knowledge-grounded Generation

In this section, we propose both extractive and generative approaches for the knowledge-grounded response generation task.

5.3.1 Answer Extraction

The simplest method for knowledge-grounded response generation is to output a part of the selected knowledge snippets. In this experiment, we developed an answer extraction baseline with the following heuristics:

- If multiple knowledge snippets are related to a given turn, randomly pick one of them. Otherwise, a sole snippet is taken as the source for answer extraction.
- If the target snippet includes multiple paragraphs, extract only the first paragraph as a system response. Otherwise, the whole paragraph is considered as the output.

5.3.2 Neural Response Generation

Given the tremendous interest and success in leveraging large pre-trained language models for downstream NLP tasks in the community, our neural baseline leverages the Generative Pre-trained Transformer (GPT-2) model (Radford et al., 2019). We

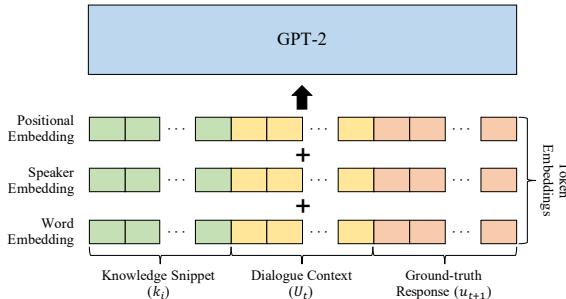


Figure 8: Input representation for GPT-2 w/ knowledge

Method	Acc	P	R	F
Anomaly Detection	0.852	0.393	0.581	0.469
Classification	0.998	0.989	0.994	0.991

Table 3: Comparisons of the knowledge-seeking turn detection performances between two baselines

fine-tuned the GPT-2 *small* model with a standard language modeling objective on our dataset, using both the knowledge-augmented and regular system turns as target sequences. To show the influence of knowledge, we compared two variants of models with different inputs, as follows:

- GPT-2 w/o knowledge: no knowledge was used during fine-tuning.
- GPT-2 w/ knowledge: the ground-truth knowledge snippets were concatenated to each input dialog context (Figure 8) for fine-tuning.

We used the *transformers* library (Wolf et al., 2019a)¹ to fine-tune the models for a fixed number of 3 epochs with a truncation window of 256 tokens for both dialog context U_t and knowledge snippet k_i . We used a train batch size of 2, performed gradient accumulation every 8 steps and gradient clipping with a max norm of 1.0, used the Adam optimizer and linearly decayed the learning rate from 6.25e-5 to 0 during fine-tuning.

We added special tokens for both speakers *user* and *agent* to our vocabulary, initialized their parameters randomly and learned them during fine-tuning. We enriched the corresponding turns in the input with speaker embeddings at a token-level by identifying their token types, exactly as described in (Wolf et al., 2019b). We used top- k , top- p nucleus sampling with temperature T (Holtzman et al., 2019) for decoding, where $k = 0$, $p = 0.9$ and $T = 0.7$. We also set a maximum decode length of 40 tokens.

¹<https://huggingface.co/transformers/>

Method	MRR@5	R@1	R@5
Retrieval (TF-IDF)	0.618	0.511	0.807
Retrieval (BM25)	0.611	0.498	0.827
Retrieval (BERT)	0.226	0.128	0.428
Classification (BERT)	0.891	0.834	0.976

Table 4: Comparisons of the knowledge selection performances by retrieval and classification methods

6 Evaluation

6.1 Knowledge-seeking Turn Detection

First, we evaluated the knowledge-seeking turn detection performances of unsupervised anomaly detection (Section 5.1.1) and supervised neural classification (Section 5.2.2) methods. Both models were built on all the user utterances in the training set and evaluated on the test set user turns in accuracy, precision, recall, and F-measure.

Table 3 shows that the unsupervised baseline has a limitation in distinguishing between API-based and knowledge-seeking turns, especially with many false positives. On the other hand, the neural classifier achieved almost perfect performance in all the metrics. Nevertheless, this utterance classifier may work well when restricted only to this data set or similar, due to lack of knowledge or API details incorporated into the model. There is much room for improvement in making the model more generalizable to unseen domains or knowledge sources.

6.2 Knowledge Selection

Knowledge selection was evaluated using a number of standard IR metrics including recall (R@1 and R@5), and mean reciprocal rank (MRR@5). For domain-knowledge dialogues, our total candidate set included all domain knowledges for the annotated domain, and for entity-knowledge dialogues our total candidate set included all entity knowledges for the annotated entity.

Table 4 shows that our bag-of-words IR baselines (Section 5.2.1) outperformed the static BERT encoder across all three metrics. However, the neural classifier model (Section 5.2.2) significantly outperformed the IR baselines, demonstrating the efficacy of downstream fine-tuning of large pre-trained neural representations. That being said, there is still a substantial performance gap in the R@1 and MRR@5 metrics, leaving room for further research into knowledge selection on this data.

Method	PPL	Unigram F1	Div. ($n = 1$)	Div. ($n = 2$)	BLEU-4	METEOR	ROUGE-L
Answer Extraction	-	0.3215	0.0356	0.0892	0.0358	0.2543	0.1769
GPT-2 w/o knowledge	5.0906	0.2620	0.0509	0.1589	0.0559	0.2202	0.1979
GPT-2 with knowledge	4.1723	0.3175	0.0509	0.1559	0.0840	0.2796	0.2403
Human	-	-	0.0806	0.3055	-	-	-

Table 5: Automated evaluation results on knowledge-grounded response generation

Method	Baseline	Appropriateness			Accuracy		
		%W	%L	%Tie	%W	%L	%Tie
Answer Extraction	Human	34.39	59.49	6.12	-	-	-
GPT-2 w/o knowledge	Human	-	-	-	4.59	27.76	67.65
GPT-2 with knowledge	Human	36.02	59.49	4.49	5.31	22.96	71.74
GPT-2 with knowledge	Answer Extraction	56.33	31.02	12.65	-	-	-
GPT-2 with knowledge	GPT-2 w/o knowledge	-	-	-	22.55	17.04	60.41

Table 6: Human evaluation results on knowledge-grounded response generation

6.3 Knowledge-grounded Generation

Responses by answer extraction (Section 5.3.1) and neural generation models (Section 5.3.2) were first evaluated using the following automated metrics: perplexity, unigram F1, n -gram diversity, BLEU-4, METEOR, and ROUGE-L. The evaluation was done only on the augmented turns with the ground-truth knowledge, in order to characterize the models’ ability to handle the external knowledge scenario. Table 5 shows that our generation models achieved better scores than the extractive baseline on most metrics. Especially, the GPT-2 model with knowledge outperformed both the answer extraction baseline and the other GPT-2 variant with no knowledge in BLEU-4, METEOR, and ROUGE-L, which indicates that our proposed neural model generates more human-like responses than the extractive baseline.

In addition, we also performed human evaluations of the generated responses with the following two crowdsourcing tasks:

- **Appropriateness:** given a dialogue context and a pair of responses generated by two methods, crowdworkers were asked to select a more appropriate response to the context.
- **Accuracy:** given a knowledge snippet and a pair of responses generated by two methods, crowdworkers were asked to select a more accurate response to the knowledge.

In both tasks, we presented each instance to three crowdworkers; asked them to choose either response or ‘not sure’ for the cases that are equally good or bad; and took the majority as the final label for the instance. Table 6 shows that our GPT-2 models generated more appropriate responses

than the answer extraction baseline. Comparing between two GPT-2 variants, the model with knowledge provided more accurate information based on explicitly given knowledge than the one without knowledge. However, this accuracy gap between two models is not very big, which depicts the need to add more diversity in knowledge content which cannot be handled just by memorizing facts from the training data.

7 Conclusions

This paper proposed a new task-oriented conversational modeling problem grounded on unstructured domain knowledge, which aims to handle out-of-API coverage user requests. To support research on our proposed tasks, we introduced an augmented version of MultiWOZ 2.1 dialogues with additional knowledge-seeking turns collected given external knowledge sources. We presented baseline methods based both on non-machine learning approaches and neural model architectures.

Furthering this work, we plan to collect more dialogues including different domains, entities, and locales from the original ones for MultiWOZ 2.1. Moreover, this new data set will include not only written conversations, but also spoken dialogues to evaluate the system performances for more realistic scenarios. Then, all the data sets and the baselines will be released for establishing a new public benchmark in dialogue research.

In addition, we will continue to iterate on the models with the following potential enhancements: end-to-end learning instead of the pipelined processing, joint modeling of both knowledge-seeking and API-driven branches, and few shot transfer learning for unseen domains or knowledge sources.

References

- Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Ifígo Casanueva, Stefan Ultes, Osman Ramadhan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.
- Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAAACL-HLT*.
- Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tür. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.
- Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao, and Bill Dolan. 2019. Grounded response generation task at dstc7. In *Proceedings of the AAAI-19 Workshop on Dialog System Technology Challenges*.
- Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Karthik Gopalakrishnan, Behnam Hedayatnia, Qinglei Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. *Proc. Interspeech 2019*, pages 1891–1895.
- Karl Moritz Hermann, Tomas Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.
- Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.
- Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing*, 8(1):11–23.
- Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1498.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.
- Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *ArXiv*, abs/1901.04085.
- Rivindu Perera and Parma Nand. 2017. Recent advances in natural language generation: A survey and classification of the empirical literature. *Computing and Informatics*, 36(1):1–32.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.
- Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.
- Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mc-test: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203.
- Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593. Association for Computational Linguistics.
- Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.
- Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, 16:105–133.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019a. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019b. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Yue Zhao, Zain Nasrullah, and Zheng Li. 2019. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.

A Appendices

A.1 Unstructured Knowledge Sources

Figure 9 and Figure 10 show examples of knowledge snippets used in our data collection for domain- and entity-specific augmented turns, respectively. While domain-level snippets include generic information that could be applicable over all the domain entities, entity-level knowledge varies depending on a given entity even for the same question.

Domain	Hotel
Title	How can I get an invoice?
Body	The property can provide you with an invoice for your stay, so please contact them directly.
Domain	Restaurant
Title	Cancellation
Body	You can cancel a reservation online or call the restaurant directly. Please note that some restaurants have implemented a 24-48 hour cancellation policy.
Domain	Train
Title	Discount Information for Children
Body	One child ages 2-12 is eligible to receive a 50% discount on the lowest available adult rail fare on most trains with each fare-paying adult (age 18+).

Figure 9: Examples of domain-level knowledge

Domain	Hotel
Entity	Gonville Hotel
Title	What is the parking charge?
Body	Parking costs GBP 14 per day.
Domain	Hotel
Entity	Hamilton Lodge
Title	Is there free parking at your lodge?
Body	Hamilton Lodge offers free parking.
Domain	Restaurants
Entity	La Margherita
Title	What credit cards are accepted?
Body	La Margherita takes Visa, Mastercard and AMEX.
Domain	Restaurants
Entity	Peking Restaurant
Title	What type of payments are accepted at this location?
Body	Peking Restaurant accepts cash only.

Figure 10: Examples of entity-level knowledge

A.2 Human Evaluation Tasks

Figure 11 and Figure 12 show the crowdsourcing user interfaces for human evaluations on *appropriateness* and *accuracy*, respectively.

Which final response is more natural and appropriate to be spoken by the agent?

Select an option
Response 1 1
Response 2 2
Not sure 3

Conversation:

User: How about a restaurant that serves Italian food, do you have one that you would recommend?
Agent: How about the zizzi cambridge in the centre?
User: Sure, I would like to book a reservation for that. But first, could you tell me what area the Alexander Bed and Breakfast is in?
Agent: It is in the centre area. Is there anything else that I can do for you?
User: Yes, I still need to book a reservation for 8 people at Zizzi Cambridge.
Agent: What day and time would you like to go to zizzi cambridge?
User: Can you tell me if they have high chairs that babies can sit in?
Response 1:
Agent: There are high chairs available for babies.
Response 2:
Agent: Yes, Zizzi offers high chairs for children. Would you like to select this restaurant?

Figure 11: Crowdsourcing user interface for the human evaluation on appropriateness

Which agent did provide more accurate information based on given the document?

Select an option
Agent 1 1
Agent 2 2
Not sure 3

Document:
Major credit cards are accepted at Fitzbillies Restaurant.

Question:
I need to use my Mastercard. That's ok right?
Agent 1:
Agent: Yes, all major credit cards are accepted at this establishment.
Agent 2:
Agent: Yes, Mastercard is accepted at the restaurant. Is there anything else I can help you with?

Figure 12: Crowdsourcing user interface for the human evaluation on accuracy

A.3 Generated Responses

Figure 13 compares the example responses generated by the proposed methods. GPT-2 models generated more natural and appropriate responses to be spoken by the agent than the answer extraction baseline. Comparing between two GPT-2 variants, the model with knowledge provided more accurate information than the one without knowledge. However, there is still room for improvement in both appropriateness and accuracy compared to the human responses.

Example #1

Dialogue Context

User	I'm also looking for info on a hotel called allenbell.
Agent	Would you like me to book it for you?
User	Am I able to smoke while staying at this hotel?
Relevant Knowledge	
	Q: Can I smoke there?
	A: Allenbell is a non-smoking establishment.
Generated Responses	
Human	The Allenbell does not allow smoking in their hotel. Can I help you answer any other questions?
Extracted	Allenbell is a non-smoking establishment.
GPT-2 w/o k.	Please contact us for more information
GPT-2 with k.	No, they don't allow smoking in the hotel. Would you like to try a different hotel?

Example #2

Dialogue Context

User	Great! I'm also looking for a place to eat in the same area. Something not too expensive, but not cheap.
Agent	There are 21 places in the centre of town. What type of cuisine would you like?
User	I am interested in moderately priced international food.
Agent	I'd recommend the Varsity Restaurant. Would you like to book a table?
Relevant Knowledge	
	Q: Payment options
	A: We prefer cash payment. We accept MasterCard and Visa with a \$20 minimum purchase.
Generated Responses	
Human	Yes. Cash is actually the preferred payment method.
Extracted	We prefer cash payment. We accept MasterCard and Visa with a \$20 minimum purchase.
GPT-2 w/o k.	Yes, all major credit cards are accepted. Would you like me to make a reservation for you?
GPT-2 with K.	Yes, the restaurant accepts cash. Do you need anything else?

Example #3

Dialogue Context

User	Can you help me find a restaurant that serves African food with a moderate price range please
Agent	I am sorry, there aren't any options available. May I ask if there is another type of restaurant you would be interested in?
User	Yes how about Asian food in the same price range?
Agent	Yes I have the Yippee Noodle Bar in the center of town on King street in the moderate price range. They serve Asian cuisine. Is there anything else I can do for you?
User	Will you please book a table for 6 at 13:45 on Friday?
Agent	Booking was successful. The table will be reserved for 15 minutes. Reference number is : 3GLLHSNN.
Relevant Knowledge	
	Q: What parking options are available at your location?
	A: Only street parking is available.
Generated Responses	
Human	There is only parking on the street. Anything else I can do for you?
Extracted	Only street parking is available.
GPT-2 w/o k.	No, the restaurant does not offer parking for parking. Would you like me to book a reservation for you?
GPT-2 with k.	Yes it does. Would you like me to make reservations for you?

Figure 13: Example responses generated by our proposed methods

Multi-Action Dialog Policy Learning with Interactive Human Teaching

Megha Jhunjhunwala, Caleb Bryant, Pararth Shah

Facebook Conversational AI

Menlo Park, CA

{meghaj, calb, pararths}@fb.com

Abstract

We present a framework for improving task-oriented dialog systems through online interactive teaching with human trainers. A dialog policy trained with imitation learning on a limited corpus may not generalize well to novel dialog flows often uncovered in live interactions. This issue is magnified in multi-action dialog policies which have a more expressive action space. In our approach, a pre-trained dialog policy model interacts with human trainers, and at each turn the trainers choose the best output among N-best multi-action outputs. We present a novel multi-domain, multi-action dialog policy architecture trained on MultiWOZ, and show that small amounts of online supervision can lead to significant improvement in model performance. We also present transfer learning experiments which show that interactive learning in one domain improves policy model performance in related domains.

1 Introduction

Task-oriented dialog systems help users to complete tasks by interacting with the user through a multi-turn natural dialogue (Pietquin, 2006; Young et al., 2013). The dialog manager module plays a key role of maintaining state across the conversation and selecting actions in each turn to drive the dialog to successful completion. Within the dialog manager, the dialog policy module chooses the system’s actions in each state (Young et al., 2013), and it is typically constructed in one of the following ways: (1) handcrafted with rules defined by a conversation designer (Bordes et al., 2017), (2) trained with imitation learning on dialog samples collected from human-human interactions (Wen et al., 2017; Liu et al., 2018; Budzianowski et al., 2018), or (3) trained with reinforcement learning with a user simulator (Zhao and Eskenazi, 2016).

In practice, each approach has its unique advantages and disadvantages, making it difficult to build

an optimal dialog policy with a single approach. Systems crafted from large numbers of rules (Bohus and Rudnicky, 2009; Lison and Kennington, 2016) are time-intensive to build and often lead to rigid dialog flows. Supervised learning over human-human dialog samples is widely studied. However, human-human dialogs collected in a Wizard-of-Oz setup (Budzianowski et al., 2018; Eric et al., 2017) cannot cover all dialog states occurring in human-machine interactions, such as dialog states occurring due to system errors. Models trained on human-human data alone do not generalize well to human-machine dialogs and face compounding errors when a deviation in a single turn takes the dialog to a new state which the model might have never seen during training (Liu et al., 2018). In contrast, dialog systems trained with reinforcement learning, either with user simulators or by receiving feedback from user interactions, have shown improved robustness in diverse dialogue scenarios (Williams et al., 2017; Liu and Lane, 2017). However, the reward signal used in RL provides distant and weak supervision, resulting in large amounts of samples required for the model to learn the credit assignment between actions and outcomes (Liu et al., 2018). A number of works attempt to combine the best of both worlds through hybrid approaches (Henderson et al., 2008; Liu et al., 2018).

Most prior work on dialog policy modeling assumes only one policy action per turn (Bordes et al., 2017; Ilievski et al., 2018; Liu and Lane, 2017), which limits interaction quality and increases dialog length, leading to more errors. Generating multiple dialog acts in a single turn can increase the system’s expressive power, and this can be formulated as a multi-label classification or a sequence generation problem (Shu et al., 2019). However, having more than one act in a single turn exponentially increases the space of possible outputs. A limited corpus is unlikely to cover a large number

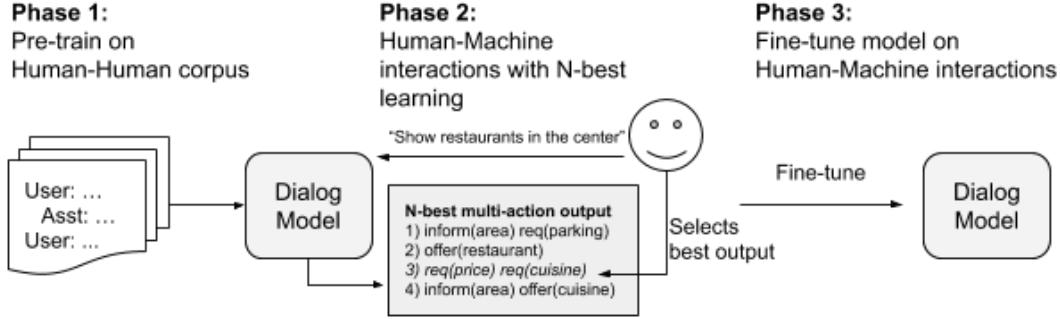


Figure 1: Policy Learning with Interactive Action Selection (PLIAS)

of combinations of output acts, and models trained with supervised learning alone will be restricted to a small subspace of the complete output action space.

In this paper, we propose ‘‘Policy Learning with Interactive Action Selection’’ (PLIAS), a generic framework for learning dialog policies which combines pre-training on human-human dialog samples and interactive learning with human-machine interactions. The interactive learning step is designed to maximize supervision quality while minimizing annotation time and cost. We employ the PLIAS framework on Dialog Action Sequence Policy (DASP), a novel multi-domain, multi-action dialog policy architecture. Experiments on Multi-WOZ (Budzianowski et al., 2018) show that PLIAS significantly improves model performance.

2 Policy Learning with Interactive Action Selection (PLIAS)

Figure 1 shows the 3-step approach of PLIAS: (1) pre-train a dialog policy model on an annotated human-human dialog corpus, (2) generate human-machine interactions where a human interacts with the model and picks the best output from N-best policy outputs, (3) fine-tune the policy model on the interactive learning dialog sessions from step 2. In this section, we describe PLIAS in context of interactively improving the DASP model.

Dialog Action Sequence Policy (DASP) model. Each task-oriented dialog is modeled as a sequence of user and system turns. Each system turn a_t is associated with a sequence of dialog acts, $a_t = (a_{t1}, a_{t2}, \dots, a_{tn})$, where each a_{ti} represents one atomic conversational action (Budzianowski et al., 2018). Some example dialog acts include *inform(hotel, name)* and *request(restaurant, price)*. A_m is the set of all such dialog act sequences up

to a fixed length m . We model DASP as a function $\pi_\theta : U \times B \times K \mapsto A_m$, where U is the set of possible input utterances, B is the set of possible belief states, K is the set of possible knowledge base results for a dialog turn, and θ is a set of parameters learned by our policy model.

Following (Budzianowski et al., 2018), DASP is modeled as a neural network that receives both sparse (text) and dense (belief state and KB result) features. The user utterance is ‘‘delexicalized’’, to replace slot value mentions with special tokens, and fed into an LSTM encoder (Wen et al., 2015). The belief state is encoded as a one-hot vector for each slot, denoting whether a slot is empty, filled, or ‘‘dont care’’. The KB is queried with the updated belief state to obtain a one-hot KB vector for each domain indicating the number of entities compatible with the current belief state. The utterance encoder’s final hidden cell and output vectors are concatenated together with the belief state and KB vectors for the current dialog turn, and passed to an LSTM decoder which produce a sequence of dialog act output tokens, with attention over the input tokens. While the dialog model in (Budzianowski et al., 2018) directly outputs the system utterance, DASP outputs semantic dialog action tokens which are fed to a separate NLG module to generate the final response. We define a flat multi-domain multi-action sequence encoding as follows:

$$a_{ti} = \{\text{Domain}, \text{Act}, \text{Slot}_1, \dots, \text{Slot}_p\} \quad (1)$$

$$a_t = \{a_{t1}, a_{t2}, \dots, a_{tn}\} (n \leq m) \quad (2)$$

For example, the dialog act sequence (*inform(hotel, address)*, *inform(hotel, price)*, *request(hotel, parking)*) is encoded as {*hotel, inform, address, price, request, parking*}.

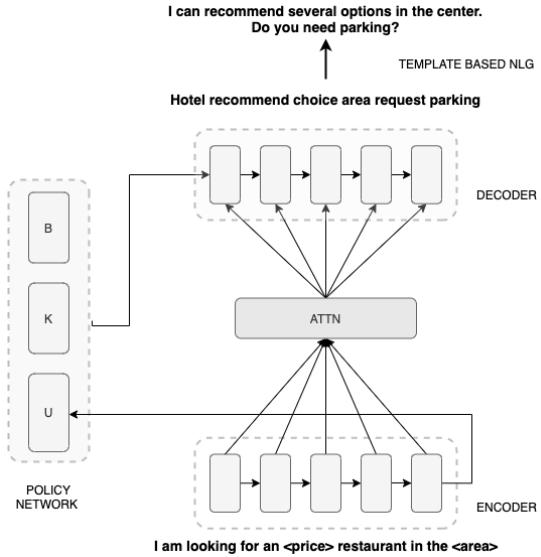


Figure 2: Dialog Action Sequence Policy (DASP) model.

increase training efficiency, we normalize the target dialog act sequences for each turn in training data by recursive alphabetical sorting: first sort each dialog act group by domain, then within each group sort by dialog act type, then sort the slot names within each dialog act.

N-best candidate action sequences. We use beam search (Graves, 2012) to generate a ranked list of predicted action sequences from the DASP model at each turn. We filter out sequences with invalid actions (e.g. informing a slot that does not exist in the current belief state), and pick the top five candidate action sequences. These candidates are fed to an NLG module to generate natural responses, which are shown to a human for interactive action selection.

Interactive action selection. The goal of the interactive learning phase is to collect high quality supervision signal with minimal annotation cost. This is achieved by designing a user interface where a human trainer interacts with the dialog system and corrects the system’s outputs (Fig 3). To reduce annotation overhead, the interface presents the top-5 candidate responses from the model, and the trainer picks the best one to continue the dialog. The trainer also gives a rating (1 to 5) for the chosen response, which aids in filtering out turns where none of the candidate responses were acceptable. The trainers are instructed to end the dialog when the task is complete or if the model returns the same incorrect response twice in a row.

Fine-tuning step. The corrected dialog samples

Table 1: Task Success Rate

	GTST			TST		
	Rest	Hotel	Multi	Rest	Hotel	Multi
PT	0.45	0.35	0.34	0.44	0.65	0.66
BT	0.53	0.53	0.66	0.45	0.65	0.71
FT	0.56	0.69	0.85	0.64	0.70	0.77
Human	0.65	0.68	0.90	0.41	0.57	0.74

Table 2: Avg. turn rating (1 to 5)

	GTST			TST		
	Rest	Hotel	Multi	Rest	Hotel	Multi
PT	4.03	2.76	2.81	2.92	3.32	2.85
BT	3.92	4.00	3.66	2.77	3.23	2.29
FT	4.09	4.28	4.22	3.52	3.90	3.32
Human	4.24	4.12	4.20	3.62	3.71	3.23

obtained from the interactive learning phase are filtered to keep only the turns with user rating greater than 3. The DASP model pre-trained on the original human-human corpus (DASP-PT) is fine-tuned (Yosinski et al., 2014) using supervised learning on the new samples to obtain DASP-FT. Fine-tuning was performed by pre-loading the original weights of DASP-PT model and using a learning rate 10 times smaller than the one used for training the pre-trained model. For comparison, we also train a model bootstrapped only on the interactive learning samples, called DASP-BT. The DASP-BT model is initialized with random weights and training with the same learning rate as the pre-trained model.

3 Experiments

We present experiments on MultiWOZ (Budzianowski et al., 2018), restricted to dialogs in two domains, *restaurant* and *hotel*, including dialogs that span both of them, which we refer to as *multi*. For all the experiments, we use a rule-based belief tracker to track the slot updates across each turn, and a template-based NLG module (Shah et al., 2018). The DASP model requires a NLU slot tagger to delexicalize the user inputs. To isolate the impact of PLIAS from the effectiveness of the slot tagger, we devised two modes in our interactive learning step: trained-slot-tagger (TST) and ground-truth-slot-tagger (GTST). In TST, we trained a seq2seq slot tagger (Hakkani-Tür et al., 2016) on user utterances in MultiWOZ corpus, and integrated it in the action selection step to tag the human trainer’s input utterances. In GTST, we switched the trainer’s input from free-form text to a search over templated user utterances extracted from MultiWOZ (Fig 3), which skips the need for slot tagging and enables us to collect interactive

Table 3: Average per-annotator score increase from interactive learning

	Rest	Hotel	Multi
PT	4.06	3.21	3.05
BT	+0.02 (0%)	+1.29 (40.2%)	+0.83 (27%)
FT	+0.18 (4.4%)	+1.79 (55.8%)	+0.92 (30.%)
Human	+0.33 (8.1%)	+1.18 (36.8%)	+1.40 (46%)

learning samples with gold NLU labels.

We pre-trained a single multi-domain model on the entire train split of MultiWOZ (4000 dialogs), then ran interactive action selection of 300 dialog sessions for each pair of restaurant, hotel, multi and TST, GTST. To measure the effectiveness of PLIAS, we evaluate all three models DASP-PT, DASP-BT and DASP-FT. In the *interactive evaluation* mode, action selection is disabled and the system responds with the top action sequence prediction. The trainer gives a 1-5 rating for each turn based on the quality of the system’s chosen output. We collected 100 sessions of interactive evaluation for each combination of DASP model, domain, and slot-tagger mode. We report two scores for each experiment: (1) Task Success Rate (TSR), which aggregates the overall rate of task completion of the model in human-machine interactions, and (2) Avg. turn-wise human rating, which aggregates the subjective per-turn feedback score given by the human trainers.

We also present a transfer learning experiment to evaluate the effectiveness of interactive policy learning to generalize knowledge to related domains. In this experiment, we trained new DASP-FT and DASP-BT models (in GTST mode) on the interactive learning samples restricted to restaurant domain, and performed interactive evaluations of these models on tasks from all three domains - restaurant, hotel and multi.

3.1 Results

We observe a clear trend of improved performance from pre-trained (PT) to bootstrapped (BT) to fine-tuned (FT), in both TSR (Table 1) and avg. human feedback scores (Table 2). For comparison, the tables also show the “Human” TSR and avg. turn rating, from the interactive *learning* sessions, where the human trainer is picking the best action sequence from top-5 candidates. The fine-tuned (FT) model closes the gap with Human performance, and also outperforms the bootstrapped (BT) model, which shows that pre-training with the larger dataset helps to improve the overall policy

Table 4: Transfer learning results

	TSR			Avg. turn rating		
	Rest	Hotel	Multi	Rest	Hotel	Multi
PT	0.45	0.35	0.34	4.03	2.76	2.81
BT	0.47	0.10	0.26	4.05	1.60	2.07
FT	0.60	0.79	0.77	4.12	4.21	3.75
Human	0.65	0.68	0.90	4.24	4.12	4.20

performance.

In order to normalize the scores across trainers, Table 3 presents the human feedback scores aggregated on a per-trainers basis. Each human trainer performed multiple dialog sessions in each evaluation job, so we first compute the average score by each trainer, then compute the delta in the score between pre-trained (PT) and all other models for that trainer, and then take a global average of the deltas across all trainers. We see that on average the same human trainer gives a higher score to the fine-tuned model compared to the original pre-trained model.

Table 4 presents the TSR and Avg Turn Rating scores for the transfer learning experiment. Since the BT model was bootstrapped only using the restaurant domain data, the lower performance on hotel and multi is expected. However, the FT model outperforms the PT model even in the hotel and multi tasks. This shows that fine-tuning a multi-domain architecture on a single domain can boost performance in other related domains.

3.2 Analysis

We present dialog samples between a human trainer and DASP models in Table 5. Both dialogs begin in a similar manner with the user asking for a guest-house with free parking and the system responding with several choices and asking more follow up questions to narrow the search. When the system cannot find any matches for a 2-star guest-house, the pre-trained model (DASP-PT) gets stuck in a loop asking for the same location slot. On the other hand, the fine-tuned model (DASP-FT) is able to recommend an alternative which leads to task completion. This example shows how fine-tuning on dialogs from the interactive action selection phase enables the policy model to properly handle a wider range of dialog states.

4 Conclusion and Future Work

In this paper, we presented PLIAS, a generic framework for policy learning through interactive action selection. We described the framework in context of a novel multi-domain multi-action policy model,

Table 5: Sample dialogs between human and DASP models. **Note:** Each system turn is annotated with the turn-wise human rating (from 1 to 5) for that turn, provided in brackets after the turn.

Task (common to both dialogs): In this task, the user is instructed to book a guesthouse with free parking and rating of 2 stars. If no such guesthouse is found, they are asked to try to find a 4-star one.	
DASP-PT model in GTST mode	DASP-FT model in GTST mode
USER I'm looking for a guesthouse with free parking. SYSTEM The number of choices is 24 with no parking and hotel type as guesthouse. Please tell me the location. (4) USER Yes, it should have 2 stars. SYSTEM I'm sorry, I could not find any matches for your query. Would there be anything else? (5) USER I'm looking for a 4 star hotel. SYSTEM Please tell me the location. (5) USER I don't really care about the area. SYSTEM Please tell me the location. (1) USER I want to find a guesthouse with free parking included. Can you help? SYSTEM The number of choices is 16 with free parking and hotel type is guesthouse. Please tell me the location. (5) USER I don't care, but it should be a 4 star establishment. SYSTEM Please tell me the location. (1) USER i don't really care about the area. SYSTEM Please tell me the location. (1) USER Bye SYSTEM Good bye, have a nice day! (5)	USER I am looking for a place to stay. The hotel should be in the type of guesthouse and should include free parking. SYSTEM The number of choices is 21 with hotel type as guesthouse. Please tell me the location and the price. (5) USER The hotel should have a star of 2. SYSTEM The number of choices is 0 with rating as 2. Please tell me the location. (5) USER How about one that has a star of 4? SYSTEM How about acorn guest house? The location is north and the rating is 4. Would there be anything else? (5) USER i want to book it for 8 people and 5 nights starting from Saturday. SYSTEM Your booking is done for Saturday at acorn guest house. The number of people in the booking is 8 and the stay is 5 nights. This is your reference number: ref1860. Would there be anything else? (5) USER No that's all. Thank you. SYSTEM You are welcome. (5)

DASP, which generates a dialog action sequence output that encodes multiple system actions across domains. Experiments with models pre-trained on MultiWOZ and further optimized through interactive learning convey that the interactive action selection step significantly improves the performance of policy models. Directions of future work include: (1) evaluating the PLIAS framework with other multi-action policy architectures (e.g. (Shu et al., 2019)), (2) using beam search variants that promote diversity in candidate sequences (e.g. (Li et al., 2016)), (3) studying the correlation between model performance and an increasing/decreasing number of human-machine interactive sessions.

References

Dan Bohus and Alexander Rudnicky. 2009. The ravenclaw dialog management framework: Archi-

ture and systems. *Computer Speech Language*, 23(3):332–361.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 2018. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.

- Alex Graves. 2012. Sequence transduction with recurrent neural networks. *ArXiv*, abs/1211.3711.
- Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, pages 715–719.
- James Henderson, Oliver Lemon, and Kallirroi Georgila. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics*, 34(4):487–511.
- Vladimir Ilievski, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. 2018. Goal-oriented chatbot dialog management bootstrapping with transfer learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI’18*, page 4115–4121. AAAI Press.
- Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.
- Pierre Lison and Casey Kennington. 2016. OpenDial: A toolkit for developing spoken dialogue systems with probabilistic rules. In *Proceedings of ACL-2016 System Demonstrations*, pages 67–72, Berlin, Germany. Association for Computational Linguistics.
- Bing Liu and Ian Lane. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE.
- Bing Liu, Gokhan Tür, Dilek Hakkani-Tür, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2060–2069, New Orleans, Louisiana. Association for Computational Linguistics.
- O. Pietquin. 2006. Consistent goal-directed user model for realistic man-machine task-oriented spoken dialogue simulation. In *2006 IEEE International Conference on Multimedia and Expo*, pages 425–428.
- Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51.
- Lei Shu, Hu Xu, Bing Liu, and Piero Molino. 2019. Modeling multi-action policy for task-oriented dialogues. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1304–1310, Hong Kong, China. Association for Computational Linguistics.
- Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.
- Jason D Williams, Kavosh Asadi Atui, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 665–677.
- Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.
- Steve Young, Milica Gasic, Blaise Thomson, and Jason Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101:1160–1179.
- Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–10, Los Angeles. Association for Computational Linguistics.

A Action Selection Interface

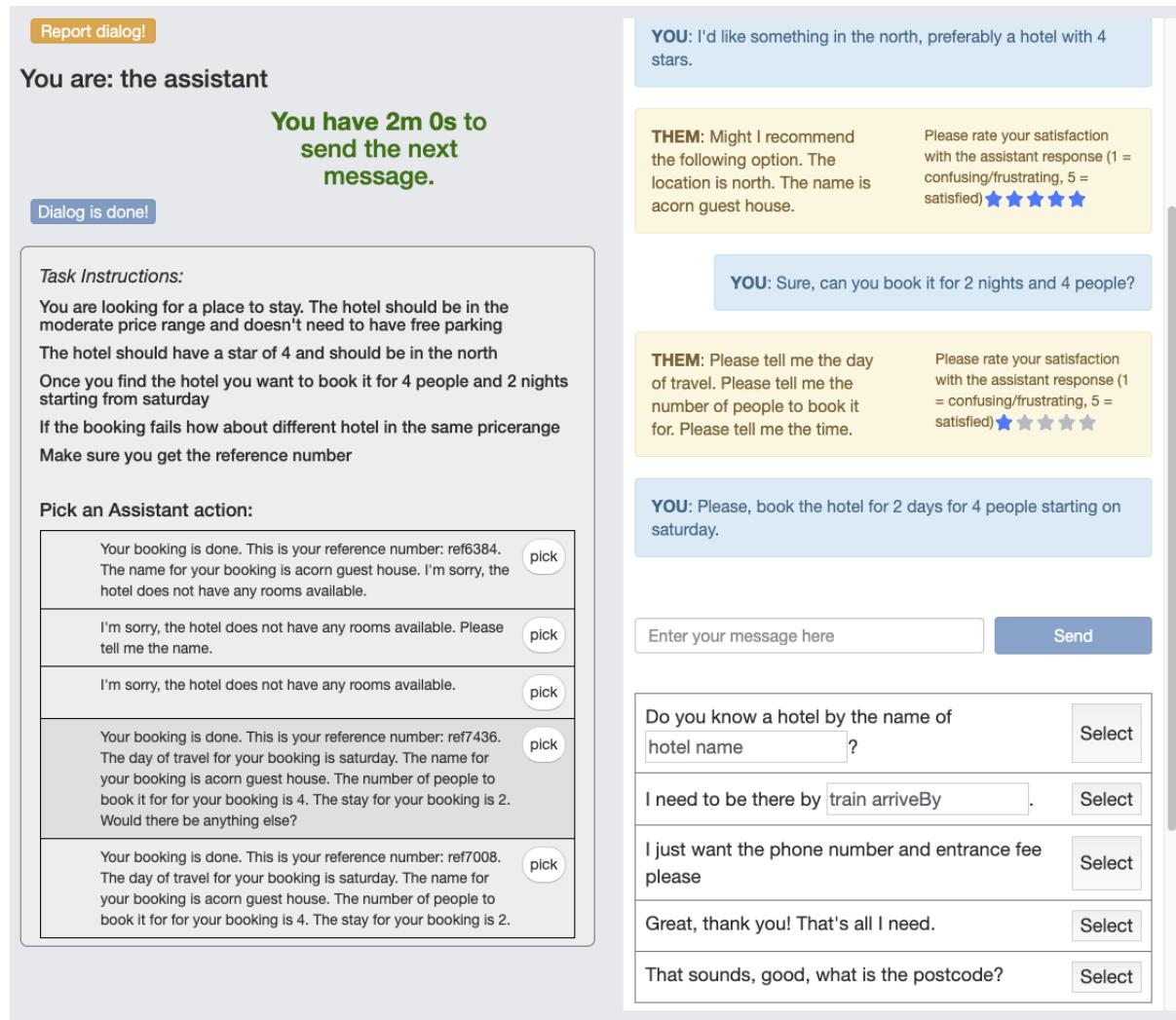


Figure 3: Interactive action selection interface. A demo video of the interface is submitted in the supplementary materials.

B Model training details

- Learning rate: 0.005
- Hidden layer size: 150 (encoder, decoder, policy network)
- Embedding size for user utterance : 50
- Max length for user utterance: 50 words
- Max length of dialog act sequence output: 50 tokens
- Teacher ratio of 1
- Beam search width = 3

Is Your Goal-Oriented Dialog Model Performing Really Well? Empirical Analysis of System-wise Evaluation

Ryuichi Takanobu¹, Qi Zhu¹, Jinchao Li², Baolin Peng², Jianfeng Gao², Minlie Huang^{1*}

¹DCST, Institute for AI, BNRist, Tsinghua University, Beijing, China

²Microsoft Research, Redmond, USA

¹{gxly19, zhu-q18}@mails.tsinghua.edu.cn aihuang@tsinghua.edu.cn

²{jincli, bapeng, jfgao}@microsoft.com

Abstract

There is a growing interest in developing goal-oriented dialog systems which serve users in accomplishing complex tasks through multi-turn conversations. Although many methods are devised to evaluate and improve the performance of individual dialog components, there is a lack of comprehensive empirical study on how different components contribute to the overall performance of a dialog system. In this paper, we perform a system-wise evaluation and present an empirical analysis on different types of dialog systems which are composed of different modules in different settings. Our results show that (1) a pipeline dialog system trained using fine-grained supervision signals at different component levels often obtains better performance than the systems that use joint or end-to-end models trained on coarse-grained labels, (2) component-wise, single-turn evaluation results are not always consistent with the overall performance of a dialog system, and (3) despite the discrepancy between simulators and human users, simulated evaluation is still a valid alternative to the costly human evaluation especially in the early stage of development.

1 Introduction

Many approaches and architectures have been proposed to develop goal-oriented dialog systems to help users accomplish various tasks (Gao et al., 2019a; Zhang et al., 2020b). Unlike open-domain dialog systems, which are designed to mimic human conversations rather than complete specific tasks and are often implemented as end-to-end systems, a goal-oriented dialog system has access to an external database on which to inquire about information to accomplish tasks for users. Goal-oriented dialog systems can be grouped into three classes based on their architectures, as illustrated in Fig. 1.

*Corresponding author

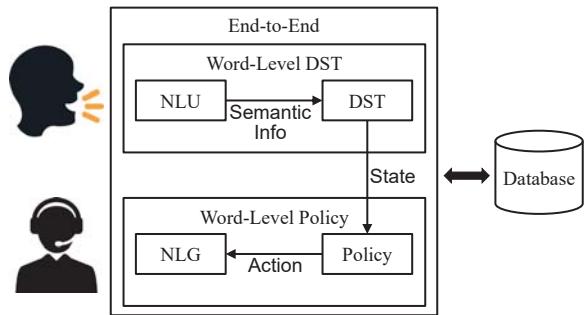


Figure 1: Different architectures of goal-oriented dialog systems. It can be constructed as a pipeline or end-to-end system with different granularity.

The first class is the pipeline (or modular) systems which typically consist of the four components: *Natural Language Understanding* (NLU) (Goo et al., 2018; Pentyala et al., 2019), *Dialog State Tracker* (DST) (Xie et al., 2015; Lee and Stent, 2016), *Dialog Policy* (Peng et al., 2017; Takanobu et al., 2019), and *Natural Language Generation* (NLG) (Wen et al., 2015; Balakrishnan et al., 2019). The second class is the end-to-end (or unitary) systems (Williams et al., 2017; Dhingra et al., 2017; Liu et al., 2018; Lei et al., 2018; Qin et al., 2019; Mehri et al., 2019), which use a machine-learned neural model to generate a system response directly from a dialog history. The third one lies in between the above two types, where some systems use joint models that combine some (but not all) of the four dialog components. For example, a joint word-level DST model combines NLU and DST (Zhong et al., 2018; Wu et al., 2019; Gao et al., 2019b), and a joint word-level policy model combines dialog policy and NLG (Chen et al., 2019; Zhao et al., 2019; Budzianowski and Vulić, 2019).

It is particularly challenging to properly evaluate and compare the overall performance of goal-oriented dialog systems due to the wide variety of system configurations and evaluation settings. Nu-

merous approaches have been proposed to tackle different components in pipeline systems, whereas these modules are merely evaluated separately. Most studies only compare the proposed models with baselines of the same module, assuming that a set of good modules can always be assembled to build a good dialog system, but rarely evaluate the overall performance of a dialog system from the system perspective. A dialog system can be constructed via different combinations of these modules, but few studies investigated the overall performance of different combinations (Kim et al., 2019; Li et al., 2020). Although end-to-end systems are evaluated in a system-wise manner, none of such systems is compared with its pipeline counterpart. Furthermore, unlike the component-wise assessment, system-wise evaluation requires simulated users or human users to interact with the system to be evaluated via multi-turn conversations to complete tasks.

To this end, we conduct both simulated and human evaluations on dialog systems with a wide variety of configurations and settings using a standardized dialog system platform, Convlab (Lee et al., 2019b), on the MultiWOZ corpus (Budzianowski et al., 2018). Our work attempts to shed light on evaluating and comparing goal-oriented dialog systems by conducting a system-wise evaluation and a detailed empirical analysis. Specifically, we strive to answer the following research questions: (RQ1) Which configurations lead to better goal-oriented dialog systems? (§3.1); (RQ2) Whether the component-wise, single-turn metrics are consistent with system-wise, multi-turn metrics for evaluation? (§3.2); (RQ3) How does the performance vary when a system is evaluated using tasks of different complexities, e.g., from single-domain to multi-domain tasks? (§3.3); (RQ4) Does simulated evaluation correlate well with human evaluation? (§3.4).

Our results show that (1) pipeline systems trained using fine-grained supervision signals at different component levels often achieve better overall performance than the joint models and end-to-end systems, (2) the results of component-wise, single-turn evaluation are not always consistent with that of system-wise, multi-turn evaluation, (3) as expected, the performance of dialog systems of all three types drops significantly with the increase of task complexity, and (4) despite the discrepancy between simulators and human users, simulated



Figure 2: Domain distribution of the user goals used in the experiments. A goal with multiple domains is counted repeatedly for each domain.

evaluation correlates moderately with human evaluation, indicating that simulated evaluation is still a valid alternative to the costly human evaluation, especially in the early stage of development.

2 Experimental Setting

2.1 Data

In order to conduct a system-wise evaluation and an in-depth empirical analysis of various dialog systems, we adopt the MultiWOZ (Budzianowski et al., 2018) corpus in this paper. It is a multi-domain, multi-intent task-oriented dialog corpus that contains 3,406 single-domain dialogs and 7,032 multi-domain dialogs, with 13.18 tokens per turn and 13.68 turns per dialog on average. The dialog states and system dialog acts are fully annotated. The corpus also provides the domain ontology that defines all the entities and attributes in the external databases. We also use the augmented annotation of user dialog acts from (Lee et al., 2019b).

2.2 User Goal

During evaluation, a dialog system interacts with a simulated or human user to accomplish a task according to a pre-defined user goal. A user goal is the description of the state that a user wants to reach in a conversation, containing *indicated constraints* (e.g., a restaurant serving Japanese food in the center of the city) and *requested information* (e.g., the address, phone number of a restaurant).

A user goal is initialized to launch the dialog session during evaluation. To ensure a fair comparison, we apply a fixed set of 1,000 user goals for both simulated and human evaluation. In the goal sampling process, we first obtain the frequency of each slot in the dataset and then sample a user goal from the slot distribution. We also apply additional rules to remove inappropriate combinations, e.g., a user cannot inform and inquire about the arrival time of a train in the same session. In the case

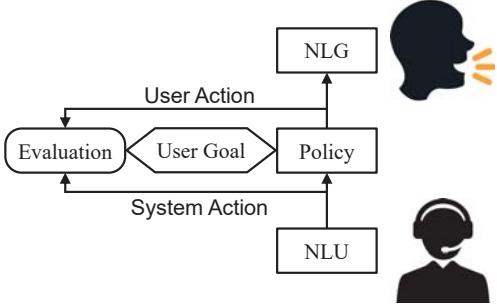


Figure 3: The framework of a user simulator and the mechanism for simulated evaluation.

where no matching database entry exists based on the sampled goal, we resample a new user goal until there is an entity in the database that satisfies the new constraints. In evaluation, the user first communicates with the system based on the initial constraints, and then can change the constraints if the system informs the user that the requested entity is not available. The detailed distribution of these goals is shown in Fig. 2. Among the 1,000 user goals, the numbers of goals involving 1/2/3 domains are 328/549/123, respectively.

2.3 Platform and Simulator

We use the open-source end-to-end dialog system platform, ConvLab (Lee et al., 2019b), as our experimental platform. ConvLab enables researchers to develop a dialog system using preferred architectures and supports system-wise simulated evaluation. It also provides an integration of crowdsourcing platforms such as Amazon Mechanical Turk for human evaluation.

To automatically evaluate a multi-turn dialog system, Convlab implements an agenda-based user simulator (Schatzmann et al., 2007). Given a user goal, the simulator’s policy uses a stack-like structure with complex hand-crafted heuristics to inform its goal and mimics complex user behaviors during a conversation. Since the system interacts with the simulator in natural language, the user simulator directly takes system utterances as input and outputs a user response. The overall architecture of user simulator is presented in Fig. 3. It consists of three modules: NLU, policy, and NLG. We use the default configuration of the simulator in Convlab: a RNN-based model MILU (**M**ulti-**I**ntent **L**anguage **U**nderstanding, extended (Hakkani-Tür et al., 2016)) for NLU, a hand-crafted policy, and a retrieval model for NLG.

2.4 Evaluation Metrics

We use the number of *dialog turns*, averaging over all dialog sessions, to measure the efficiency of accomplishing a task. A user utterance and a subsequent system utterance are regarded as one dialog turn. The system should help each user accomplish his/her goal within 20 turns, otherwise the dialog is regarded as failure. We utilize two other metrics: *inform F1* and *match rate* to estimate the task success. Both metrics are calculated based on the *dialog act* (Stolcke et al., 2000), an abstract representation that extracts the semantic information of an utterance. The dialog act from the input and output of the user simulator’s policy will be used to calculate two scores, as shown in Fig. 3. Inform F1 evaluates whether all the information *requests* are fulfilled, and match rate assesses whether the offered entity meets all the *constraints* specified in a user goal. The dialog is marked as successful if and only if both inform recall and match rate are 1.

2.5 System Configurations

To investigate how much system-wise and component-wise evaluations differ, we compare a set of dialog systems that are assembled using different state-of-the-art modules and settings in our experiments. The full list of these systems are shown in Table 1, which includes 4 pipeline systems (*SYSTEM-1~4*), 10 joint-model systems (*SYSTEM-5~13*) and 2 end-to-end systems (*SYSTEM-15~16*). Note that some systems (e.g. *SYSTEM-4*, *SYSTEM-10*) generate *delexicalized* responses where the slot values are replaced with their slot names. We convert these responses to natural language by filling the slot values based on dialog acts and/or database query results.

In what follows, we briefly introduce these modules and the corresponding models¹ used in our experiments. The component-wise evaluation results of these modules are shown in Table 2. For published works, we train all the models using the open-source code with the training, validation and test split offered in MultiWOZ, and replicate the performance reported in the original papers or on the leaderboard.

NLU A *natural language understanding* module identifies user intents and extracts associated information from users’ raw utterances. We consider

¹All state-of-the-art models mentioned in this paper are based on the open-source code that is available and executable as of February 29, 2020.

two approaches that can handle multi-intents as reference: a RNN-based model MILU which extends (Hakkani-Tür et al., 2016) and is fine-tuned on multiple domains, intents and slots; and a fine-tuned BERT model (Devlin et al., 2019). Following the joint tagging scheme (Zheng et al., 2017), the labels of intent detection and slot filling are annotated for domain classification during training. Both models use dialog history up to the last dialog turn as context. Note that there can be multiple intents or slots in one sentence, we calculate two F1 scores for intents and slots, respectively.

DST A *dialog state tracker* encodes the extracted information as a compact set of dialog state that contains a set of informative slots and their corresponding values (user constraints), and a set of requested slots². We have implemented a rule-based DST to update the slot values in the dialog state based on the output of NLU. We then compare four word-level DST: a multi-domain classifier MDBT (Ramadan et al., 2018) which enumerates all possible candidate slots and values, SUMBT (Lee et al., 2019a) that uses a BERT encoder and a slot-utterance matching architecture for classification, TRADE (Wu et al., 2019) that shares knowledge among domains to directly generate slot values, and COMER (Ren et al., 2019) which applies a hierarchical encoder-decoder model for state generation. We use two metrics for evaluation. The joint goal accuracy compares the predicted dialog states to the ground truth at each dialog turn, and the output is considered correct if and only if all the predicted values exactly match the ground truth. The slot accuracy individually compares each (domain, slot, value) triplet to its ground truth label.

Policy A *dialog policy* relies on the dialog state provided by DST to select a system action. We compare two dialog policies: a hand-crafted policy, and a reinforcement learning policy GDPL (Takanobu et al., 2019) that jointly learns a reward function. We also include in our comparison three joint models, known as word-level policies, which combine the policy and the NLG module to produce natural language responses from dialog states. They are MDRG (Wen et al., 2017) where an at-

²Dialog state can include everything a system must know in order to make a decision about what to do next, e.g., DSTC2 corpus (Henderson et al., 2014) contains *search method* representing user intents in the dialog state, but only aforementioned items are taken into account as our experiments are conducted on MultiWOZ in this paper.

tention mechanism is conditioned on the dialog states, HDSA (Chen et al., 2019) that decodes response from predicted hierarchical dialog acts, and LaRL (Zhao et al., 2019) which uses a latent action framework. We use BLEU score (Papineni et al., 2002), inform rate and task success rate as metrics for evaluation. Note that the inform rate and task success for evaluating policies are computed at the turn level, while the ones used in system-wise evaluation are computed at the dialog level.

NLG A *natural language generation* module generates a natural language response from a dialog act representation. We experiment with two models: a retrieval-based model that samples a sentence randomly from the corpus using dialog acts, and a generation-based model SCLSTM (Wen et al., 2015) which appends a sentence planning cell in RNN. To evaluate the performance of NLG, we adopt BLEU score to evaluate the quality of the generated text, and slot error rate (SER) to measure whether the generated response contains missing or redundant slot values.

E2E An *end-to-end* model takes user utterances as input and directly output system responses in natural language. We experiment with two models: TSCP (Lei et al., 2018) that uses belief spans to represent dialog states, and DAMD (Zhang et al., 2020a) that further uses action spans to represent dialog acts as additional information. For single-turn evaluation, BLEU, inform rate and success rate are provided.

3 Empirical Analysis

3.1 Performance under Different Settings (RQ1)

We compare the performance of three types of systems, pipeline, joint-model and end-to-end. Results in Table 1 show that pipeline systems often achieve better overall performance than the joint models and end-to-end systems because using fine-grained labels at the component level can help pipeline systems improve the task success rate.

NLU with DST or joint DST It is essential to predict dialog states to determine what a user has expressed and wants to inquire. The dialog state is used to query the database, predict the system dialog act, and generate a dialog response. Although many studies have focused on the word-level DST that directly predicts the state using the

ID	Configuration				Turn	Inform			Match	Succ.
	NLU	DST	Policy	NLG		Prec.	Rec.	F1		
1	BERT	rule	rule	retrieval	6.79	0.79	0.91	0.83	90.54	80.9
2	MILU	rule	rule	retrieval	7.24	0.76	0.88	0.80	87.93	77.6
3	BERT	rule	GDPL	retrieval	10.86	0.72	0.69	0.69	68.34	54.1
4	BERT	rule	rule	SCLSTM	13.38	0.64	0.58	0.58	51.41	43.0
5		MDBT	rule	retrieval	16.55	0.47	0.35	0.37	39.76	18.8
6		SUMBT	rule	retrieval	13.71	0.51	0.44	0.44	46.44	27.8
7		TRADE	rule	retrieval	9.56	0.39	0.41	0.37	38.37	22.4
8		COMER	rule	retrieval	16.79	0.30	0.28	0.28	29.06	17.3
9	BERT	rule		MDRG	17.90	0.35	0.34	0.32	29.07	19.2
10	BERT	rule		HDSA	15.91	0.47	0.62	0.50	39.21	34.3
11	BERT	rule		LaRL	13.08	0.40	0.68	0.48	68.95	47.7
12		SUMBT		HDSA	18.67	0.27	0.32	0.26	14.78	13.7
13		SUMBT		LaRL	13.92	0.36	0.64	0.44	57.63	40.4
14		TRADE		LaRL	14.44	0.35	0.57	0.40	36.07	30.8
15			TSCP		18.20	0.37	0.32	0.31	13.68	11.8
16			DAMD		11.27	0.64	0.69	0.64	59.67	48.5

Table 1: System-wise simulated evaluation with different configurations and models. We use *SYSTEM-<ID>* to represent the configuration’s abbreviation throughout the paper.

user query, we also investigate the cascaded configuration where an NLU model is followed by a rule-based DST. As shown in Table 1, the success rate has a sharp decline when using word-level DST, compared to using an NLU model followed by a rule-based DST (17.3%~27.8% in *SYSTEM-5~8* vs. 80.9% in *SYSTEM-1*). The main reason is that the dialog act predicted by NLU contains both slot-value pairs and **user intents**, whereas the dialog state predicted by the word-level DST only records the user constraints in the current turn, causing information loss for action selection (via dialog policy) as shown in Fig. 4. For example, a user may want to confirm the booking time of the restaurant, but such an intent cannot be represented in the slot values. However, we can observe that word-level DST achieves better overall performance by combining with word-level policy, e.g., 40.4% success rate in *SYSTEM-13* vs. 27.8% in *SYSTEM-6*. This is because word-level policy implicitly detects user intents by encoding the **user utterance** as additional input, as presented in Fig. 5. Nevertheless, all those joint approaches still under-perform traditional pipeline systems.

NLG from dialog act or state We compare two strategies for generating responses. One is based on an ordinary NLG module that generates a response according to dialog act predicted by dialog policy. The other uses the word-level policy to di-

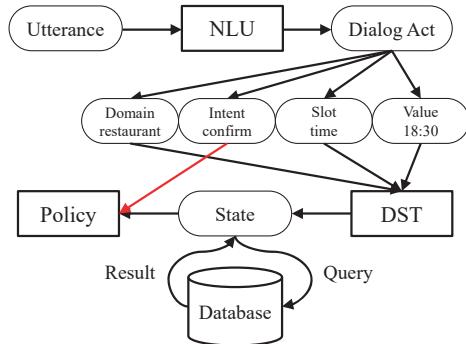


Figure 4: Illustration of NLU and DST in the dialog system. The intent information (red arrow) is missing in the dialog state on MultiWOZ if the system merges a word-level DST with a dialog policy.

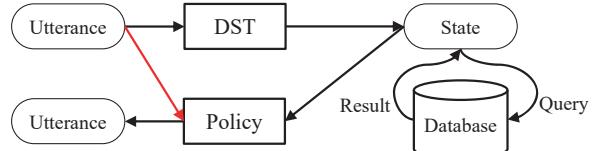


Figure 5: The common architecture of a system using word-level or end-to-end models. User utterances are encoded again (red arrow) for response generation.

rectly generates a natural language response based on dialog state and user query. As we can see in Table 1 that the performance drops substantially when we replace the retrieval NLG module with a joint model such as *MDRG* or *HDSA*. This indicates that

the dialog act has encoded sufficient semantic information so that a simple retrieval NLG module can give high-quality replies. However, the fact, that *SYSTEM-11* which uses word-level policy *LaRL* even outperforms *SYSTEM-4* which uses the NLG model *SCLSTM* in task success (47.7% vs. 43.0%), indicates that response generation can be improved by jointly training policy and NLG modules.

Database query As part of dialog management, it is crucial to identify the correct entity that satisfies the user goal. MultiWOZ contains a large number of entities across multiple domains, making it impossible to explicitly learn the representations of all the entities in the database as previous work did (Dhingra et al., 2017; Madotto et al., 2018). This requires the designed system to deal with a large-scale external database, which is closer to reality. It can be seen in Table 1 that most joint models have a lower match rate than the pipeline systems. In particular, *SYSTEM-15* rarely selects an appropriate entity during the dialog (13.68% match rate) since the proposed *belief spans* only copy the values from utterances without knowing which domain or slot type the values belong to. Due to the poor performance in dialog state prediction, it cannot consider the external database selectively, thereby failing to satisfy the user’s constraints. In comparison, *SYSTEM-16* has achieved the highest success rate (48.5%) and the second-highest match rate (59.67%) among all the systems using joint models (*SYSTEM-5~14*). This is because *DAMD* utilizes *action spans* to predict both user and system dialog acts in addition to belief spans, which behaves like a pipeline system. This indicates that an explicit dialog act supervision can improve dialog state tracking.

3.2 Component-wise vs. System-wise Evaluation (RQ2)

It is important to verify whether the component-wise evaluation is consistent with system-wise evaluation. By comparing the results in Table 1 and Table 2, we can observe that sometimes they are consistent (e.g., *BERT* > *MILU* in Table 2a, and *SYSTEM-1* > *SYSTEM-2*), but not always (e.g., *TRADE* > *SUMBT* in Table 2b, but *SYSTEM-6* > *SYSTEM-7*).

In general, a better NLU model leads to a better multi-turn conversation, and *SYSTEM-1* outperforms all other configurations in completing user goals. With respect to DST, though word-

Model	Slot	Intent	Overall
MILU	81.90	85.82	83.27
BERT	84.25	89.84	86.21

(a) NLU		
Model	Slot Acc.	Joint Acc.
MDBT [†]	89.53	15.57
SUMBT [†]	96.44	46.65
TRADE [†]	96.92	48.62
COMER	95.52	48.79

(b) Word-level DST			
Model	BLEU	Inform	Succ.
MDRG [†]	18.8	71.3	61.0
HDSA [†]	23.6	82.9	68.9
LaRL [†]	12.8	82.8	79.2

(c) Word-level Policy		
Model	BLEU	SER
Retrieval	33.1	–
SCLSTM	51.6	3.10

(d) NLG			
Model	BLEU	Inform	Succ.
TSCP	15.5	66.4	45.3
DAMD	16.6	76.3	60.4

(e) E2E		
---------	--	--

Table 2: Component-wise performance of each module. †: results from the MultiWOZ leaderboard.

level DST models directly predict dialog states without explicitly detecting user intents, most of them perform poorly in terms of joint accuracy as shown in Table 2b. This severely harms the overall performance because the downstream tasks strongly rely on the predicted dialog states. Interestingly, *TRADE* has higher accuracy than *SUMBT* on DST. But *TRADE* performs worse than *SUMBT* in system-wise evaluation (22.4% in *SYSTEM-7* vs. 27.8% in *SYSTEM-6*). The observation is similar to *COMER* vs. *TRADE*. This indicates that the results of component-wise evaluation in DST are not consistent with those of system-wise evaluation, which may be attributed to the noisy dialog state annotations (Eric et al., 2019).

As for word-level policy, *HDSA* that uses explicit dialog acts in supervision has higher BLEU than *LaRL* that uses latent dialog acts, but *LaRL* that is finetuned with reinforcement learning has much higher match rate than *HDSA* in system-wise evaluation (68.95% vs. 39.21%). Although there is small difference between *MDRG* and *HDSA* in component-wise evaluation (61.0% vs. 68.9% in

ID	Restaurant				Train				Attraction		
	Turn	Info.	Match	Succ.	Turn	Info.	Match	Succ.	Turn	Info.	Succ.
1	2.82	0.94	96.9	98	3.06	1.0	100	100	3.12	0.69	63
2	2.84	0.92	100	98	2.99	1.0	94.2	97	3.70	0.73	65
3	8.68	0.70	69.4	70	6.07	0.80	67.3	75	5.61	0.67	62
4	6.00	0.77	68.8	78	11.53	0.71	67.3	55	12.57	0.57	46
6	9.41	0.64	72.7	60	5.13	0.97	90.4	93	14.79	0.23	9
11	9.91	0.39	66.7	61	4.02	0.86	88.5	97	4.73	0.68	80
13	8.35	0.40	65.6	60	4.19	0.85	94.2	96	6.06	0.60	73
15	14.72	0.37	11.5	27	16.02	0.46	11.5	25	16.12	0.51	24
16	6.36	0.80	92.2	90	10.21	0.61	55.8	58	8.32	0.69	67

Table 3: Performance with different single domain. Most systems achieve better performance in *Restaurant* and *Train* than *Attraction*.

ID	Single				Two				Three			
	Turn	Info.	Match	Succ.	Turn	Info.	Match	Succ.	Turn	Info.	Match	Succ.
1	3.22	0.84	84.7	87	6.96	0.81	94.9	78	8.15	0.82	88.4	69
2	3.90	0.78	79.7	82	6.74	0.76	95.3	72	10.54	0.79	85.0	66
3	9.18	0.67	66.7	60	12.38	0.60	42.9	42	13.55	0.50	44.6	21
4	8.65	0.66	58.3	62	17.24	0.38	28.0	14	18.03	0.46	24.4	13
6	10.35	0.44	60.4	41	14.74	0.44	50.9	17	15.97	0.25	20.9	0
11	8.79	0.45	72.2	55	13.37	0.52	74.0	59	19.30	0.39	50.4	0
13	8.48	0.45	62.5	61	14.08	0.45	61.0	47	18.95	0.36	40.7	0
15	15.09	0.33	10.0	26	19.10	0.25	17.8	8	20.00	0.19	0.0	1
16	8.89	0.66	68.1	65	13.48	0.52	57.1	34	18.59	0.58	45.5	12

Table 4: Performance with different number of domains. All systems have performance drop as the number of domains increases.

Table 2c), the gap is increased (19.2% in *SYSTEM-9* vs. 34.3% in *SYSTEM-10*) in system-wise evaluation. In addition, even *SCLSTM* achieves a higher BLEU score than the retrieval-based model (51.6% vs. 33.1% in Table 2d), it only obtains a lower success rate (43.0% in *SYSTEM-4* vs. 80.9% in *SYSTEM-1*) when assembled with other modules. These results show again the discrepancy between component-wise and system-wise evaluation. The superiority of the systems using retrieval models may imply that lower SER in NLG is more critical than higher BLEU in goal-oriented dialog systems.

Error in multi-turn interactions Most existing work only evaluates the model with single-turn interactions. For instance, *inform rate* and *task success* at each dialog turn are computed given the current user utterance, dialog state and database query results for context-to-context generation (Wen et al., 2017; Budzianowski and Vulić, 2019). A strong assumption is that the model would be fed with the ground truth from the upstream modules or the last dialog turn. However, this assumption does not hold since a goal-oriented dialog consists of a sequence of associated inquiries and responses between the system and its user, and the system may produce erroneous output at any time. The errors may propagate to the downstream mod-

ules and affect the following turns. For instance, end-to-end models get worse success rate in multi-turn interactions than in single-turn evaluation in Table 2e. A sample dialog from *SYSTEM-1* and *SYSTEM-6* is provided in Table 6. *SYSTEM-6* does not extract the *pricerange* slot (highlighted in red color) correctly. The incorrect dialog state further harms the performance of dialog policy, and the conversation gets stuck where the user (simulator) is always asking for the postcode, thereby failing to complete the task.

To summarize, the component-wise, single-turn evaluation results do not reflect the real performance of the system well, and it is essential to evaluate a dialog system in an end-to-end, interactive setting.

3.3 Performance of Task with Different Complexities (RQ3)

With the increasing demands to address various situations in multi-domain dialog, we choose 9 representative systems across different configurations and approaches to further investigate how their performance varies with the complexities of the tasks. 100 user goals are randomly sampled under each domain setting. Results in Table 3 and 4 show that the overall performance of all systems varies with different task domains and drops significantly

with the increase of task complexity, while pipeline systems are relatively robust to task complexity.

Performance with different single domains
Table 3 shows the performance with respect to different single domains. *Restaurant* is a common domain where users inquiry some information about a restaurant and make reservations. *Train* has more entities and its domain constraints can be more complex, e.g., the preferred train should *arrive before* 5 p.m. *Attraction* is an easier one where users do not make reservations. There are 7/6/3 informative slots that need to be tracked in *Restaurant*/*Train*/*Attraction* respectively. Surprisingly, most systems perform better in *Restaurant* or *Train* than *Attraction*. This may result from the noise database in *Attraction* where *pricerange* information is missing sometimes, and from the uneven data distribution where *Restaurant* and *Train* appear more frequently in the training set. In general, pipeline systems perform more stably across multiple domains than joint models and end-to-end systems.

Performance with different number of domains
Table 4 demonstrates how the performance varies with the number of domains in a task. We can observe that most systems fall short to deal with multi-domain tasks. Though some systems such as *SYSTEM-13* and *SYSTEM-16* can achieve a relatively high inform F1 or match rate for a single domain, the overall success rate drops substantially on two-domain tasks, and most systems fail to complete three-domain tasks. The number of dialog turns also increases remarkably when the number of domains increases. Among all these configurations, only the pipeline systems *SYSTEM-2* and *SYSTEM-1* can keep a high success rate when there are three domains in a task. These results show that current dialog systems are still insufficient to deal with complex tasks, and that pipeline systems outperform joint models and end-to-end systems.

3.4 Simulated vs. Human Evaluation (RQ4)

Since the ultimate goal of a task-oriented dialog system is to help users accomplish real-world tasks, it is essential to justify the correlation between simulated and human evaluation. For human evaluation, 100 Amazon Mechanical Turk workers are hired to interact with each system and then give their judgement on task success. The ability of Language Understanding (LU) and Response Appropriateness (RA) of the systems are assessed at

ID	Turn	LU	RA	Succ.	Corr.
1	18.58	3.62	3.69	62	0.57
6	20.63	2.85	2.91	27	0.72
11	19.98	2.36	2.41	23	0.53
13	19.26	2.17	2.49	14	0.46
16	16.33	2.61	2.65	23	0.55

Table 5: System-wise evaluation with human users. Correlation coefficient between simulated and human evaluation is presented in the last column.

the same time, and each worker gives a score on these two metrics with a five-point scale. We compare 5 systems that achieve the best performance in the simulated evaluation under different settings.

Table 5 shows the human evaluation results of 5 dialog systems. Comparing with the simulated evaluation in Table 1, we can see that Pearson’s correlation coefficient lies around 0.5 to 0.6 for most systems, indicating that simulated evaluation correlates moderately well with human evaluation. Similar to simulated evaluation, the pipeline system *SYSTEM-1* obtains the highest task success rate in human evaluation. A sample human-machine dialog from *SYSTEM-1* and *SYSTEM-6* is provided in Table 7. The result is similar to the simulated session in Table 6 but *SYSTEM-6* fails to respond with the *phone* number in Table 7 instead (highlighted in red color). All these imply the reliability of the simulated evaluation in goal-oriented dialog systems, showing that simulated evaluation can be a valid alternative to the costly human evaluation for system developers.

However, compared to simulated evaluation, we can observe that humans converse more naturally than the simulator, e.g., the user confirms with *SYSTEM-1* whether it has booked 7 seats in Table 7, and most systems have worse performance in human evaluation. This indicates that there is still a gap between simulated and human evaluation. This is due to the discrepancy between the corpus and human conversations. The dataset only contains limited human dialog data, on which the user simulator is built. Both the system and the simulator are hence limited by the training corpus. As a result, the task success rate of most systems decreases significantly in human evaluation, e.g., from 40.4% to 14% in *SYSTEM-13*. This indicates that existing dialog systems are vulnerable to the variation of human language (e.g., the sentence highlighted in brown in Table 7), which demonstrates a lack of ro-

bustness in dealing with real human conversations.

4 Related Work

Developers have been facing many problems when evaluating a goal-oriented dialog system. A range of well-defined automatic metrics have been designed for different components in the system, e.g., joint goal accuracy in DST and task success rate in policy optimization introduced in Table 2b and 2c. A broadly accepted evaluation scheme for the goal-oriented dialog was first proposed by PARADISE (Walker et al., 1997). It estimates the user satisfaction by measuring two types of aspects, namely *dialog cost* and *task success*. Paek (2001) suggests that a useful dialog metric should provide an estimate of how well the goal is met and allow for a comparative judgement of different systems. Though a model can be optimized against these metrics via supervised learning, each component is trained or evaluated separately, thus difficult to reflect real user satisfaction.

As human evaluation by asking crowd-sourcing workers to interact with a dialog system is much expensive (Ultez et al., 2013; Su et al., 2016) and prone to be affected by subjective factors (Higashinaka et al., 2010; Schmitt and Ultes, 2015), researchers have tried to realize automatic evaluation of dialog systems. Simulated evaluation (Araki and Doshita, 1996; Eckert et al., 1997) is widely used in recent works (Williams et al., 2017; Peng et al., 2017; Takanobu et al., 2019, 2020) and platforms (Ultez et al., 2017; Lee et al., 2019b; Papangelis et al., 2020; Zhu et al., 2020), where the system interacts with a user simulator which mimics human behaviors. Such evaluation can be conducted at the dialog act or natural language level. The advantages of using simulated evaluation are that it can support multi-turn language interaction in a full end-to-end fashion and generate dialogs unseen in the original corpus.

5 Conclusion and Discussion

In this paper, we have presented the system-wise evaluation result and empirical analysis to estimate the practicality of goal-oriented dialog systems with a number of configurations and approaches. Though our experiments are only conducted on MultiWOZ, we believe that such results can be generalized to all goal-oriented scenarios in dialog systems. We have the following observations:

1) We find that rule-based pipeline systems generally outperform state-of-the-art joint systems and end-to-end systems, in terms of both overall performance and robustness to task complexity. The main reason is that fine-grained supervision on dialog acts would remarkably help the system plan and make decisions, because the system should predict the user intent and take proper actions during the conversation. This supports that good pragmatic parsing (e.g. dialog acts) is essential to build a dialog system.

2) Results show that component-wise, single-turn evaluation results are not always consistent with the overall performance of dialog systems. In order to accurately assess the effectiveness of each module, system-wise, multi-turn evaluation should be used from the practical perspective. We advocate assembling the proposed model of a specific module into a complete system, and evaluating the system with simulated or human users via a standardized dialog platform, such as Rasa (Bocklisch et al., 2017) or ConvLab. Undoubtedly, this will realize a full assessment of the module’s contribution to the overall performance, and facilitate fair comparison with other approaches.

3) Simulated evaluation can have a good assessment of goal-oriented dialog systems and show a moderate correlation with human evaluation, but it remarkably overestimates the system performance in human interactions. Thus, there is a need to devise better user simulators that resemble humans more closely. A simulator should be able to generate a natural and diverse response, and may change goals in complex dialog, etc. In addition, the simulator itself may make mistakes which derive the wrong estimation of the performance. However even with human evaluation a dialog system needs to deal with more complicated and uncertain situations. Therefore, it is vital to enhance the robustness of the dialog systems. Despite the discrepancy between simulators and human users, simulated evaluation is still a valid alternative to the costly human evaluation especially in the early stage of development.

Acknowledgements

This work was jointly supported by the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096), and the National Key R&D Program of China (Grant No. 2018YFC0830200). We would like to thank anonym-

mous reviewers for their valuable suggestions, and Sungjin Lee for helpful discussions.

References

- Masahiro Araki and Shuji Doshita. 1996. Automatic evaluation environment for spoken dialogue systems. In *Workshop on Dialogue Processing in Spoken Language Systems*, pages 183–194. Springer.
- Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural nlg from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844.
- Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.
- Paweł Budzianowski and Ivan Vulić. 2019. Hello, its gpt-2-how can i help you? towards the use of pre-trained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.
- Wenhui Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmad, and Li Deng. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–495.
- Wieland Eckert, Esther Levin, and Roberto Pieraccini. 1997. User modeling for spoken dialogue system evaluation. In *Proceedings of the 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 80–87. IEEE.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.
- Jianfeng Gao, Michel Galley, and Lihong Li. 2019a. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2–3):127–298.
- Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019b. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGDIAL Meeting on Discourse and Dialogue*, pages 264–273.
- Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.
- Dilek Hakkani-Tür, Gökhan Tür, Aslı Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of the 17th Annual Conference of the International Speech Communication Association*, pages 715–719.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.
- Ryuichiro Higashinaka, Yasuhiro Minami, Kohji Dohsaka, and Toyomi Meguro. 2010. Issues in predicting user satisfaction transitions in dialogues: individual differences, evaluation criteria, and prediction models. In *Proceedings of the Second international conference on Spoken dialogue systems for ambient environments*, pages 48–60.
- Seokhwan Kim, Michel Galley, R. Chulaka Gu-nasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, Minlie Huang, Luis Lastras, Jonathan K. Kummerfeld, Walter S. Lasecki, Chiori Hori, Anoop Cherian, Tim K. Marks, Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, and Raghav Gupta. 2019. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*.
- Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019a. Sumbt: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483.

- Sungjin Lee and Amanda Stent. 2016. Task lineages: Dialog state tracking for flexible interaction. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 11–21.
- Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Zheng Zhang, Yaoqin Zhang, Xiang Li, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, and Jianfeng Gao. 2019b. Convlab: Multi-domain end-to-end dialog system platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 64–69.
- Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.
- Jinchao Li, Baolin Peng, Sungjin Lee, Jianfeng Gao, Ryuichi Takanobu, Qi Zhu, Minlie Huang, Hannes Schulz, Adam Atkinson, and Mahmoud Adada. 2020. Results of the multi-domain task-completion dialog challenge. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, Eighth Dialog System Technology Challenge Workshop*.
- Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2060–2069.
- Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478.
- Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 165–177.
- Tim Paek. 2001. Empirical methods for evaluating dialog systems. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.
- Alexandros Papangelis, Mahdi Namazifar, Chandra Khatri, Yi-Chia Wang, Piero Molino, and Gokhan Tur. 2020. Plato dialogue system: A flexible conversational ai research platform. *arXiv preprint arXiv:2001.06463*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.
- Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2240.
- Shiva Pentyala, Mengwen Liu, and Markus Dreyer. 2019. Multi-task networks with universe, group, and task feature learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 820–830.
- Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, Yangming Li, and Ting Liu. 2019. Entity-consistent end-to-end task-oriented dialogue system with kb retriever. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 133–142.
- Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437.
- Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1876–1885.
- Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152.
- Alexander Schmitt and Stefan Ultes. 2015. Interaction quality: assessing the quality of ongoing spoken dialog interaction by experts and how it relates to user satisfaction. *Speech Communication*, 74:12–36.
- Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.
- Pei-Hao Su, Milica Gasic, Nikola Mrkšić, Lina M Rojas Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2431–2441.

- Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 100–110.
- Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, et al. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pages 73–78.
- Stefan Ultes, Alexander Schmitt, and Wolfgang Minker. 2013. On quality ratings for spoken dialogue systems—experts vs. users. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 569–578.
- Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella. 1997. Paradise: a framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 271–280.
- Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449.
- Jason D Williams, Kavosh Asadi Atui, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 665–677.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.
- Qizhe Xie, Kai Sun, Su Zhu, Lu Chen, and Kai Yu. 2015. Recurrent polynomial network for dialogue state tracking with mismatched semantic parsers. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 295–304.
- Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020a. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.
- Zheng Zhang, Ryuichi Takanobu, Minlie Huang, and Xiaoyan Zhu. 2020b. Recent advances and challenges in task-oriented dialog system. *arXiv preprint arXiv:2003.07490*.
- Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1208–1218.
- Suncong Zheng, Feng Wang, Hongyun Bao, Yueming Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236.
- Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467.
- Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Hotel			Restaurant	
Info	Reqt	Book	Info	Book
parking=yes pricerange=moderate	postcode address	stay=2 day=sunday	food=british pricerange=moderate area=centre	day=sunday time=16:30 people=8
SYSTEM-1			SYSTEM-6	
U: I would prefer it in the centre area . Actually I need a moderately priced restaurant . Are there any fitting that description ? This restaurant should serve british food too . D: <i>restaurant-food:british, restaurant-pricerange:moderate, restaurant-area:centre</i> S: I have 4 options for you ! How about the copper kettle ? It is located in the centre .			U: I would prefer it in the centre area . Actually I need a moderately priced restaurant . Are there any fitting that description ? This restaurant should serve british food too . D: <i>restaurant-food:british, restaurant-pricerange:moderate, restaurant-area:centre</i> S: I have 4 options for you ! It is in the centre area . That is a british restaurant . the copper kettle is just your thing . It is in the centre area .	
U: I need a table for 8 people . Can you try to book it at 16:30 . We will be there sunday night . D: <i>restaurant-people:8, restaurant-day:sunday, restaurant-time:16:30</i> S: Your reference number is 00000013 .			U: It will be for 8 people . Can you try for 16:30 ? I am also looking for a restaurant to book for sunday. D: <i>restaurant-people:8, restaurant-day:sunday, restaurant-time:16:30</i> S: All set . Your reference number is 00000013 .	
U: I would like to keep it in the moderate range , please . Oh , i need parking . D: <i>hotel-parking:yes, hotel-pricerange:moderate</i> S: I have 14 different options for you ! How about hamilton lodge ? Fits your request perfectly . It has free wifi .			U: Hello , I would like to find a hotel that includes free parking . I also need a moderate place to stay . D: <i>hotel-parking:yes</i> S: Reference number is : 00000013 .	
U: That sounds fine , I just need the postcode though . D: S: The postal code for that hotel is cb41da .			U: What is the postcode ? D: S: Thank you for using our services .	
U: I need a room starting on sunday . There is going to be 8 of us. How about only 2 nights . D: <i>hotel-day:sunday, hotel-people:8, hotel-stay:2</i> S: Here is the booking information : Booking was successful . Reference number is : 00000001 .			U: That sounds fine , I just need the postcode though . D: S: Glad to have been of help . Thank you for using the service . Enjoy the rest of your day !	
U: Thank you very much , goodbye . D: S: You are very welcome. Goodbye .			U: I just need the postcode . D: S: Ok , thank you . Have a good day .	
Success			Failure: System fails to track the value of price range in the dialog state and does not answer the postcode.	

Table 6: A sample dialog session comparison between *SYSTEM-1* and *SYSTEM-6* in simulated evaluation. For simplicity, we only present the updated slots and values at each turn, instead of the entire dialog state (lines start with D).

Train			Attraction	
Info	Reqt	Book	Info	Reqt
destination=peterborough day=saturday departure=cambridge leaveAt=13:00	duration trainID	people=7	area=centre	phone postcode
SYSTEM-1			SYSTEM-6	
U: I am looking for a train that leaves at 13:00	U: Hi! I'm looking for a train departing from Cambridge after 13:00			
S: What day would you like ? Where would you like your train to take you ? Where did you want to depart from ?	S: What day did you have in mind ? Where are you heading to ?			
U: I would like to leave on Saturday and depart from Cambridge	U: Saturday and heading to peterborough			
S: What is your destination ?	S: There is a train arriving at 13:00 would you like me to book tickets for that one ? I can book you on saturday. Would you like me to book a train to peterborough for you ? Would you like me to book a train from cambridge for you ?			
U: I need to go to Peterborough	U: Yes, please book for 7 people			
S: We can book you for the train leaving at 13:00 . I can book you on Saturday. Would you like me to book a train to peterborough for you ? Would you like me to book a train from cambridge for you ?	S: All set . Your reference number is 00000066 .			
U: Yes. I need the train ID and booking for 7 people	U: Thanks! I'm looking forward to seeing local tourist attractions			
S: Reference number is : 00000054 .	S: Thank you for using our services			
U: I want to see the tourist attractions so it should be near the Centre	U: Do you know of any attractions in the centre?			
S: There are 44 . Would you like me to recommend one for you ? Would you like vue cinema ?	S: We have 44 options , can i recommend for you ? I would suggest sheep's green and lammes land park fen causeway .			
U: Yes. I also want the phone number and postcode for the cinema	U: Can I get the phone number for Sheep's Green?			
S: The phone number is 08712240240 and postcode is cb11ps . They are located within the centre .	S: What time would you like to leave ?			
U: Do you have me booked for a party of 7 people?	U: Around noon			
S: Here is the booking information : Booking was successful . Reference number is : 00000054.	S: Have a good day .			
U: Are there 7 seats booked?	U: Can you give me the phone number of the tourist attraction?			
S: Reference number is : 00000054 .	S: Thank you for contacting the help desk . Have a great day .			
U: Thank You for your help and Goodbye!	U: Can you give me a list of attractions?			
S: You are welcome . Have a good day !	S: Have a good day .			
Success	Failure: System does not answer the phone number of the requested tourist attraction.			

Table 7: A sample dialog session comparison between SYSTEM-1 and SYSTEM-6 in human evaluation.

Similarity Scoring for Dialogue Behaviour Comparison

Stefan Ultes and Wolfgang Maier

Mercedes-Benz Research & Development

Sindelfingen, Germany

{stefan.ultes,wolfgang.mw.maier}@daimler.com

Abstract

The differences in decision making between behavioural models of voice interfaces are hard to capture using existing measures for the absolute performance of such models. For instance, two models may have a similar task success rate, but very different ways of getting there. In this paper, we propose a general methodology to compute the similarity of two dialogue behaviour models and investigate different ways of computing scores on both the semantic and the textual level. Complementing absolute measures of performance, we test our scores on three different tasks and show the practical usability of the measures.

1 Introduction and Related Work

Interacting with technical systems through voice is prevalent in our every day lives and in the focus of industry and research alike. For evaluating the behaviour of voice interfaces, interaction-based or corpus-based methods have been employed, both aiming at producing absolute measures like dialogue success. While this is clearly an important aspect of dialogue behaviour evaluation, it remains superficial and does not touch on the actual difference of two behaviour models.

The goal of this paper is to propose a method to quantify the similarity of two behaviour models—the learned or hand-crafted dialogue system decision—by means of a similarity score. The core idea is to use well-defined dialogue contexts—moments within a dialogue where the system needs to make a decision of how to respond—and compare the resulting system response of each behaviour model. We propose different similarity measures and demonstrate their usefulness in different scenarios.

Being able to compare behaviour models on a deeper level opens the door to a deeper understand-

ing of the learned behaviour. It aims to answer questions like:

1. When does the behaviour, i.e., the resulting response in a given context, of a reinforcement learning behaviour model converge?
2. Which effect do modifications of the learning parameters or learning set-up have, e.g. different random seeds (minor) or reward models (significant), on the resulting learned behaviour models? Do these modified behaviour models still result in exhibiting the same behaviour? What difference in behaviour causes the differences in absolute measures? Are there sub-sets of dialogue contexts that are fundamental for these differences?
3. How different are single responses of different behaviour models for the same given dialogue context?

These questions are of high relevance in cases where not only the average absolute performance is of interest but also the actual learned behaviour. On an application level, the answers to those question can help to decide which behaviour model to apply for a concrete live application, as they can support decision such as when to stop learning, or reveal the properties of different random seeds. From a more scientific point of view, the questions contribute to the overall problem of what we can learn about the interaction characteristics from the learned models.

The core task of a voice interface, also called spoken dialogue systems (SDS), is the decision of how to respond to a given user input and a dialogue context. This task is either modelled explicitly or implicitly. An explicit behaviour model usually comprises a distinct dialogue system module called dialogue policy taking in a dialogue state—a combined and dense representation of the current user

input interpretation and the dialogue context—and producing an abstract system response. In a subsequent step, this abstract system response is then transferred into text by a natural language generator. An implicit behaviour model uses a neural network to learn a text response directly based on text input thus combining user input interpretation, dialogue context integration, and dialogue response selection in one model.

Absolute measures to evaluate the performance of these behaviour models through the interaction with real or simulated users are, for example, task success or dialogue length (Gašić and Young, 2014; Lemon and Pietquin, 2007; Daubigney et al., 2012; Levin and Pieraccini, 1997; Young et al., 2013; Su et al., 2016; Ultes et al., 2015; Wen et al., 2017). Other measures are user satisfaction (Walker et al., 1997; Chu-Carroll and Nickerson, 2000; Dzikovska et al., 2011; Ultes et al., 2015; Wen et al., 2016; Ultes et al., 2017a) or quality of interaction (Möller et al., 2008; Schmitt and Ultes, 2015). All are often acquired through interaction-based studies¹.

Others have employed corpus-based evaluation by comparing textual system responses with transcriptions of actual interaction as absolute evaluation criterion where the response in the corpus is treated as ground truth (Serban et al., 2016; Sordoni et al., 2015; Li et al., 2016a; Lowe et al., 2015). Text comparison metrics like BLEU (Papineni et al., 2002) have been adopted from machine translation to evaluate how well the system response matches the one in the database, e.g., (Li et al., 2016b; Sordoni et al., 2015). This way of evaluating has been criticised widely as a system response that is different from the one in the database can still be a valid system response simply leading to a different subsequent dialogue. Furthermore, the BLEU score evaluation hardly correlates with human judgements (Liu et al., 2016; Novikova et al., 2017).

Dismissing text similarity measures as not useful for dialogue evaluation, however, is overhasty and shortsighted. While those measures may not help with absolute comparison of policies, they may be valuable to compare two policies with each other. In other words, they can help to reveal the similarity between two models without explicitly judging their performance.

In this work, we propose a framework to com-

pute the similarity of two dialogue behaviour models. This comprises the following contributions:

- a set-up to compare behaviour models on the level of single decisions
- similarity scores to compare behaviour models on the level of single decisions
- applications of similarity scoring offering a deeper understanding of the learned behaviour

The remainder of this paper is structured as follows: In Section 2, we introduce the general approach for quantifying the similarity of behaviour models. We then investigate the usability of several different ways of computing a similarity score in Sec. 3, considering scores on the semantic and the textual level. In Sec. 4 and 5, we describe our experimental setup, test our scores on three different tasks, and show their correlation confirming their practical usability.

2 Scoring Framework

To compare two dialogue behaviour models, this paper explores the usage of similarity measures instead of relying on absolute performance measures. The main idea is—in addition to knowing about the absolute performance—to learn about how similar or different two behaviour models are. For this, a defined set of dialogue contexts is applied to each behaviour model to generate corresponding system responses. These responses are then compared to learn about the overall similarity of the models. The general approach illustrated in Figure 1 is as simple as effective:

1. Define a set of dialogue contexts C .
2. Evaluate each behaviour model m in a deterministic way and collect the resulting system responses a_c^m for each context $c \in C$.
3. Calculate similarity scores $\sigma(a_c^m, a_c^{m'})$ for each system response pair $(a_c^m, a_c^{m'})$, e.g., by using one of the measures proposed in Section 3.

Aside from finding suitable similarity measures, one of the key challenges is to find good dialogue contexts that may be used as basis for comparison. Here, a dialogue context is a sub-dialogue either represented by a set of system utterances and user utterances (which is necessary, e.g., for end-to-end dialogue generators) or directly by the dense

¹For a good overview over absolute metrics including a taxonomy, please refer to Hastie (2012).

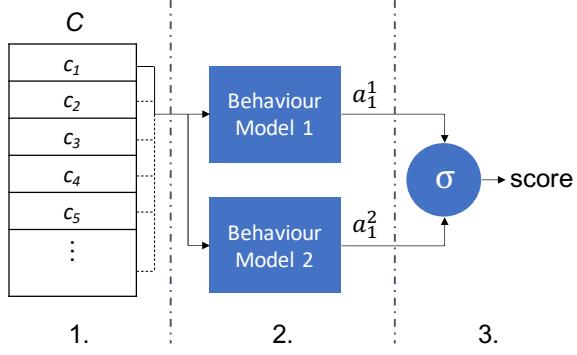


Figure 1: The three steps of the scoring framework.

representation of a dialogue state following the Markovian idea (often only available in modular dialogue systems). The proposed framework relies on a well-defined set of dialogue contexts, avoiding the evaluation of unrealistic situations which would directly influence the similarity scores.

In this work, the focus lies on modular dialogue systems where dialogue states are available to represent the dialogue context. Thus, there are two natural options of finding a set of dialogue contexts: collecting dialogues with corresponding dialogue states from actual real dialogues noted as C_{col} or generating a set of dialogue states noted as C_{gen} .

3 Similarity Scores

A similarity score is computed for the comparison of two behavioural models π and π' . Depending on the nature of the behavioural model, for each context $c_i \in C$, each may produce an abstract system response actions a_i , and an text response p_i . Each abstract system action $a_i = act_i(s_1^i = v_1^i, \dots, s_j^i = v_j^i)$ consists of a dialogue act act_i , representing the communicative function like *inform* or *request*, and a set S_i of j slot-value-pairs $S_i = \{(s_1^i, v_1^i), \dots, (s_j^i, v_j^i)\}$ representing the concepts and their respective values². To compute each similarity score, $|C|$ action/text response pairs are compared using the following similarity score measures.

Total Match Rate The total match rate (TMR) is based on a binary score that regards two actions a, a' as equal only if they completely match, i.e., $\delta_{a,a'} = 1$ iff $a = a'$, else 0. The TMR is then

²For the abstract system action $a = \text{inform}(\text{name}='Golden House', \text{area}=\text{centre})$, $act = \text{inform}$ and $S = \{(\text{name}, 'Golden House'), (\text{area}, \text{centre})\}$.

defined by

$$TMR = \frac{1}{|C|} \sum_{i=1}^{|C|} \delta_{a_i, a'_i}. \quad (1)$$

Dialogue Act Match Rate The dialogue act match rate (DMR) is based on a binary score comparing the actions a, a' where both match if the corresponding dialogue acts are the same: $\delta_{act, act'} = 1$ iff $act = act'$, else 0. The DMR is defined by

$$DMR = \frac{1}{|C|} \sum_{i=1}^{|C|} \delta_{act_i, act'_i}. \quad (2)$$

Concept Error Rate The concept error rate is a measure usually used for evaluating natural language understanding systems that translate text input to a semantic representation. The concept error rate then is computed by comparing the resulting semantic representation with a ground truth. Instead of comparing a semantic representation a with a ground truth, it can also be used to compare it to another representation a' produced by a different behaviour model.

Similar to the word error rate, it is based on the Levenshtein distance of two dialogue actions having one as hypothesis h and one as reference r :

$$dist(h, r) = \#ins + \#del + \#sub. \quad (3)$$

$\#ins$, $\#del$, and $\#sub$ are the number of insertions, deletions and substitutions, respectively, when computing the Levenshtein distance of the concepts of the sets S_1 and S_2 where each slot s_j^i and each value v_j^i are treated as individual items.

The concept error CE is then defined by

$$CE(h, r) = \frac{|r| - dist(h, r)}{|r|} \quad (4)$$

normalising the error by the length of r . Clearly, this is an asymmetric quantity. To make it symmetric, it is calculated using a and a' both as hypothesis and reference:

$$\tilde{CE}(a, a') = \frac{CE(a, a') + CE(a', a)}{2}. \quad (5)$$

The concept error rate is then calculated with

$$CER = \frac{1}{|C|} \sum_{i=1}^{|C|} \delta_{act, act'} \cdot \tilde{CE}(a_i, a'_i). \quad (6)$$

Concept Match Rate As an alternative to the asymmetric CER, we propose the symmetric concept match rate. Instead of basing it on an error comparing a hypothesis with a reference, it counts concepts γ that are present in both dialogue actions where $\tilde{m}(a, a', \gamma)$ defines if a match occurred:

$$\tilde{m}(a, a', \gamma) = \begin{cases} 1, & \text{if } \gamma \in S \text{ and } \gamma \in S' \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

The concept match CM takes into account the dialogue acts and the unified set of concepts $\tilde{S} = S_1 \cup S_2$ of both dialogue actions treating slots s and values v in \tilde{S} as individual γ :

$$\tilde{CM}(a, a') = \delta_{act, act'} + \sum_{\gamma \in \tilde{S}} \tilde{m}(a, a', \gamma) \quad (8)$$

A concept match of two dialogue actions a and a' is thus defined by

$$CM(a, a') = \frac{\tilde{CM}(a, a')}{1 + |\tilde{S}|} \quad (9)$$

and the concept match rate by

$$CMR = \frac{1}{|C|} \sum_{i=1}^{|C|} CM(a, a') . \quad (10)$$

Cosine Similarity and angular similarity The Universal Sentence Encoder (USE) (Cer et al., 2018) is a generic sentence encoder which employs two measures for the computation of the distances between encoded sentences, namely cosine similarity and angular similarity:

$$cosine-sim = \text{USE}(p) \cdot \text{USE}(p') \quad (11)$$

$$angular-sim = 1 - \frac{\arccos(cosine-sim)}{\pi} \quad (12)$$

BLEU The BLEU score (Papineni et al., 2002) is a measure used for the evaluation of machine translation systems. It is based on an n -gram precision φ_n , computed as the number of common n -grams between reference p and candidate phrase p' (and vice versa) divided by the number of n -grams of the candidate phrase. The score of a corpus is the geometric mean of modified precision scores multiplied with a brevity penalty v :

$$BLEU = v \cdot \exp(\sum_n w_n \log \varphi_n) , \quad (13)$$

where v is 1 if $|p| > |p'|$ and $e^{\frac{1-|p'|}{|p|}}$ otherwise. $BLEU$ is computed for multiple values of $n \leq 4$ and geometrically averaged (called BLUE-4). The final score is made symmetric in accordance with Eq. 5.

Table 1: Absolute results of the simulated experiments for R_{TS} and R_{IQ} after different number of training dialogues showing task success rate (TSR), average interaction quality (AIQ), and average dialogue length (ADL) in number of turns. Each value is computed after 100 evaluation dialogues averaged over three trials.

# Training Dialogues	TSR		AIQ		ADL	
	R_{TS}	R_{IQ}	R_{TS}	R_{IQ}	R_{TS}	R_{IQ}
1,000	0.98	0.99	3.78	3.85	4.46	4.44
5,000	0.99	0.98	3.78	3.81	4.41	4.51
10,000	1.00	0.98	3.81	3.80	4.32	4.47
15,000	1.00	0.99	3.79	3.81	4.36	4.43
20,000	1.00	0.97	3.86	3.73	4.15	4.62
25,000	1.00	0.98	3.77	3.85	4.37	4.30
30,000	1.00	0.96	3.71	3.87	4.49	4.41
35,000	0.99	0.96	3.73	3.84	4.42	4.46
40,000	1.00	0.94	3.77	3.77	4.35	4.75

BERTscore The BERTscore (Zhang* et al., 2020) is an automatic evaluation metric used for text generation that has shown a high correlation with human ratings. Given a function β which returns the BERT embedding (Devlin et al., 2018) for a given token, recall and precision along with the F1-score are computed for a reference p and a candidate p' as

$$R_{BERT} = \frac{1}{|p|} \sum_{p_i \in p} \max_{p'_j \in p'} \beta(p_i)^\top \beta(p'_j) , \quad (14)$$

$$P_{BERT} = \frac{1}{|p'|} \sum_{p'_j \in p'} \max_{p_i \in p} \beta(p_i)^\top \beta(p'_j) , \quad (15)$$

$$F_{BERT} = 2 \frac{R_{BERT} \cdot P_{BERT}}{R_{BERT} + P_{BERT}} . \quad (16)$$

F_{BERT} has been selected as a symmetric similarity score that also represents a reasonable balance between R_{BERT} and P_{BERT} .

Examples scores are shown in Appendix A.

4 Application Scenarios of Similarity Score Evaluation

We present three different scenarios addressing the following questions: When does the behaviour of a reinforcement learning policy converge? Which effect do modifications of the random seeds have on the resulting learned policies? Which effect do modifications of the reward models have on the resulting learned policies?

4.1 Evaluation Setup

To answer these question, we apply the following evaluation setup.

Table 2: Similarity measures for testing convergence of each trial (random seed) for R_{TS} employing task success and R_{IQ} employing interaction quality for C_{real} .

	<i>Trial</i>	# Training Dialogues	<i>TMR</i>	<i>DMR</i>	<i>CER</i>	<i>CMR</i>	<i>ang sim</i>	<i>cos sim</i>	<i>BLEU-4</i>	<i>BERTscore</i>
Task Success	0	10,000	0.978	0.978	0.978	0.980	0.863	0.868	0.470	0.905
		20,000	0.989	0.989	0.989	0.989	0.863	0.877	0.481	0.910
		30,000	0.984	0.995	0.991	0.990	0.874	0.888	0.518	0.916
		40,000	1.000	1.000	1.000	1.000	0.871	0.891	0.493	0.918
	1	10,000	0.945	0.978	0.962	0.953	0.855	0.843	0.507	0.907
		20,000	0.962	0.978	0.970	0.966	0.860	0.858	0.529	0.917
		30,000	0.995	0.995	0.995	0.995	0.874	0.885	0.517	0.922
		40,000	0.978	0.989	0.986	0.985	0.875	0.885	0.532	0.925
	2	10,000	0.885	0.940	0.917	0.907	0.837	0.815	0.462	0.893
		20,000	0.995	0.995	0.995	0.995	0.872	0.876	0.519	0.913
		30,000	0.984	0.989	0.988	0.988	0.868	0.880	0.485	0.911
		40,000	0.978	0.984	0.981	0.982	0.876	0.885	0.525	0.918
Interaction Quality	0	10,000	0.944	0.944	0.944	0.950	0.860	0.856	0.484	0.901
		20,000	0.972	0.972	0.972	0.972	0.859	0.863	0.441	0.897
		30,000	0.994	0.994	0.994	0.997	0.859	0.870	0.422	0.897
		40,000	0.978	0.983	0.980	0.984	0.867	0.880	0.461	0.901
	1	10,000	0.972	0.972	0.972	0.972	0.875	0.884	0.551	0.928
		20,000	0.966	0.978	0.974	0.973	0.837	0.827	0.463	0.899
		30,000	0.994	0.994	0.994	0.995	0.845	0.833	0.474	0.903
		40,000	0.994	0.994	0.994	0.996	0.849	0.846	0.491	0.907
	2	10,000	0.961	0.961	0.961	0.965	0.848	0.830	0.498	0.901
		20,000	0.978	0.978	0.978	0.978	0.837	0.820	0.458	0.895
		30,000	0.983	0.983	0.983	0.984	0.848	0.841	0.502	0.903
		40,000	0.989	0.989	0.989	0.990	0.846	0.837	0.495	0.902

4.1.1 Policy Training

For the evaluation, two policies are trained to reflect two different set-ups. One set-up uses the conventional task success as main reward component as heavily used within the literature (Gašić and Young, 2014; Vandyke et al., 2015; Su et al., 2016, e.g.) and the other set-up uses the interaction quality (IQ) (Schmitt and Ultes, 2015) representing user satisfaction as described by Ultes et al. (Ultes et al., 2017a; Ultes, 2019). IQ is defined on a five-point scale from five (satisfied) down to one (extremely unsatisfied). To derive a reward from this value,

$$R_{IQ} = -T + (iq - 1) \cdot 5 \quad (17)$$

is used where R_{IQ} describes the final reward. It is applied to the final turn of the dialogue of length T with a final IQ value of iq . Thus, a per-turn penalty of -1 is added to the dialogue outcome. This results in a reward range of 19 down to $-T$ which is consistent with related work in which binary task success (TS) was used to define the reward as:

$$R_{TS} = -T + \mathbb{1}_{TS} \cdot 20, \quad (18)$$

where $\mathbb{1}_{TS} = 1$ only if the dialogue was successful, $\mathbb{1}_{TS} = 0$ otherwise.

For each set-up, three policies with different random seeds were trained in a simulation environ-

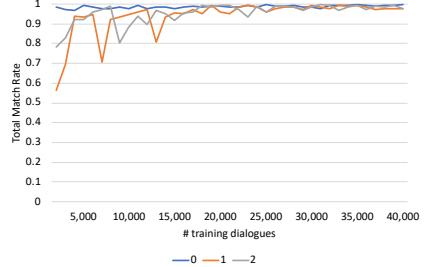


Figure 2: Convergence of each trial (random seed) for R_{TS} evaluated with the total match rate and on C_{col}^{TS} . The curves for R_{IQ} as well as C_{gen} set are similar.

ment using the PyDial Statistical Spoken Dialogue System Toolkit (Ultes et al., 2017b) with an agenda-based user simulator (Schatzmann and Young, 2009). For each trial, a GP-SARSA (Gašić and Young, 2014) policy model was trained—a learning algorithm known for its high sample-efficiency—with dialogues in the Cambridge restaurants domain about finding restaurants in Cambridge, UK. The domain comprises three slots used as search constraints (area, price range, food type). For belief state tracking—updating the probability distribution over all possible dialogue states in each turn—the focus belief tracker is used (Henderson et al., 2014). Prompts were generated using the SC-LSTM (Wen et al., 2015) natural language generator implementation of PyDial.

To ensure consistency, the standardised Environment 1 from Casanueva et al. (2017) is used. The interaction quality is estimated using a BiLSTM with self-attention as described by Ultes (2019).

For each trial of the task success and the interaction quality set-ups, a policy was trained with 40,000 dialogues and evaluated after each 1,000 training dialogues with 100 evaluation dialogues. The absolute performance of each set-up in terms of task success rate (TSR), average interaction quality (AIQ) as estimated at the end of each dialogue, and the average dialogue length (ADL) is shown in Table 1 averaged over all three trials.

4.1.2 Collected and Generated Context Sets

For computing the similarity scores described in Section 3, two types of dialogue context sets are used: collected dialogue contexts C_{col} and generated dialogue contexts C_{gen} .

The contexts of C_{col} are collected from the evaluation cycles of the 40,000 training batch of R_{TS} and R_{IQ} . From each trial, 10 evaluation dialogues are taken to constitute C_{col}^{TS} and C_{col}^{IQ} . This results

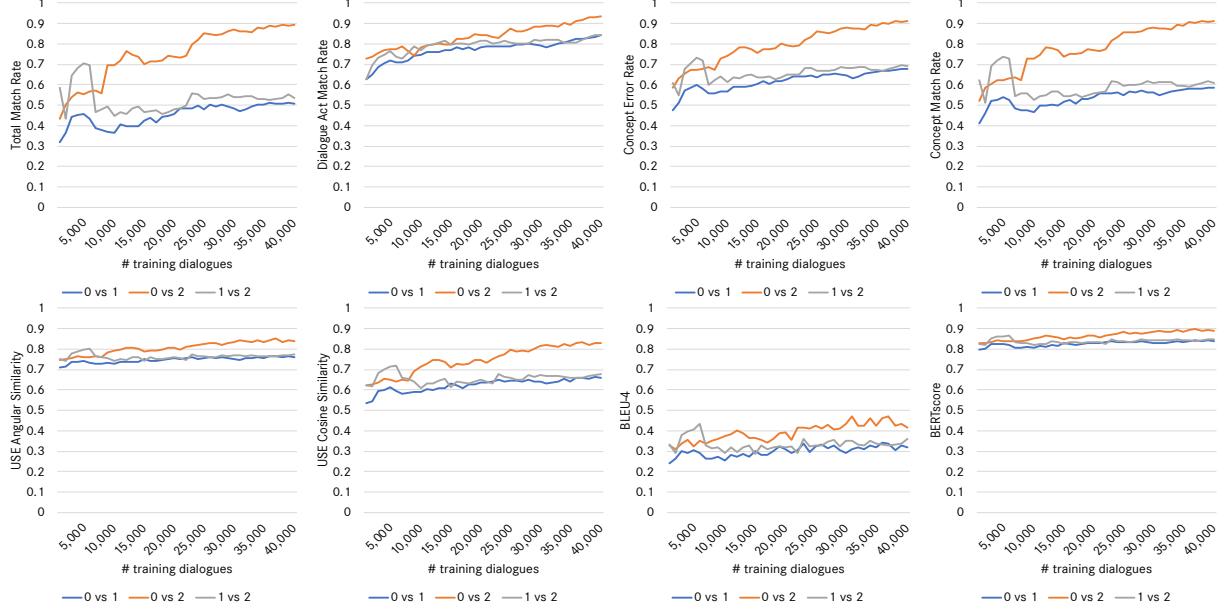


Figure 3: All similarity scores for C_{col}^{TS} for comparing the different trials / random seeds with each other (trial 0 vs. trial 1, trial 0 vs. trial 2, trial 1 vs. trial 2) of the R_{TS} policies evaluated after each training cycle of 1,000 dialogues.

in a total of 30 dialogues each with 183 dialogue contexts in C_{col}^{TS} and 178 collected dialogue contexts in C_{col}^{IQ} .

To generate dialogue contexts C_{gen} , the most relevant parts of a dialogue state are considered. For the Cambridge Restaurants domain, these are the three main search constraints *area*, *pricerange*, *foodtype* as well as the *method* of how to look for information. In the belief state used by PyDial, the joint probability of the dialogue state $P(s)$ is divided based on independence assumptions so that each slot probability is modelled separately. Hence, dialogue contexts are generated with probabilities for each slot in 0.1 steps, e.g., for a value of slot *area*³, dialogue contexts with a probability of 0.0, 0.1, 0.2, ..., 1.0, respectively, are created. With four slots and taking into account all possible slot and probability combinations, this results in a total of 1,296 generated dialogue contexts C_{gen} used for both R_{TS} and R_{IQ} .

4.2 Experiments and Results

The experimental results of applying above setup are described in the following.

4.2.1 Computing Similarity Scores to Test for Policy Convergence

The first scenario addresses the question if and when each policy converges in its behaviour.

³The actual value to pick is not important due to the way the dialogue state is used by the GP-SARSA algorithm.

Thus, a similarity score is computed comparing each policy before and after each training iteration, i.e., the additional training of 1,000 dialogues⁴. If the policy converges, the similarity score should be close to 1.0 for all similarity measures. The resulting similarity scores for C_{col}^{TS} and C_{col}^{IQ} for each reward and each trial are shown in Table 2. For convergence testing, the total match rate is used as the main criterion as two behaviour models are the same if they result in the exact same action for each dialogue context. The resulting learning curve for R_{TS} is shown in Figure 2 which is similar to the curve of R_{IQ} . Results for C_{gen} are omitted as they are almost identical to C_{col} . Notably, even though the differences are very small, all policies might still change after 40,000 training dialogues.

4.2.2 Computing Similarity Scores to Test for Seed Convergence

The second scenario addresses the question if and when policies trained with different random seeds. For this, each policy trained with R_{TS} and each policy trained with R_{IQ} are compared with the other policies trained with the same reward at each training iteration. As there are three trials / random seeds for each set-up, this results in three

⁴A policy after 2,000 training dialogues is compared with the same policy after 1,000 training dialogues, then again the policy after 3,000 training dialogues with the policy after 2,000 training dialogues, and so on.

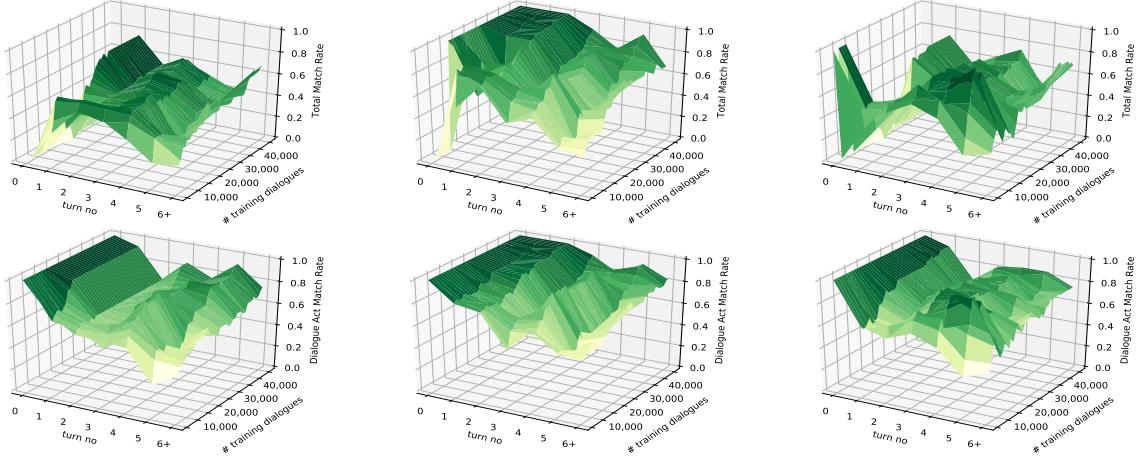


Figure 4: Turn-based results for total match rate (top row) and dialogue act match rate (bottom row) for all three trials using the TS-based reward.

comparisons for both R_{TS} and R_{IQ} ⁵. If the trials converge to the same policy, the similarity score is close to 1.0 for all similarity measures. The resulting similarity scores for the respective $C_{col}^{TS}/C_{col}^{IQ}$ and C_{gen} for each reward are shown in Table 4 with a visualisation for R_{TS} on C_{col}^{TS} for all metrics and training iterations in Figure 3.

Evidently, neither the policies of R_{TS} nor the policies of R_{IQ} converge to the same behaviour. Instead, they only reach a maximal TMR of 0.896 for R_{TS} and 0.68 for R_{IQ} for only one pair in each case using C_{col} . Even though the policies do not converge to the identical behaviour, the convergence in terms of DMR is much better and all policy models tend to learn the same basic behaviour—the respective dialogue acts—*independent* of the random seed that is used.

Comparing the scores of $C_{col}^{TS}/C_{col}^{IQ}$ with C_{gen} shows that for the latter, the scores are much lower but the overall tendencies of the similarity scores are the same. This shows that the basis that is used is important when looking at absolute scores but not relevant when only the tendency is of interest. One explanation for this difference in absolute scores is that C_{col} may contain more dialogue contexts that are very similar to each other where the policies rather agree. C_{gen} contains each dialogue context only once. Additionally, C_{gen} may also contain dialogue contexts that have not been visited during training or evaluation and thus it is harder for the policy model to learn consistent behaviour.

⁵For example, the policy of trial 1 after 3,000 training dialogues is compared with the policy of trial 2 after 3,000 training dialogues, the policy of trial 1 after 4,000 training dialogues is compared with the policy of trial 2 after 4,000 training dialogues, and so on.

Analysing all used similarity scores generally, Figure 3 shows that for all similarity scores except DMR, the curves are similar in terms of shape but different in terms of scores and differences between trials. Each of the text-based scores angular similarity, BLEU-4 and BERTscore seems to produce values in the same range within one set-up. Thus, the scores are not very suitable for comparison.

For R_{TS} on C_{col}^{TS} , a more detailed analysis has been conducted on the similarities of the dialogue behaviour models with respect to the progression through the dialogue, i.e., what are the similarity scores when only looking at the first turn, the second turn, etc. Figure 4 shows that, for the first system turn, behaviour is learned where both models either always agree or always disagree in terms of TMR but always agree in terms of DMR. Again, the agreement on the communicative function is evident. This is not surprising as in the beginning, the system needs to acquire information from the user with the *request* dialogue act.

4.2.3 Computing Similarity Scores to Compare Policies from Different Reward Models

The final scenario addresses the question of how similar the dialogue behaviour of two models is that are trained with the different rewards R_{TS} and R_{IQ} . As common base, both collected contexts are combined to $C_{col}^{TS+IQ} = C_{col}^{TS} \cup C_{col}^{IQ}$. The results are shown in Table 3 with the TMR and DMR compared to the results of scenario 2 in Figure 5.

The cross-comparison of R_{TS} and R_{IQ} shows that the TMR and DMR are a bit lower than for the comparison of policies within R_{TS} and R_{IQ} , re-

Table 3: All similarity scores for comparing the respective policies trained with R_{TS} and R_{IQ} with each other using C_{gen}^{TS+IQ} after 40,000 training dialogues each.

$TS \text{ vs } IQ$	TMR	DMR	CER	CMR	$USE(\text{ang})$	$USE(\cos)$	$BLEU-4$	$BERTscore$
$0 \text{ vs } 0$	0.645	0.825	0.739	0.695	0.783	0.700	0.325	0.845
$0 \text{ vs } 1$	0.310	0.789	0.556	0.428	0.725	0.589	0.243	0.808
$0 \text{ vs } 2$	0.288	0.756	0.529	0.413	0.723	0.585	0.240	0.801
$1 \text{ vs } 0$	0.338	0.867	0.606	0.458	0.735	0.608	0.263	0.826
$1 \text{ vs } 1$	0.485	0.831	0.665	0.574	0.778	0.698	0.329	0.843
$1 \text{ vs } 2$	0.380	0.798	0.596	0.486	0.757	0.658	0.281	0.825
$2 \text{ vs } 0$	0.615	0.803	0.709	0.668	0.776	0.682	0.313	0.840
$2 \text{ vs } 1$	0.307	0.759	0.542	0.421	0.722	0.576	0.246	0.803
$2 \text{ vs } 2$	0.296	0.748	0.527	0.414	0.722	0.579	0.232	0.799

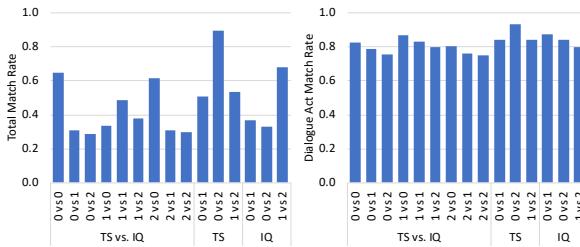


Figure 5: Total match rates and dialogue act match rates for the cross compare experiments computing the similarity scores for each policy of R_{TS} with each policy of R_{IQ} after 40,000 training dialogues. Along with that, the results of the internal policy similarity scores of R_{TS} and R_{IQ} shown for comparison.

spectively, but generally, the differences are similar. This means that, generally, the differences of policies trained with R_{TS} compared to policies trained with R_{IQ} are not much bigger than just using a different random seed.

5 Correlation of Scores

To analyse how complementary the different scores are, all system behaviour pairs of all experiments have been used to compute correlation and mean squared error for each score pair. The results are shown in Figure 6. An interesting finding is that the CER and CMR have a very high correlation and a very low error. Thus, both seem to capture the same similarities. In contrast to that, the DMR has a very low correlation with other scores and thus does provide additional information. BLEU-4 also does not have a high correlation with other metrics but does also not provide a huge variety, as shown in the example in Figure 3. Comparing semantic-based similarity scores with text-based similarity scores shows that CER and USE-based cosine distance have a quite high correlation and a

relatively small mean squared error. Thus, the similarity of two systems that provide semantic output and the similarity of two other systems that only provide text output can be comparably quantified with the CER and the USE-based cosine distance.

The overall total match rate of the samples used for calculating the correlation and mean squared error is 63.3%. Thus, the matches govern the correlation scores. Computing the correlation only on the samples that do not match reveals slightly different numbers that still match the overall impression. The main difference is that the correlation between CER and CMR drops down to 0.466.

6 Conclusion

This work proposes a first step towards a more detailed analysis of dialogue behaviour models by proposing a framework to compute similarity scores. A similarity score is meant to quantify how similar the decisions made by one dialogue behaviour model are compared to a second dialogue behaviour model. Using a fixed set of dialogue contexts, each model is evaluated and the resulting system responses—as semantic representations and/or as text utterances—are captured and used for the similarity score. We proposed eight similarity scores and applied them to three different scenarios.

By doing that, we were able to validate supposed certainties about reinforcement-based policy learning. We could observe that in the used set-ups, all policy models converged towards a fixed behaviour while still showing minor behavioural changes even after a very large number of training iterations.

Modifications of the random seeds, however, already result in a noticeable differences in the converged behaviour in the applied evaluation setup. The quantified differences are even similar in magnitude to a modification of the reward model, i.e., changing a random seed has a similar effect on the learned policy as switching from task success to interaction quality as the principal reward component.

Out of the eight proposed similarity scores, many seem to capture different aspects of similarity, so it remains to the application to decide which score is more useful. Only text-based scores coming from the language translation field like BLEU and BERTscore seem not to be too useful. One reason for this might be the dependency of the absolute score on the prompt length: quantifying textual

Table 4: All similarity measures for comparing the trials (random seeds) with each other for R_{TS} employing task success and for R_{IQ} employing interaction quality as main reward component. Results for the respective C_{col} are on the left and C_{gen} are on the right.

		Comparison		# Training Dialogues		TMR		DMR		CER		CMR		ang sim		cos sim		BLEU-4		BERTscore	
Task Success		$O_{vs. I}$		10,000	0.366	0.749	0.568	0.467	0.729	0.592	0.254	0.807	10,000	0.199	0.526	0.363	0.317	0.666	0.454	0.139	0.754
		$O_{vs. 2}$		20,000	0.459	0.781	0.627	0.540	0.755	0.638	0.311	0.832	20,000	0.259	0.542	0.400	0.362	0.677	0.475	0.163	0.762
		$O_{vs. 3}$		30,000	0.486	0.792	0.645	0.561	0.752	0.639	0.290	0.828	30,000	0.357	0.586	0.471	0.436	0.700	0.525	0.209	0.783
		$O_{vs. 4}$		40,000	0.508	0.842	0.678	0.585	0.762	0.657	0.317	0.840	40,000	0.398	0.615	0.507	0.465	0.712	0.548	0.233	0.790
Interaction Quality		$O_{vs. I}$		10,000	0.694	0.781	0.744	0.727	0.792	0.715	0.373	0.852	10,000	0.660	0.696	0.678	0.734	0.781	0.708	0.343	0.845
		$O_{vs. 2}$		20,000	0.738	0.842	0.794	0.770	0.805	0.746	0.393	0.865	20,000	0.707	0.732	0.720	0.764	0.792	0.726	0.357	0.854
		$O_{vs. 3}$		30,000	0.869	0.885	0.879	0.880	0.836	0.815	0.433	0.887	30,000	0.752	0.775	0.763	0.800	0.804	0.754	0.369	0.865
		$O_{vs. 4}$		40,000	0.896	0.934	0.915	0.914	0.840	0.828	0.416	0.890	40,000	0.823	0.843	0.832	0.854	0.825	0.795	0.406	0.886
Task Success		$I_{vs. I}$		10,000	0.448	0.770	0.613	0.525	0.742	0.611	0.291	0.820	10,000	0.258	0.532	0.395	0.383	0.683	0.488	0.171	0.774
		$I_{vs. 2}$		20,000	0.481	0.814	0.651	0.560	0.759	0.651	0.320	0.835	20,000	0.305	0.577	0.441	0.419	0.693	0.510	0.194	0.782
		$I_{vs. 3}$		30,000	0.541	0.814	0.681	0.610	0.771	0.675	0.351	0.843	30,000	0.371	0.580	0.476	0.463	0.705	0.537	0.217	0.791
		$I_{vs. 4}$		40,000	0.536	0.842	0.692	0.609	0.774	0.679	0.361	0.850	40,000	0.405	0.612	0.508	0.476	0.714	0.552	0.231	0.795
Interaction Quality		$O_{vs. I}$		10,000	0.371	0.792	0.581	0.473	0.736	0.599	0.290	0.826	10,000	0.011	0.601	0.306	0.154	0.635	0.404	0.079	0.751
		$O_{vs. 2}$		20,000	0.337	0.781	0.565	0.450	0.720	0.575	0.237	0.803	20,000	0.066	0.710	0.388	0.205	0.649	0.435	0.104	0.759
		$O_{vs. 3}$		30,000	0.354	0.843	0.602	0.471	0.735	0.612	0.260	0.819	30,000	0.157	0.730	0.443	0.282	0.670	0.475	0.142	0.775
		$O_{vs. 4}$		40,000	0.365	0.876	0.618	0.480	0.741	0.619	0.283	0.826	40,000	0.248	0.747	0.497	0.356	0.692	0.520	0.175	0.790
Task Success		$I_{vs. I}$		10,000	0.320	0.764	0.548	0.434	0.713	0.556	0.247	0.801	10,000	0.003	0.856	0.430	0.188	0.653	0.458	0.085	0.754
		$I_{vs. 2}$		20,000	0.343	0.860	0.603	0.462	0.732	0.596	0.265	0.813	20,000	0.002	0.863	0.432	0.190	0.653	0.456	0.077	0.750
		$I_{vs. 3}$		30,000	0.337	0.876	0.607	0.468	0.726	0.596	0.242	0.810	30,000	0.117	0.843	0.480	0.285	0.679	0.512	0.129	0.774
		$I_{vs. 4}$		40,000	0.331	0.843	0.589	0.459	0.736	0.613	0.263	0.820	40,000	0.206	0.813	0.510	0.345	0.695	0.541	0.161	0.784

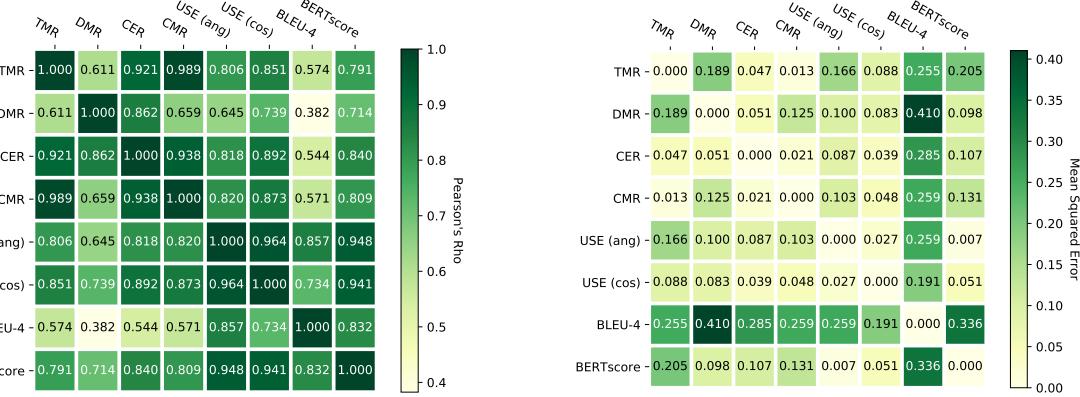


Figure 6: Correlation coefficients (left) and mean squared error (right) when comparing all similarity scores for all experiments.

difference in this way does not necessarily capture the relevant semantic differences.

Moreover, the set of dialogue contexts has a high impact on the absolute score for all similarity metrics but not on the trend when comparing two dialog behaviour models with each other.

For future work, the analysis must be more fine-grained, e.g., by sub-dividing the set of dialogue contexts into meaningful sub-sets. Furthermore, the proposed evaluation method is also suitable for directly looking at the actual behaviour of models by identifying crucial dialogue contexts and comparing the actual system reaction.

References

- Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Stefan Ultes, Lina Rojas-Barahona, Steve Young, and Milica Gašić. 2017. A benchmarking environment for reinforcement learning based task oriented dialogue management. In *Deep Reinforcement Learning Symposium, 31st Conference on Neural Information Processing Systems (NIPS)*.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

- Jennifer Chu-Carroll and Jill Suzanne Nickerson. 2000. Evaluating automatic dialogue strategy adaptation for a spoken dialogue system. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, page 202–209, USA. Association for Computational Linguistics.
- Lucie Daubigney, Matthieu Geist, and Olivier Pietquin. 2012. **Off-policy Learning in Large-scale POMDP-based Dialogue Systems.** In *Proceedings of the 37th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pages 4989–4992, Kyoto (Japan). IEEE.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Myroslava O Dzikovska, Johanna D Moore, Natalie Steinhauser, and Gwendolyn Campbell. 2011. Exploring user satisfaction in a tutorial dialogue system. In *Proceedings of the SIGDIAL 2011 Conference*, pages 162–172. Association for Computational Linguistics.
- Milica Gašić and Steve J. Young. 2014. Gaussian processes for POMDP-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40.
- Helen Hastie. 2012. **Metrics and evaluation of spoken dialogue systems.** In Oliver Lemon and Olivier Pietquin, editors, *Data-Driven Methods for Adaptive Spoken Dialogue Systems*, pages 131–150. Springer New York.
- Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The second dialog state tracking challenge. In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, volume 263.
- Oliver Lemon and Olivier Pietquin. 2007. Machine learning for spoken dialogue systems. In *European Conference on Speech Communication and Technologies (Interspeech'07)*, pages 2685–2688.
- Esther Levin and Roberto Pieraccini. 1997. A stochastic model of computer-human interaction for learning dialogue strategies. In *Eurospeech*, volume 97, pages 1883–1886.
- Jiwei Li, Michel Galley, Chris Brockett, Georgios Spathourakis, Jianfeng Gao, and Bill Dolan. 2016a. **A persona-based neural conversation model.** In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany. Association for Computational Linguistics.
- Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. **Deep reinforcement learning for dialogue generation.** In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.
- Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.
- Ryan Lowe, Nissan Pow, Iulian Vlad Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.
- Sebastian Möller, Klaus-Peter Engelbrecht, and Robert Schleicher. 2008. **Predicting the quality and usability of spoken dialogue services.** *Speech Communication*, 50(8-9):730–744.
- Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. **Why we need new evaluation metrics for NLG.** In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. **Bleu: a method for automatic evaluation of machine translation.** In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Jost Schatzmann and Steve J. Young. 2009. The hidden agenda user simulation model. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(4):733–747.
- Alexander Schmitt and Stefan Ultes. 2015. **Interaction quality: Assessing the quality of ongoing spoken dialog interaction by experts—and how it relates to user satisfaction.** *Speech Communication*, 74:12–36.
- Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarah Chandar, Aaron Courville, and Yoshua Bengio. 2016. **Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus.** In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 588–598, Berlin, Germany. Association for Computational Linguistics.
- Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. **A neural network approach to context-sensitive generation of conversational responses.** In *Proceedings*

of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 196–205, Denver, Colorado. Association for Computational Linguistics.

Pei-Hao Su, M. Gašić, N. Mrkšić, L. Rojas-Barahona, Stefan Ultes, D. Vandyke, T. H. Wen, and S. Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441. Association for Computational Linguistics.

Stefan Ultes. 2019. Improving interaction quality estimation with BiLSTMs and the impact on dialogue policy learning. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 11–20, Stockholm, Sweden. Association for Computational Linguistics.

Stefan Ultes, Paweł Budzianowski, Iñigo Casanueva, Nikola Mrkšić, Lina Rojas-Barahona, Pei-Hao Su, Tsung-Hsien Wen, Milica Gašić, and Steve Young. 2017a. Domain-independent user satisfaction reward estimation for dialogue policy learning. In *Interspeech*, pages 1721–1725. ISCA.

Stefan Ultes, Matthias Kraus, Alexander Schmitt, and Wolfgang Minker. 2015. Quality-adaptive spoken dialogue initiative selection and implications on reward modelling. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 374–383. ACL.

Stefan Ultes, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gašić, and Steve Young. 2017b. Py-Dial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78, Vancouver, Canada. Association for Computational Linguistics.

David Vandyke, Pei-Hao Su, Milica Gašić, Nikola Mrkšić, Tsung-Hsien Wen, and Steve Young. 2015. Multi-domain dialogue success classifiers for policy training. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 763–770. IEEE.

Marilyn Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: a framework for evaluating spoken dialogue agents. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics (EACL)*, pages 271–280, Morristown, NJ, USA. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, pages 120–129, San Diego, California. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*, pages 438–449. ACL.

Steve J. Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Example Similarity Scores

	Action a / Prompt p	Action a' / Prompt p'	TM	DM	C_E	CM	$USE(\text{ang})$	$USE(\cos)$	BLEU-4	BERTscore
0	<i>request(pricerange)</i> what price range are you interested in	<i>request(food)</i> what kind of food are you looking for	0	1	0.5	0.2	0.68	0.54	0.07	0.82
1	<i>request(food)</i> what type of food are you looking for	<i>request(pricerange)</i> what price range are you interested in	0	1	0.5	0.2	0.67	0.52	0.07	0.83
2	<i>inform(food="mediterranean", pricerange="expensive", name="la mimosa", area="centre")</i> la mimosa is a expensive mediterranean restaurant in the centre area	<i>inform(food="mediterranean", pricerange="expensive", name="la mimosa", area="centre")</i> la mimosa is a expensive mediterranean restaurant in the centre area	1	1	1	1	1.00	1.00	1.00	1.00
3	<i>inform(food="mediterranean", pricerange="expensive", name="shiraz restaurant", area="centre")</i> shiraz restaurant is in the centre area and is expensively priced and serves mediterranean food	<i>inform(food="mediterranean", pricerange="expensive", name="shiraz restaurant", area="centre")</i> shiraz restaurant is in the centre area and serves mediterranean food and is expensively priced	1	1	1	1	0.95	0.99	0.80	0.98
4	<i>inform(food="mediterranean", pricerange="expensive", name="la mimosa", area="centre")</i> la mimosa is in the centre area that is expensively priced and serves mediterranean food	<i>request(pricerange)</i> what price range would you like	0	0	0	0.1	0.53	0.09	0.00	0.61

Collection and Analysis of Dialogues Provided by Two Speakers Acting as One

Tsunehiro Arimoto¹, Ryuichiro Higashinaka¹, Kou Tanaka¹, Takahito Kawanishi¹
Hiroaki Sugiyama¹, Hiroshi Sawada¹, and Hiroshi Ishiguro²

¹NTT Communication Science Laboratories

²Osaka University

{arimoto.tsunehiro.ub, ryuichiro.higashinaka.tp, kou.tanaka.ef
takahito.kawanishi.fx, hiroaki.sugiyama.kf, hiroshi.sawada.wn } @hco.ntt.co.jp
ishiguro@irl.sys.es.osaka-u.ac.jp

Abstract

We are studying a cooperation style where multiple speakers can provide both advanced dialogue services and operator education. We focus on a style in which two operators interact with a user by pretending to be a single operator. For two operators to effectively act as one, each must adjust his/her conversational content and timing to the other. In the process, we expect each operator to experience the conversational content of his/her partner as if it were his/her own, creating efficient and effective learning of the other's skill. We analyzed this educational effect and examined whether dialogue services can be successfully provided by collecting travel guidance dialogue data from operators who give travel information to users. In this paper, we report our preliminary results on dialogue content and user satisfaction of operators and users.

1 Introduction

Such dialogue services as counseling (Dowling and Rickwood, 2013) are often provided through telecommunication systems that enable speakers (typically called operators) to talk from remote places (Crabtree et al., 2006; Sakamoto et al., 2007; Yamashita et al., 2011; Kristoffersson et al., 2013). For such services to be more productive, it is desirable that the skills of the operators are improved.

In this paper, we propose a unique learning style in which multiple operators with different skills cooperate and pretend to be one person (Fig. 1). For two operators to effectively act as one, each must adjust his/her conversational content and timing to the other. In this style, each operator may experience the conversational content of his/her partner as if it were his/her own, creating efficient and effective learning of the other's skill. Users also benefit; they do not have to interact

ID	Spk	Utterance
1	U	Hello. I am planning trips to Nara and Osaka prefectures. What sightseeing spots do you recommend?
2	GN	Hello. In the Nara area, I recommend Todaiji Temple and Nara Park.
3	U	I see. How can I get to them?
4	GN	You can walk to Todaiji Temple from Kin-tetsu Nara Station through Nara Park.
5	U	Thank you. How about Osaka?
6	GN	(Your turn.)
7	GO	(Ok.)
8	GO	Well, in Osaka, I recommend Osaka Castle and Universal Studios.
9	U	Those are both famous.
10	GO	You can easily get to them by train.
11	U	I'm glad they are so convenient. By the way, in Nara, do you recommend any restaurants where I can eat local food around those two spots?
12	GO	(Why don't you answer?)
13	GN	(Sure.)
14	GN	I recommend Asuka Nabe.
15	U	I see. Any idea how much it costs?

Figure 1: Example of Mixto1 condition where two guides with different skills pretend to be one guide who talks to a user (U). One guide has knowledge about travel in Nara (GN), and the other knows Osaka (GO). For readability, user utterances are shown in bold. Parentheses represent invisible to a user.

with a lot of operators and can establish one-to-one relationships. There were studies that aimed at increasing the perceived number of speakers for better interaction despite that there is only a single operator (Yamane et al., 2011; Arimoto et al., 2014); our idea here is the opposite.

Many prior studies exist where multiple actors work together to provide dialogue services. Cooperative architectures with multiple agents or human operators have attracted attention with regards to the development of dialogue systems (Lin et al., 1999; Komatani et al., 2009;

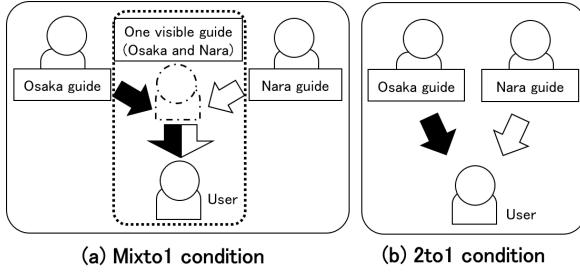


Figure 2: Cooperation style of Osaka and Nara guides under Mixto1 and 2to1 conditions

Nakano et al., 2011) as well as Wizard-of-Oz systems (Marge et al., 2016; Abbas et al., 2020). Users talking with a dialogue assistant controlled by multiple speakers on the cloud are reported to receive more reasonable responses (Lasecki et al., 2013). However, no research has examined the basic effect of behaving as one speaker on the satisfaction of the operators and their interlocutors. It remains especially unclear whether multiple operators who are acting as one promote mutual skill learning.

The following is the contribution of this study. First, we show a method for collecting text-chat dialogues in which two speakers acting as one person. Second, we show the basic effects of two speakers who are pretending to be just one person on the dialogue’s content and the satisfaction of the operators and the interlocutors.

2 Collection of text chats in which two speakers act as one

2.1 Dialogue design

Our study focuses on the dialogue services of two human operators with different knowledge. With different knowledge, the two operators can provide a larger variety of information than when they are separate. We collected travel guide text-chat dialogues about two neighboring prefectures. The dialogues were conducted by either one or two operators. We categorized the travel guidance knowledge for one prefecture as each operator’s skill. We have the following three conditions for conducting a dialogue:

Mixto1 condition Two operators with different specialties (as their skills) acted as one speaker. For example, we paired an operator who is familiar with Osaka prefecture and another who is familiar with Nara prefecture. Nara and Osaka are geographically adjacent.

They acted as one visible guide with knowledge of both prefectures (Fig. 2(a)).

2to1 condition Two operators with different specialties took turns talking directly (Fig. 2(b)) with one user in a three-party dialogue. This condition was collected as a baseline to evaluate the validity of the Mixto1 condition.

1to1 condition One operator gave recommendation to one user about two prefectures. The operator has much knowledge about one of them, but the other is outside his/her skill set.

Collaborative dialogues (Mixto1 and 2to1 conditions) are expected to positively affect the operators’ learning. We collected the 1to1 condition dialogues before and after the Mixto1 and 2to1 conditions to examine such educational effects.

2.2 Environment

All the speakers used Slack¹ to communicate in a text-chat format. They played either a guide (operator) or a user.

In the Mixto1 condition, two guides acted as one guide and interacted with one user. Each guide opened two Slack windows in one display. One window was used to interact with the user, and the other was used to consult with the other guide. The guides discussed their strategy for talking with the user in a window hidden from the user. The user opened a window to interact with the guide in one display and talked with both guides about his/her trip to the two pre-designated prefectures. The two guides used the same account to talk to the user; the user didn’t realize he/she was talking to two guides.

In the 2to1 condition, two guides and one user also participated in the dialogue as in the Mixto1 condition. However, both talked to the user using different accounts. Each guide opened a window to interact with the user without opening an additional window to just interact with the other guide.

In the 1to1 condition, one operator and one user each opened a window and directly interacted with each other.

2.3 Subjective questionnaires

Since it is unclear how our collected interactions affected the satisfaction of the guides, they answered a 12-item subjective questionnaire to assess task achievement and their impressions of

¹<https://slack.com>

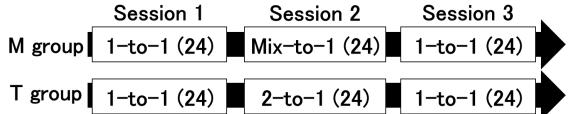


Figure 3: Data collection. Numbers in parentheses represent the number of interactions.

each conversation. For the Mixto1 and 2to1 conditions, the guides also answered three items about their impressions of performing the conversation as one or two people. They also freely described their experience at the end of Session 2 (Section 2.4). The users answered ten items regarding their impressions of the task achievement and the conversations.

2.4 Data collection

We recruited speakers to act as operators or users. The operators and users were paid for their participation. All dialogues were conducted in Japanese. Sixteen operators participated as guides. Operators were assigned to their home prefecture as their specialty (we assume that operators were knowledgeable about their home prefectures). Their ages ranged from 20 to 50 years, with six males and ten females. Two guides of the same gender from neighboring prefectures were paired.

Forty-eight speakers (16 males and 32 females) whose ages ranged from 20 to 50 participated in the dialogues as users. Each participated in a travel guide dialogue outside their home prefecture.

We collected the data over three sessions (Fig. 3). All the guides participated in all three sessions. Sixteen were divided into two groups of eight; the M group having the Mixto1 condition and the T group having the 2to1 condition in Session 2. Users participated in only one of the sessions and talked three times with different guides or guide pairs under the same condition. Each dialogue lasted ten minutes. We collected 144 travel text-chat dialogues and questionnaires from each guide and each user. The following are the descriptions of Sessions 1 to 3:

Sessions 1 and 3 All guides talked under the 1to1 condition. Each guide had text chats three times with a different user in each dialogue. We collected 48 dialogues for each session.

Session 2 The M group’s guide pair worked under the Mixto1 condition and the T group worked under the 2to1 condition. Each guide pair

had text chats six times with a different user in each dialogue. Therefore, we collected 24 Mixto1 dialogues and 24 2to1 dialogues.

3 Analysis

3.1 Approach

Evaluation of dialogue flows Using the collected text chat, we qualitatively analyzed how the guides facilitated the travel decisions under each condition. Under the 1to1 condition, the guides had limited knowledge that assisted them with travel to prefectures outside their specialty. Under the Mixto1 and 2to1 conditions, the operator had the opportunity to provide trip guidance while talking in turns with the other guide. We observed how the guides made recommendations based on the conditions.

Number of guide utterances for non-specialty prefectures The guides touched on the expertise of the other guides under the Mixto1 and 2to1 conditions. These guides may have gained information about the non-specialized prefectures from the conversations of the other guides, educating them about these unfamiliar prefectures. We analyzed whether the Mixto1 condition, acting as a single guide, increases the utterances of the non-specialized prefectures of guides.

3.2 Results

Dialogue flows The actual examples of collected dialogues for the Mixto1 and 2to1 conditions are shown in Figs 1 and 4 (translated from Japanese to English by authors).

In the 1to1 condition, the guide talked one-to-one with one user. In some scenes, the guide was unable to answer questions outside his specialty. For example, the guides frequently said “I’m sorry I don’t know” in the conversation.

In the Mixto1 example (Fig 1), two guides provided travel recommendations for Nara and Osaka prefectures. For “I am planning trips to Nara and Osaka prefectures. What sightseeing spots do you recommend? (ID = 1),” the Nara guide introduced Nara (“Hello. In the Nara area, I recommend Todaiji (ID = 2)”) and the Osaka guide introduced Osaka (“Well, in Osaka, I recommend Osaka Castle and Universal Studios (ID = 8)”). By using the window that was hidden from the user, the guides could consult when to switch among themselves (e.g., “Your turn (ID = 6)” and “Ok (ID = 7)”).

ID	Spk	Utterance
1	U	Nice to meet you. I am thinking of traveling around Fukuoka and Kumamoto for 3 or 4 nights. I'd like to go to Aso Nature Land, Dazifu Tenmangu, and the food stalls in Nakasu. What other places do you recommend?
2	GK	If you have time, I recommend Kurokawa Onsen.
3	U	I see. I also want to go to a hot spring. This'll be my first visit to Fukuoka.
4	GF	Hello. If you have time, how about Moji Port in Kitakyushu City because its retro streets are cute. Lots of fancy cafes and souvenir shops, too.
5	U	Oh, that sounds nice. I love eating, too.
6	GF	If you are looking for gourmet food, the iron-pot gyoza and mizutaki around Haruyoshi are delicious in Fukuoka.
7	GK	In Kumamoto, the Kumamoto ramen, basashi, and red ox dishes are famous.

Figure 4: Example of 2to1 condition: U, GF, and GK represent user, guide for Fukuoka, and guide for Kumamoto. For readability, user utterances are shown in bold.

In the 2to1 condition, the two guides talked individually to directly help the user. Figure 4 shows the travel guide dialogue for Kumamoto and Fukuoka prefectures by the Kumamoto and Fukuoka guides. Both guides talked about their specialty. The Fukuoka guide said, “the iron-pot gyoza and mizutaki around Haruyoshi are delicious in Fukuoka (ID = 6).” The Kumamoto guide said, “In Kumamoto, the Kumamoto ramen, basashi, and red ox dishes are famous (ID = 7).”

These observations show that the guides had the opportunity to provide trip assistance while speaking in turns with the other guide under the Mixto1 and 2to1 conditions.

Number of guide utterances for non-specialty prefectures We annotated whether each utterance in the dialogue was related to each of the two prefectures and counted the number of utterances of the guides for their non-specialized prefectures. For each group (M and T), we analyzed whether there was a difference in the number of utterances in Sessions 1 and 3 before and after completing Session 2.

A Wilcoxon’s rank-sum test showed that the M group under the Mixto1 condition showed a significant tendency to increase the number of utterances regarding non-specialized prefectures 1to1 of Session 1 (M group) = 2.5, 1to1 of Session 3 (M group) = 4.0, W = 198, p < .1). On the other

hand, we found no significant difference in the T group who experienced the 2to1 condition (1to1 of Session 1 (T group) = 1.5, 1to1 of Session 3 (T group) = 3.0, W = 227, p = n.s.). This result suggests that the M group guides gained knowledge about their non-specialties by experiencing the Mixto1 conditions.

4 Subjective Impressions of Speakers

We analyzed the overall satisfaction impressions of the guides and users on a 7-point Likert scale (7 = totally agree, 1 = totally disagree).

4.1 Approach

Guide satisfaction Our study focused on the situation where two guides talk as one. Such a situation might be confusing for guides and users. To ensure that the guides did not have any difficulty speaking under this condition, we used the following statement: “When I talked to the user, I sometimes felt it was difficult.”

In the Mixto1 condition, two guides talked as one. By sharing the dialogue context as one operator, each operator may experience the conversational content of his/her partner as if it were his/her own, creating efficient and effective learning of the other’s skill. In the Mixto1 condition, the guides may also be more aware of cooperating with the other guides and deepen their mutual trust.

We used the following three items to evaluate the guide’s satisfaction with the other guide’s co-operation: Statement (a) assessed feelings of respect for the other guide: “I felt a sense of trust in the other guide.” To evaluate the ease of co-operation with the other guide, we used statement (b): “I was able to work with the other guide.” To evaluate the impressions of learning from the other guide, we used statement (c): “I learned from the other guide’s responses.”

User satisfaction The easy-to-talk impressions felt by users under the Mixto1 and 2to1 conditions may differ. In the latter, the user distinguishes between the two guides and interacts in a multi-party manner. However, the user does not distinguish between them in the Mixto1 condition. This difference might affect the user’s speaking ease. To evaluate whether users felt it was difficult to talk, we used questionnaire item (d): “There were times when I felt it was hard to talk.”

We also evaluated whether users felt they accomplished their task with questionnaire item (e):

“Through the dialogue with the guide(s), I obtained useful information” to evaluate whether the users obtained the necessary knowledge for their travel.

4.2 Results

Guide satisfaction To analyze the impressions of the guides’ difficulty in speaking, we calculated the median of each condition. The median of each condition was lower than four points. This indicates that the guides did not perceive particular difficulty in speaking.

For their impressions of cooperating with another guide, we compared (a), the trust of another guide, under the Mixto1 and 2to1 conditions. Wilcoxon’s rank-sum test showed that the Mixto1 condition was significantly higher than the 2to1 condition ($\text{Mixto1} = 6$, $\text{2to1} = 5$, $W = 1520.5$, $p < .05$).

We also compared (b), measure of cooperation satisfaction, with the Mixto1 and 2to1 conditions. The Mixto1 condition was significantly higher than the 2to1 condition ($\text{Mixto1} = 6$, $\text{2to1} = 4$, $W = 1831$, $p < .05$).

The Mixto1 and 2to1 conditions were also compared for (c), an evaluation item of learning impression. The Mixto1 condition was significantly higher than the 2to1 condition ($\text{Mixto1} = 6$, $\text{2to1} = 5$, $W = 1445$, $p < .05$).

From the above results, the guides’ satisfaction was higher in the Mixto1 condition than in the 2to1 condition. The guides felt a sense of cooperation and trust with the other guide, adding that under the Mixto1 condition, they acquired more knowledge than under the 2to1 condition.

One possible factor that resulted in such positive impressions for the Mixto1 condition was that the guides were engaged in first-person conversations. Probably they quickly became absorbed in the conversations because the users acted like just one guide. Perhaps the guides felt that they had acquired knowledge because it was easy to regard the utterances of the other guides as their own. In the future, we must clarify which factor deepens the guides’ impressions of subjective learning by scrutinizing the dialogue content.

In addition, it may also be necessary to examine the effect of a hidden channel used by the guides because it may have had particular effects on the cooperation of the guides.

User satisfaction We did not find a significant difference in (d), the users’ perceived difficulty of speaking, in a Wilcoxon’s rank-sum test that compared the Mixto1 and 2to1 conditions ($\text{Mixto1} = 2$, $\text{2to1} = 3$, $W = 235$, $p = \text{n.s.}$). Both median values were lower than four (= neither), suggesting that they did not find it difficult to talk under either condition.

Next we analyzed (e), the impression of the users’ information collection. When the Mixto1 and 2to1 conditions were compared, no significant difference was detected ($\text{Mixto1} = 6$, $\text{2to1} = 6$, $W = 258$, $p = \text{n.s.}$). Both conditions had high scores. Perhaps the task of acquiring travel knowledge was relatively easy. Differences might surface in more difficult tasks.

In this experiment, we identified no significant differences in the user satisfaction between the Mixto1 and 2to1 conditions. However, we also found no evidence that the Mixto1 condition negatively impacted the users. Whether Mixto1 can improved the dialogue quality must be investigated with another situation in the future.

5 Conclusion

We evaluated a situation in which two operators with different skills acted as one. We collected travel guide dialogues where two operators acting as one speaker, as two speakers, and alone. We evaluated the contents under each condition as well as the satisfaction of the operators and users. The operators experienced increased satisfaction with their learning and cooperation. The users were not dissatisfied with the situation of two operators speaking as one. It is suggested that the proposed cooperation style gives operators an opportunity to engage in advanced dialogue services as well as to learn the skills of the other operators.

In the future, we must scrutinize how the operators increased their satisfaction with learning and evaluate what kind of knowledge sharing occurred between the operators. We also need to examine a combination of other kinds of skills.

Acknowledgments

This work was supported by JST-Mirai Program Grant Number JPMJMI18C6, Japan.

References

- Tahir Abbas, Vassilis-Javed Khan, and Panos Markopoulos. 2020. Coz: A crowd-powered system for social robotics. *SoftwareX*, 11:100421.
- Tsunehiro Arimoto, Yuichiro Yoshikawa, and Hiroshi Ishiguro. 2014. Nodding responses by collective proxy robots for enhancing social telepresence. In *Proceedings of the Second International Conference on Human-Agent Interaction*, pages 97–102.
- Andy Crabtree, Jacki O’Neill, Peter Tolmie, Stefania Castellani, Tommaso Colombino, and Antonietta Grasso. 2006. The practical indispensability of articulation work to immediate and remote help-giving. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 219–228.
- Mitchell Dowling and Debra Rickwood. 2013. Online counseling and therapy for mental health problems: A systematic review of individual synchronous interventions using chat. *Journal of Technology in Human Services*, 31(1):1–21.
- Kazunori Komatani, Naoyuki Kanda, Mikio Nakano, Kazuhiro Nakadai, Hiroshi Tsujino, Tetsuya Ogata, and Hiroshi G Okuno. 2009. Multi-domain spoken dialogue system with extensibility and robustness against speech recognition errors. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 9–17.
- Annica Kristoffersson, Silvia Coradeschi, and Amy Loutfi. 2013. A review of mobile robotic telepresence. *Advances in Human-Computer Interaction*, 2013.
- Walter S Lasecki, Rachel Wesley, Jeffrey Nichols, Anand Kulkarni, James F Allen, and Jeffrey P Bigham. 2013. Chorus: a crowd-powered conversational assistant. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 151–162.
- Bor-shen Lin, Hsin-min Wang, and Lin-shan Lee. 1999. A distributed architecture for cooperative spoken dialogue agents with coherent dialogue state and history. In *Proceedings of the 1999 IEEE Workshop on Automatic Speech Recognition and Understanding Workshop (ASRU)*.
- Matthew Marge, Claire Bonial, Kimberly A Pollard, Ron Artstein, Brendan Byrne, Susan G Hill, Clare Voss, and David Traum. 2016. Assessing agreement in human-robot dialogue strategies: A tale of two wizards. In *Proceedings of Intelligent Virtual Agents*, pages 484–488.
- Mikio Nakano, Shun Sato, Kazunori Komatani, Kyoko Matsuyama, Kotaro Funakoshi, and Hiroshi G Okuno. 2011. A two-stage domain selection framework for extensible multi-domain spoken dialogue systems. In *Proceedings of the SIGDIAL 2011 Conference*, pages 18–29.
- Daisuke Sakamoto, Takayuki Kanda, Tetsuo Ono, Hiroshi Ishiguro, and Norihiro Hagita. 2007. Android as a telecommunication medium with a human-like presence. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 193–200.
- Masaru Yamane, Michiya Yamamoto, and Tomio Watanabe. 2011. Development of a real-space sharing edutainment system based on communication support with make-believe play. In *SICE Annual Conference 2011*, pages 2571–2574.
- Naomi Yamashita, Hideaki Kuzuoka, Keiji Hirata, Shigemi Aoyagi, and Yoshinari Shirai. 2011. Supporting fluid tabletop collaboration across distances. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2827–2836.

Adaptive Dialog Policy Learning with Hindsight and User Modeling

Yan Cao¹ Keting Lu² Xiaoping Chen¹ Shiqi Zhang³

¹School of Computer Science, University of Science and Technology of China

²Commercialization Recommending Researching Department, Baidu Inc.

³Department of Computer Science, SUNY Binghamton

caotian@mail.ustc.edu.cn; ktlu@mail.ustc.edu.cn;

xpchen@ustc.edu.cn; zhangs@binghamton.edu

Abstract

Reinforcement learning methods have been used to compute dialog policies from language-based interaction experiences. Efficiency is of particular importance in dialog policy learning, because of the considerable cost of interacting with people, and the very poor user experience from low-quality conversations. Aiming at improving the efficiency of dialog policy learning, we develop algorithm LHUA (Learning with Hindsight, User modeling, and Adaptation) that, for the first time, enables dialog agents to adaptively learn with hindsight from both simulated and real users. Simulation and hindsight provide the dialog agent with more experience and more (positive) reinforcements respectively. Experimental results suggest that, in success rate and policy quality, LHUA outperforms competitive baselines from the literature, as well as its no-simulation, no-adaptation, and no-hindsight counterparts.

1 Introduction

Dialog systems have enabled intelligent agents to communicate with people using natural language. For instance, virtual assistants, such as Siri, Alexa, and Cortana, have been increasingly popular in daily life. We are particularly interested in goal-oriented dialog systems, where the task is to efficiently and accurately exchange information with people, and the main challenge is on the ubiquitous ambiguity in natural language processing (spoken or text-based). Goal-oriented dialog systems typically include components for language understanding, dialog management, and language synthesis, while sometimes the components can be constructed altogether, resulting in end-to-end dialog systems (Bordes et al., 2016; Williams and Zweig, 2016; Wen et al., 2017; Young et al., 2018;

Yang et al., 2017). In this paper, we focus on the problem of policy learning for dialog management.

Reinforcement learning (RL) algorithms aim at learning action policies from trial-and-error experiences (Sutton and Barto, 2018), and have been used for learning dialog policies (Young et al., 2013; Levin et al., 1997). Deep RL methods (e.g. (Mnih et al., 2013)) have been developed for dialog policy learning in dialog domains with large state spaces (Su et al., 2016a; Fatemi et al., 2016; Serban et al., 2017). While it is always desirable for RL agents to learn from the experiences of interacting with the real world, such interactions can be expensive, risky, or both in practice. Back to the context of dialog systems, despite all the advances in RL (deep or not), dialog policy learning remains a challenge. For instance, interacting with people using natural language is very costly, and low-quality dialog policies produce very poor user experience, which is particularly common in early learning phases. As a result, it is critical to develop sample-efficient RL methods for learning high-quality dialog policies with limited conversational experiences.

In this paper, we develop an algorithm called LHUA (Learning with Hindsight, User modeling, and Adaptation) for sample-efficient dialog policy learning. LHUA, for the first time, enables a dialog agent to simultaneously learn from real, simulated, and hindsight experiences, which identifies the key contribution of this research. Simulated experience is generated using learned user models, and hindsight experience (of successful dialog samples) is generated by manipulating dialog segments and goals of the (potentially many) unsuccessful samples. Dialog experience from simulation and hindsight respectively provide more dialog samples and more positive feedback for dialog policy learning. To further improve the sample efficiency, we develop a meta-agent for LHUA that adaptively

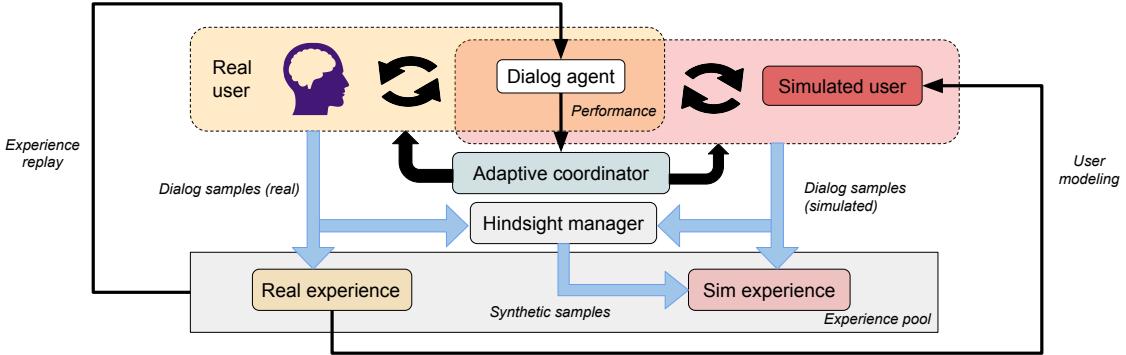


Figure 1: An overview of LHUA. A *dialog agent* interacts with both real and simulated users while learning a dialog policy from this interaction experience. A *simulated user* is modeled using real dialog samples, and interacting with this simulated user provides the dialog agent with simulated dialog samples. An *adaptive coordinator* learns from the dialog agent’s recent performance to adaptively assign one user (real or simulated) for the dialog agent to interact with. A *hindsight manager* manipulates both real and simulated dialog samples (of mixed qualities) to “synthesize” successful dialog samples.

learns to switch between real and simulated users in the dialog-based interactions, which identifies the second contribution of this research. An overview of LHUA is shown in Figure 1.

Experiments were conducted using a realistic movie-ticket booking platform (Li et al., 2017). LHUA has been compared with state-of-the-art methods (Peng et al., 2018; Lu et al., 2019; Su et al., 2018) in dialog policy learning tasks. Results suggest that ablations of LHUA produce comparable (or better) performances in comparison to competitive baselines in success rate, and LHUA as a whole performed the best.

2 Related Work

In this section, we summarize three different ways of improving the efficiency of dialog policy learning (namely user modeling, hindsight experience replay, and reward shaping), and qualitatively compare them with our methods.

Researchers have developed “two-step” algorithms that first build user models through supervised learning with real conversational data, and then learn dialog policies by interacting with the simulated users (Schatzmann et al., 2007; Li et al., 2016b). In those methods, user modeling must be conducted offline before the start of dialog policy learning. As a result, the learned policies are potentially biased toward the historical conversational data. Toward online methods for dialog policy learning, researchers have developed algorithms for simultaneously constructing models of real users, and learning from the simulated interaction experience with user models (Asri et al., 2016; Su et al., 2016b; Lipton et al., 2016; Zhao and Eskenazi,

2016; Williams et al., 2017; Dhingra et al., 2017; Li et al., 2017; Liu and Lane, 2017; Peng et al., 2017; Wu et al., 2019; Li et al., 2016a). Those methods enable agents to simultaneously build and leverage user models in dialog policy learning. However, the problem of learning high-quality user models by itself can be challenging. Our algorithms support user modeling, while further enabling agents to adaptively learn from both hindsight and real conversations.

In comparison to many other RL applications, goal-oriented dialog systems have very sparse feedback from the “real world” (human users), where one frequently cannot tell dialogs being successful or not until reaching the very end. Positive feedback is even rarer, when dialog policies are of poor qualities. Hindsight experience replay (HER) (Andrychowicz et al., 2017) methods have been developed to convert unsuccessful trials into successful ones through goal manipulation. The “policy learning with hindsight” idea has been applied to various domains, including dialog (Lu et al., 2019). Our methods support the capability of learning from hindsight experience, while further enabling user modeling and learning from simulated users.

Within the dialog policy learning context, reward shaping is another way of providing the dialog agents with extra feedback, where a dense reward function can be manually designed (Su et al., 2015), or learned (Su et al., 2016b). Researchers also developed efficient exploration strategies to speed up the policy learning process of dialog agents, e.g., (Pietquin et al., 2011; Lagoudakis and Parr, 2003). Those methods are orthogonal to ours, and

can potentially be combined to further improve the dialog learning efficiency. In comparison to all methods mentioned in this section, LHUA is the first that enables dialog policy learning from real, simulated, and hindsight experiences simultaneously, and its performance is further enhanced through a meta-policy for switching between interactions with real and simulated users.

3 Background

In this section, we briefly introduce the two building blocks of this research, namely Markov decision process (MDP)-based dialog management, and Deep Q-Network (DQN).

3.1 MDP-based Dialog Management

Markov Decision Processes (MDPs) can be specified as a tuple $\langle \mathcal{S}, \mathcal{A}, T, \mathcal{R}, s_0 \rangle$, where \mathcal{S} is the state set, \mathcal{A} is the action set, T is the transition function, \mathcal{R} is the reward function, and s_0 is the initial state. In MDP-based dialog managers, dialog control can be modeled using MDPs for selecting language actions. $s \in \mathcal{S}$ represents the current dialog state including the agent's last action, the user's current action, the distribution of each slot, and other domain variables as needed. $a \in \mathcal{A}$ represents the agent's response. The reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbf{R}$ gives the agent a big bonus in successful dialogs, a big penalty in failures, and a small cost in each turn.

Solving an MDP-based dialog management problem produces π , a dialog policy. A dialog policy maps a dialog state to an action, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, toward maximizing the discounted, accumulative reward in dialogs, i.e., $R_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$, where $\gamma \in [0, 1]$ is a discount factor that specifies how much the agent favors future rewards.

3.2 Deep Q-Network

Deep Q-Network (DQN) (Mnih et al., 2015) is a model-free RL algorithm. The approximation of the optimal Q-function, $Q^* = Q(s, a; \theta)$, is used by a neural network, where a is an action executed at state s , and θ is a set of parameters. Its policy is defined either in a greedy way: $\pi_Q(s) = \text{argmax}_{a \in \mathcal{A}} Q(s, a; \theta)$ or being ϵ -greedy, i.e., the agent takes a random action in probability ϵ and action $\pi_Q(s)$ otherwise. The loss function for minimization in DQN is usually defined using TD-error:

$$\mathcal{L} = \mathbf{E}_{s,a,r,s'}[(Q(s, a; \theta) - y)^2], \quad (1)$$

where $y = r + \gamma \max_{a' \in \mathcal{A}} Q(s', a'; \theta)$.

To alleviate the problem of unstable or non-convergence of Q values, two techniques are widely used. One is called *target network* whose parameters are updated by θ once every many iterations in the training phase. The other technique is *experience replay*, where an experience pool ε stores samples, each in the form of (s_t, a_t, r_t, s_{t+1}) . It randomly selects small batches of samples from ε each time during training. Experience replay can reduce the correlation between samples, and increases the data efficiency.

4 Algorithms

In this section, we first introduce Learning with Hindsight, and User modeling (**LHU**), and then present LHU with Adaptation (**LHUA**), where algorithms LHU and LHUA point to the main contribution of this research.

LHU, for the first time, enables a dialog agent to learn dialog policies from **three dialog sources**, namely real users, simulated users, and hindsight dialog experience. More specifically, a real user refers to the human who converses with the dialog agent, and a simulated user refers to a learned user model that captures real users' interactive behaviors with our dialog agent. In this way, a simulated user is used for generating "human-like" dialog experience for speeding up the process of dialog policy learning. The last dialog source of "hindsight dialog experience" is used for creating many *successful* dialog samples using both successful and unsuccessful dialog samples, where the source samples are from both real and simulated users. Different from "simulated users" that generate dialog samples of mixed qualities, hindsight experience produces only successful (though not real) dialog samples, which is particularly useful for dialog policy learning at the early phase due to the very few successful samples.

Among the three dialog sources, hindsight experience is "always on", and synthesizes dialog samples throughout the learning process. The "real" and "simulated" dialog sources bring in the selection problem: *At a particular time, from which source should the agent obtain dialog experience for policy learning?* The "adaptation" capability of LHUA aims at enabling the dialog agent to learn to, before starting a dialog, select which user (real or simulated) to interact with.

4.1 Learning with Hindsight, and User Modeling

In this subsection, we focus on two components of LHUA, including user modeling, and hindsight management, which together form LHU, an ablation algorithm of LHUA. The two components' shared goal is to generate additional dialog experience (simulated and hindsight experiences respectively) to speed up dialog policy learning.

Dialog (Sub)Goal and Segmentation Goal-oriented dialog agents help users accomplish their goals via language-based multi-turn communications. Goal G includes a set of constraints C and a set of requests R , where $G = (C, R)$. Consider a service request “*I’d like to purchase one ticket of Titanic for this evening. Which theater is available?*” In this example, the goal is of the form:

$$G = (C = [ticket = one, time = eve, movie = titanic], R = [theater = ?])$$

We define G' as a subgoal of $G = (C, R)$: $G' = (C', R')$, where $C' \subseteq C$, $R' \subseteq R$, and G' cannot be empty. Continuing the “titanic” example, one of its subgoals is

$$G' = (C' = [ticket = one, movie = titanic], R' = \emptyset).$$

Given an intact dialog D , we say D_{seg} is a segment of D , if D_{seg} includes a consecutive sequence of turns of D . With the concepts of dialog segment and subgoal, we introduce two segment sets (head and tail), which are later used in *hindsight manager*. A head segment set Ω consists of dialog segments D_{head} that include the early turns in the intact dialog with the corresponding completed subgoal G' .

$$\Omega = \{(D_{head}, G')\} \quad (2)$$

We use function *HeadSegGen* to collect a head segment set Ω during dialog interactions. *HeadSegGen* receives a dialog segment D_{seg} , and a goal G , then checks all subgoals of G , and finally outputs pairs (D_{seg}, G') where D_{seg} accomplishes subgoal G' of G .

A tail segment set Γ consists of dialog segments D_{tail} that include the late turns in the intact dialog with the corresponding completed subgoal G' .

$$\Gamma = \{(D_{tail}, G')\} \quad (3)$$

Function *TailSegGen* is implemented to generate tail segments after interactions terminate. It receives a dialog D , a goal G and a corresponding head segment Ω . If the dialog D accomplishes the goal G , for each pair (D_{head}, G') from the head segment set Ω , *TailSegGen* outputs a corresponding pair $(D \ominus D_{head}, G')$, where $D_1 \ominus D_2$ produces a dialog segment by removing D_2 from D_1 .

Hindsight Manager Given head and tail segment sets (Ω and Γ), the *hindsight manager* is used for stitching two tuples, (D_{head}, G'_{head}) and (D_{tail}, G'_{tail}) , respectively to “synthesize” successful dialog samples. There are two conditions for synthesization:

1. The two subgoals from head and tail segments are identical, $G'_{head} == G'_{tail}$, and
2. The last state of D_{head} , s_{last} , and the first state of D_{tail} , s'_{first} , are of sufficient similarity.

We use *KL Divergence* to measure the similarity between two states:

$$D_{KL}(s_{last} || s'_{first}) \leq \delta \quad (4)$$

where $\delta \in R$ is a threshold parameter. We implement a function to synthesize successful dialog samples as hindsight experience for dialog policy learning, as follows:

$$D_{hind} \leftarrow HindMan(\delta, \Omega, \Gamma) \quad (5)$$

HindMan takes a threshold δ , a head segment set Ω , and a tail segment set Γ . It generates successful dialog samples D_{hind} that satisfy the above two conditions of synthesization.

Dialog with Simulated Users In dialog policy learning, dialog agents can learn from interactions with real users, where the generated real experience is stored in reply buffer B^R . To provide more experience, we develop a simulated user for generating simulated dialog experience to further speed up the learning of dialog policies.

The simulated user is of the form:

$$s', r \leftarrow M(s, a; \theta_M)$$

where, $M(s, a; \theta_M)$ takes the current dialog state s and the last dialog agent action a as input, and generates the next dialog state s' , and reward r . M is implemented by a Multi-Layer Perceptron (MLP) parameterized by θ_M , and refined via stochastic

Algorithm 1 Algorithm LHU

Input: K , the times of interactions with the simulated user; δ , KL-divergence threshold

Output: the success rate $SR^{D_{lg}}$, and average rewards $R^{D_{lg}}$ of agent $^{D_{lg}}$; $Q(\cdot)$ for agent $^{D_{lg}}$

```

1: Initialize  $Q(s, a; \theta_Q)$  of agent $^{D_{lg}}$  and  $M(s, a; \theta_M)$  of
   the simulated user via pre-training on human conversational data
2: Initialize experience replay buffers  $B^R$  and  $B^S$  for the
   interaction of agent $^{D_{lg}}$  with real and simulated users
3: Initialize head and tail dialog segment sets:
    $\Omega \leftarrow \emptyset$ , and  $\Gamma \leftarrow \emptyset$ 
4: Collect initial state,  $s$ , by interacting with a real user
   following goal  $G^{Real}$ 
5: Initialize  $D^{Real} \leftarrow \emptyset$  for storing dialog turns (real)
6: while  $s \notin \text{term}$  do // Start a dialog with real user
7:   Select  $a \leftarrow \text{argmax}_{a'} Q(s, a'; \theta_Q)$ , and execute  $a$ 
8:   Collect next state  $s'$ , and reward  $r$ 
9:   Add dialog turn  $d = (s, a, r, s')$  to  $B^R$  and  $D^{Real}$ 
10:   $\Omega \leftarrow \Omega \cup \text{HeadSegGen}(D^{Real}, G^{Real})$ 
11:   $s \leftarrow s'$ 
12: end while
13:  $\Gamma \leftarrow \Gamma \cup \text{TailSegGen}(D^{Real}, G^{Real}, \Omega)$ 
14: for  $k = 1 : K$  do //  $K$  interactions with simulated user
15:   Sample goal  $G^{Sim}$ , and initial state  $s$ 
16:   Initialize  $D^{Sim} \leftarrow \emptyset$  for storing dialog turns (sim)
17:   while  $s \notin \text{term}$  do // The  $k^{th}$  dialog with sim user
18:      $a \leftarrow \text{argmax}_{a'} Q(s, a'; \theta_Q)$ , and execute  $a$ 
19:     Collect next state  $s'$ , and reward  $r$  from
         $M(s, a; \theta_M)$ 
20:     Add dialog turn  $d = (s, a, r, s')$  to  $B^S$  and  $D^{Sim}$ 
21:      $\Omega \leftarrow \Omega \cup \text{HeadSegGen}(D^{Sim}, G^{Sim})$ 
22:      $s \leftarrow s'$ 
23:   end while
24:    $\Gamma \leftarrow \Gamma \cup \text{TailSegGen}(D^{Sim}, G^{Sim}, \Omega)$ 
25: end for
26: Synthesize hindsight experience, and store it in  $B^S$ :
    $D_{hind} \leftarrow \text{HindMan}(\delta, \Gamma, \Omega)$  // Hindsight Manipulation
27: Calculate the success rate  $SR^{D_{lg}}$  and average rewards
    $R^{D_{lg}}$  of total interactions
28: Randomly sample a minibatch from both  $B^R$  and  $B^S$ ,
   and update agent $^{D_{lg}}$  via DQN // agent $^{D_{lg}}$  training
29: Randomly sample a minibatch from  $B^R$ , and update simulated
   user via SGD // User modeling
30: return  $SR^{D_{lg}}, R^{D_{lg}}, Q(\cdot)$ 

```

gradient descent (SGD) using real experience in B^R to improve the quality of simulated experience.

Simulated experience generated from interactions between the dialog agent and the simulated user is stored in the simulated replay buffer B^S , which is also manipulated by the *hindsight manager* to synthesize hindsight experience.

The LHU Algorithm Algorithm 1 presents the learning process, where our dialog agent interacts with a real user for one dialog, and a simulated user for k dialogs. In addition to parameter k , there is a *KL-divergence* threshold δ as a part of the input. We refer to this algorithm using LHU(k).

Algorithm 1 starts with an initialization of the

dialog agent’s real and simulated experience replay buffers (B^R and B^S respectively), the model of the simulated user, $M(\theta_M)$, and two segment sets for *hindsight manager* (Ω and Γ respectively). In the first *while* loop (starting in Line 6), the dialog agent interacts with a real user and stores the real experience in B^R . Then, k dialogs with the simulated user are conducted in the *for* loop, where simulated experience is stored in B^S . During interactions with both real and simulated users, head and tail segment sets are simultaneously collected (Lines 21 and 24). After all dialog interactions end, the *hindsight manager* is used to synthesize successful dialog samples and store them in B^S . Finally, the dialog agent is trained on B^R and B^S , and the simulated user is trained on B^R .

The output of Algorithm 1 is used in the next section, where we introduce how to further enable the dialog agent to learn a meta-policy for adaptively determining which user (real or simulated) to interact with.

4.2 LHU with Adaptation (LHUA)

Adaptively determining which user (real or simulated) the LHU agent should interact with can further speed up the dialog policy learning process. The idea behind it is that, if a simulated user can generate high-quality, realistic dialog experience, interactions with the simulated user should be encouraged. To enable this adaptive “switching” behaviors, we develop an *adaptive coordinator* that learns a meta-policy for selecting between real and simulated users for collecting interaction experience. We learn this adaptive coordinator using reinforcement learning, producing the LHUA algorithm, which is described next.

State In each turn of interaction with the LHU agent, *adaptive coordinator* updates the adaptation state s^A using the equation below:

$$s_i^A = \begin{cases} [0, 0, 0, 0] & i = 0 \\ [SR_i, R_i, SR_i - SR_{i-1}, R_i - R_{i-1}] & i > 0 \end{cases} \quad (6)$$

where SR_i and R_i are respectively average success rate and rewards from LHU agent’s training performance at i^{th} episode. In practice, R is normalized to have values between 0 and 1, same as SR . This form of adaptation state provides accessible information on different training phrases to represent LHU agent’s current performance.

Action Based on the state s^A , *adaptive coordinator* chooses action k to determine, after each dialog

Algorithm 2 LHU with Adaptation (LHUA)

Input: H , the max length of adaptation episode; δ , KL-divergence threshold; N , training times

Output: Π , the dialog policy;

```

1: Initialize  $A(s^A, k; \theta_A)$  of  $agent^{Adp}$ , and replay buffer
    $B^A$  as empty
2: for  $i = 1 : N$  do
3:   Initialize adaptation state  $s^A$  using Eqn. 6
4:   Initialize turn counter  $h: h = 0$ 
5:   while  $h \leq H$  do
6:     Select action  $k: k \leftarrow argmax_{k'} A(s^A, k'; \theta_A)$ 
7:     Execute action  $k$ :
         $SR^{Dlg}, R^{Dlg}, Q(\cdot) \leftarrow LHU^1(k, \delta)$ 
8:     Collect reward  $r^A$  via Eqn. 7, and next adaptation
       state  $\hat{s}^A$  using Eqn. 6
9:      $B^A \leftarrow B^A \cup (s^A, k, r^A, \hat{s}^A)$ ,  $s^A \leftarrow \hat{s}^A$ , and  $h \leftarrow$ 
        $h + 1$ 
10:    end while
11:   Sample a minibatch from  $B^A$ , and update  $\theta_A$  via DQN
12: end for
13: for all  $s \in \mathcal{S}$ :  $\Pi(s) \leftarrow argmax_{a'} Q(s, a'; \theta_Q)$ 
14: return  $\Pi(\cdot)$ 
```

with the real user, how many dialogs should be conducted with the simulated user. The value of action k ranges from 1 to K .

Reward *Adaptive coordinator* receives immediate rewards after executing an action k (i.e. LHU(k)) each time. We use success rate increment of LHU agent to design the reward function, as shown below:

$$r_i^A = \frac{SR_i - SR_{i-1}}{SR_i} \cdot \frac{k_i}{L_i} \quad (0 < i \leq H) \quad (7)$$

where k_i is the i^{th} action chosen by *adaptive coordinator*, and L_i means the total number of times of interactions with both real and simulated users, i.e. $L_i = k_i + 1$. Reward is continuously harvested, until the H^{th} turn.

Due to the continuous state space, the approximated value function of *adaptive coordinator* is implemented using a two-layer fully connected neural network, $A(s^A, k; \theta_A)$, parameterized by θ_A . Interactions between the *adaptive coordinator* and the LHU agent start with an initial state. In each turn, the *adaptive coordinator* obtains the state s^A using Eqn. 6, and selects the action k via ϵ -greedy policy to execute. Then, the current training performance of LHU agent is used for acquiring the reward r^A using Eqn. 7, and updating the next state \hat{s}^A . Finally, the experience (s^A, k, r^A, \hat{s}^A) is stored for meta-policy learning. We improve the value function by adjusting θ_A to minimize the mean-squared loss function.

The LHUA Algorithm Algorithm 2 presents the dialog policy learning process, where our dialog agent adaptively learns from both simulated and real users. In addition to parameter δ for KL-divergence threshold, there is parameter H representing the length of one episode for adaptive coordinator as a part of the input.

Algorithm 2 starts with an initialization of replay buffer B^A for adaptive coordinator, and the value function $A(s^A, k; \theta_A)$. Before the start of each episode, a turn counter h is initialized as zero for turn counting. Adaptive coordinator interacts with LHU agent for H turns while collecting and saving experience in B^A . At the end of each adaptation episode, we use DQN to update θ_A .

LHUA enables the dialog agent to simultaneously learn from the dialogs with both real and simulated users. At the same time, *hindsight manager* manipulates both real and simulated dialog samples to synthesize more successful dialog samples. The *adaptive coordinator* is learned at runtime for adaptively switching between real and simulated users in the dialog policy learning process to further improve the sample efficiency. So far, LHUA enables dialog agents to adaptively learn with hindsight from both simulated and real users.

5 Experiment

Experiments have been conducted in a dialog simulation platform, called TC-bot (Li et al., 2016b, 2017).¹ TC-bot provides a realistic simulation platform for goal-oriented dialog system research. We use its *movie-ticket booking* domain that consists of 29 slots of two types, where one type is on *search constraints* (e.g., number of people, and date), and the other is on *system-informable* properties that are needed for database queries (e.g., critic rating, and start time). The dialog agent has 11 dialog actions, representing the system intent (e.g., confirm question, confirm answer, and thanks).

A dialog is considered successful only if movie tickets are booked successfully, and the provided information satisfies all the user’s constraints. By the end of a dialog, the agent receives a bonus (positive reward) of $2 * L$ if successful, or a penalty (negative reward) of $-L$ for failure, where L is the maximum number of turns allowed in each dialog. We set $L = 40$ in our experiments. The

¹To avoid possible confusions, we use “real user” to refer to the user directly provided by TC-bot, and use “simulated user” to refer to the user model learned by our dialog agents.

agent receives a unit cost in each dialog turn to encourage shorter conversations.

Implementation Details In line with existing research (Peng et al., 2018), all dialog agents are implemented using Deep Q-Network (DQN). The DQN includes one hidden layer with 80 hidden nodes and ReLU activation, and its output layer of 11 units corresponding to 11 dialog actions. We set the discount factor $\gamma = 0.95$. The techniques of target network and experience replay are applied. Both B^R and B^S share the buffer size of 5000, and we use uniform sampling in experience replay. The target value function is updated at the end of each epoch. In each epoch, $Q(\cdot)$ and $M(\cdot)$ are refined using one-step 16-tuple-minibatch update. We then pre-filled the experience replay buffer with 100 dialogs before training. The simulated experience buffer B^S is initialized as empty. Neural network parameters are randomly initialized, and optimized using RMSProp (Hinton et al., 2012).

The simulated user model, $M(\cdot)$, is a multi-task neural network (Liu et al., 2015), and contains two shared hidden layers and three task-specific hidden layers, where each layer has 80 nodes. Stitching threshold of *hindsight manager* δ is set 0.2. The policy network of *adaptive coordinator* is a single-layer neural network of size 64. Parameters k and H are described in Algorithm 2, and have the value of $k = 20$ and $H = 8$.

LHUA and Three Baselines Our key hypothesis is that adaptively learning from real, simulated, and hindsight experiences at the same time performs better than baselines from the literature. To evaluate this hypothesis, we have selected three competitive baselines for goal-oriented dialog policy learning, including DDQ (Su et al., 2018), D3Q (Wu et al., 2019), and S-HER (Lu et al., 2019). In implementing the DDQ agent, the ratio of interaction experiences between simulated and real users is ten, which is consistent with the original implementation (Su et al., 2018). The differences between LHUA and the baseline methods are qualitatively discussed in Section 2.

It is necessary to explain how the curves are generated in the figures to be reported. For each of the four methods (LHUA and three baselines), we have conducted five “runs”, where each run includes 250 episodes. In each run, after every single episode for learning, we let the dialog agent interact with the real user for 50 dialogs, only for

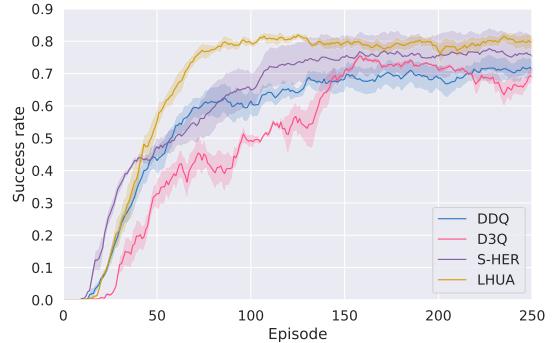


Figure 2: The performances of LHUA (ours), and three baseline methods, including DDQ (Su et al., 2018), D3Q (Wu et al., 2019), and S-HER (Lu et al., 2019). We see that, except for the very early phase (first 50 episodes), LHUA outperformed all baselines.

evaluation. We then compute the success rate over the 50 dialogs. Each data point in the figure is an average over the five success rates collected from the five runs of each method.

Figure 2 presents the key results of this research on the quantitative comparisons between LHUA and the three baselines. We can see that, except for the very early learning phase, LHUA performed consistently better than the three baseline methods. In particular, LHUA reached the success rate of 0.75 after about 70 episodes, whereas none of the baselines were able to achieve comparable performance within 150 episodes. The gap between LHUA and S-HER in early phase is due to the fact that LHUA needs to learn a user model, which requires extra interaction in early phase. Once the user model is of reasonable quality, LHUA is able to learn from the interaction experience with simulated users, and soon (after 45 episodes) LHUA outperformed S-HER.

LHUA and Its Ablations Results reported in Figure 2 have shown the advantage of LHUA over the three baseline methods. However, it is still unclear how much each component of LHUA contributes to its performance. We removed components from LHUA, and generated four different ablations of LHUA, including DQN, DDQ (LU, or Learning with User modeling), S-HER (LH, or Learning with Hindsight), LHU, and LHUA.

Figure 3 shows the ablation experiment’s results. From the results, we see that LHUA performed much better than no-hindsight (LU), and no-user-modeling (S-HER, or LH) ablations. When both “hindsight” and “user modeling” are activated, there is LHUA’s ablation of LHU, which performed bet-

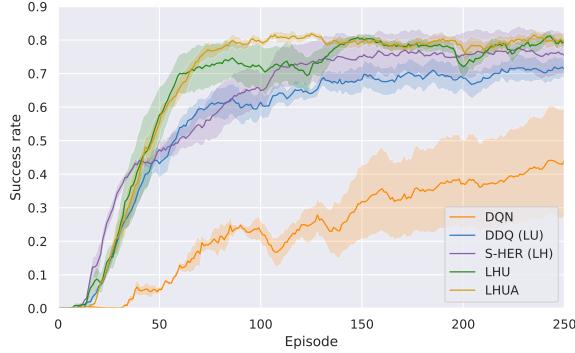


Figure 3: Comparisons between LHUA and its ablations: DQN (no hindsight manager, no user modeling, and no adaptive coordinator), DDQ (no hindsight manager, and no adaptive coordinator), S-HER (no user modeling, and no adaptive coordinator), and LHU (no adaptive coordinator). A complete LHUA includes all the components, including DQN (for naive dialog policy learning), hindsight manager, user modeling, and adaptive coordinator.

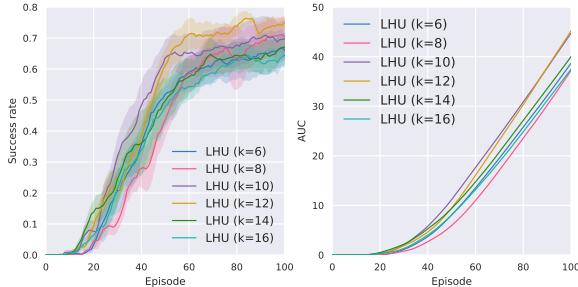


Figure 4: Success rate on the left, and Area under Curve (AUC) on the right, where we implemented six different versions of LHU with different k values, ranging from 6 to 16 at an interval of 2.

ter than all the other ablations. LHU still cannot generate comparable performance, c.f., LHUA, which justified the necessity of the adaptive coordinator. It should be noted that performances of two of the ablations have been reported in Figure 2. We intentionally include their results in Figure 3 for the completeness of comparisons.

Adaptive Coordinator Learning Results reported in Figure 3 have shown the necessity of our adaptive coordinator in LHUA. In this experiment, we look into the learning process of the adaptive coordinator. More specifically, we are interested in how the value of k is selected (see Algorithm 2). We have implemented LHU with six different values of k , and their performances are reported in Figure 4, where the left subfigure is on success rate, and the right is on Area under Curve (AUC).

The AUC metric has been used for the evaluation of learning speed (Taylor and Stone, 2009; Stadie et al., 2015). We see that, in early learning phase (within 100 episodes), the k value of 10 produced the best performance overall, though the performance is comparable to that with $k = 12$ to some level.

Figure 5 reports

the selection of k values by our adaptive coordinator. Each bar corresponds to an average over the k values of 25 episodes. We see that the value of k was suggested to

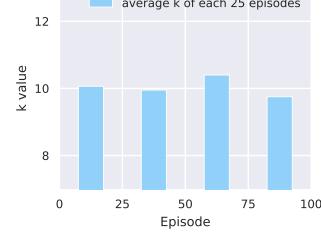


Figure 5: The k values selected by the *adaptive coordinator* of our LHUA agent

be around 10 within the first 100 episodes, which is consistent to our observation from the results of Figure 4. The consistency further justified our adaptive coordinator’s capability of learning the interaction strategy in switching between real and simulated users.

6 Conclusions and Future Work

In this work, we develop an algorithm called LHUA (Learning with Hindsight, User modeling, and Adaptation) for sample-efficient dialog policy learning. LHUA enables dialog agents to adaptively learn with hindsight from both simulated and real users. Simulation and hindsight provide the dialog agent with more experience and more (positive) reinforcements respectively. Experimental results suggest that LHUA outperforms competitive baselines (including success rate and learning speed) from the literature, including its no-simulation, no-adaptation, and no-hindsight counterparts. This is the first work that enables a dialog agent to adaptively learn from real, simulated, and hindsight experiences all at the same time.

In the future, we plan to evaluate our algorithm using other dialog simulation platform, e.g., PyDial (Ultes et al., 2017). Another direction is to combine other efficient exploration strategies, including learning directed exploration policies with different trade-offs between exploration and exploitation (Puigdomènech Badia et al., 2020). We will also focus on generating more synthetic dialog experience of different quality (Lu et al., 2020), to further improve the dialog learning efficiency.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under grant number U1613216. Zhang is supported in part by grants from the National Science Foundation (IIS-1925044), Ford Motor Company (URP Award), OPPO (Faculty Research Award), and SUNY Research Foundation.

References

- Marcin Andrychowicz, Filip Wolski, Alex Ray, et al. 2017. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 5048–5058.
- Layla El Asri, Jing He, and Kaheer Suleman. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. *Interspeech 2016*.
- Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog.
- Bhuwan Dhingra, Lihong Li, Xiuju Li, et al. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.
- Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. 2016. Policy networks with two-stage training for dialogue systems. *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.
- Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*.
- Michail G Lagoudakis and Ronald Parr. 2003. Least-squares policy iteration. *Journal of machine learning research*.
- Esther Levin, Roberto Pieraccini, and Wieland Eckert. 1997. Learning dialogue strategies within the markov decision process framework. In *IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*.
- Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Xiuju Li, Yun-Nung Chen, Lihong Li, et al. 2017. End-to-end task-completion neural dialogue systems. In *International Joint Conference on Natural Language Processing*.
- Xiuju Li, Zachary C. Lipton, Bhuwan Dhingra, et al. 2016b. A user simulator for task-completion dialogues.
- Zachary C. Lipton, Jianfeng Gao, Lihong Li, Xiuju Li, Faisal Ahmed, and Li Deng. 2016. Efficient exploration for dialogue policy learning with bbq networks & replay buffer spiking. Technical report.
- Bing Liu and Ian Lane. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE.
- Xiaodong Liu, Jianfeng Gao, Xiaodong He, et al. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Keting Lu, Shiqi Zhang, and Xiaoping Chen. 2019. Goal-oriented dialogue policy learning from failures. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Keting Lu, Shiqi Zhang, and Xiaoping Chen. 2020. Autoeg: Automated experience grafting for off-policy deep reinforcement learning. *arXiv preprint arXiv:2004.10698*.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. 2015. Human-level control through deep reinforcement learning. *Nature*.
- Baolin Peng, Xiuju Li, Jianfeng Gao, et al. 2018. Deep Dyna-Q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.
- Baolin Peng, Xiuju Li, Lihong Li, et al. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, and Hervé Frezza-Buet. 2011. Sample-efficient batch reinforcement learning for dialogue management optimization. *ACM Transactions on Speech and Language Processing (TSLP)*.
- Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, et al. 2020. Never give up: Learning directed exploration strategies. *arXiv*.
- Jost Schatzmann, Blaise Thomson, Karl Weilhammer, et al. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *The Conference of the North American Chapter of the Association for Computational Linguistics*.

- Iulian V. Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, et al. 2017. **A deep reinforcement learning chatbot.**
- Bradly C Stadie, Sergey Levine, and Pieter Abbeel. 2015. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*.
- Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, et al. 2016a. **Continuously learning neural dialogue management.**
- Pei-Hao Su, Milica Gašić, Nikola Mrkšić, et al. 2016b. **On-line active reward learning for policy optimisation in spoken dialogue systems.** In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.
- Pei-Hao Su, David Vandyke, Milica Gasic, Nikola Mrksic, Tsung-Hsien Wen, and Steve Young. 2015. Reward shaping with recurrent neural networks for speeding up on-line policy learning in spoken dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.
- Shang-Yu Su, Xiujun Li, Jianfeng Gao, et al. 2018. **Discriminative deep dyna-q: Robust planning for dialogue policy learning.** *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Richard S Sutton and Andrew G Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.
- Matthew E Taylor and Peter Stone. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685.
- Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, et al. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, et al. 2017. **A network-based end-to-end trainable task-oriented dialogue system.** *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.
- Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. **Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning.** *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.
- Jason D Williams and Geoffrey Zweig. 2016. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*.
- Yuxin Wu, Xiujun Li, Jingjing Liu, et al. 2019. Switch-based active deep dyna-q: Efficient adaptive planning for task-completion dialogue policy learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- X. Yang, Y. Chen, D. Hakkani-Tür, P. Crook, et al. 2017. End-to-end joint learning of natural language understanding and dialogue manager. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*.
- Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with common-sense knowledge. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Tiancheng Zhao and Maxine Eskenazi. 2016. **Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning.** In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Dialogue Policies for Learning Board Games through Multimodal Communication

Maryam Zare, Ali Ayub, Aishan Liu, Sweekar Sudhakara, Alan Wagner, and Rebecca Passonneau

Pennsylvania State University, University Park

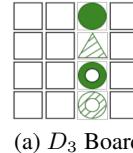
{muz50, aja5755, az153, sks6492, azw78, rjp49}@psu.edu

Abstract

This paper presents MDP policy learning for agents to learn strategic behavior—how to play board games—during multimodal dialogues. Policies are trained offline in simulation, with dialogues carried out in a formal language. The agent has a temporary belief state for the dialogue, and a persistent knowledge store represented as an extensive-form game tree. How well the agent learns a new game from a dialogue with a simulated partner is evaluated by how well it plays the game, given its dialogue-final knowledge state. During policy training, we control for the simulated dialogue partner’s level of informativeness in responding to questions. The agent learns best when its trained policy matches the current dialogue partner’s informativeness. We also present a novel data collection for training natural language modules. Human subjects who engaged in dialogues with a baseline system rated the system’s language skills as above average. Further, results confirm that human dialogue partners also vary in their informativeness.

1 Introduction

Agents that can learn by communicating with human have many potential benefits for human-agent interaction in real world situations, including making it easier for ordinary people to integrate agents into their daily activities. Agents that can communicate to learn games could help us understand how to design agents that can communicate to learn how to make strategic decisions, meaning to pursue a goal when the state of the world changes. Games are a useful testbed, given our reliance on extensive-form game trees, which supports generalization across games. Games model a space of interactions from very simple two-player settings (e.g., tic-tac-toe) to highly complex multi-party interactions (e.g., bridge). Our agent learns Markov Decision Process (MDP) dialogue policies to learn *in-a-row* board games by asking questions of dia-



Question

- **MRL:** *ContextSwitch(D_3), Confirm(Color)*
NL: Let’s step back to the third example for a moment. Does this count because all four pieces are green?

Answer

- **MRL:** *Unknown()*
NL: I’m not sure about that!

Figure 1: A turn exchange from our dialogue data collection. The original dialogues were between an agent learning Quarto and a simulated dialogue partner, using a meaning representation language (MRL). For 960 dialogues, experts translated each MRL into English. (Best viewed in color.)

logue partners, with policy differences that derive from differences in game complexity, and differences in dialogue partners.

Our MDP policies are trained offline through simulation. Agent dialogues are carried out in a general meaning representation language (MRL) we developed for communicating about games. The agent can request a visual demonstration, or can formulate context-specific verbal questions, including “*yes/no*” questions, as illustrated in Figure 1, and open-ended “*wh-*” questions. Because different humans can have different knowledge, or different dispositions for how much information to give when answering questions, we investigated the impact of policy learning that is sensitive to the informativeness of the dialogue partner. We show that an agent acquires better game knowledge from dialogues when its trained policy matches the dialogue partner. We also add elementary natural language capability, and show that human dialogue partners vary in their informativeness.

Learning through communication (Chai et al.,

2018) is related to learning from demonstration (LfD) (Mulling et al., 2013; Rana et al., 2017), where the goal is for agents to learn through immediate and direct experience rather than through offline processing of large datasets. Previous work on learning through communication has focused on joint grounding of perception and language in task learning (Liu et al., 2016), complex concept grounding (Matuszek, 2018), or collaborative action (Galescu et al., 2018; Perera et al., 2018b), rather than dialogue management. Our work investigates reinforcement learning of dialogue policies, which makes it easy to produce and compare many policies. We exploit the ability to control the behavior of simulated dialogue partners to investigate policy training when dialogue partners vary in informativeness. We develop a policy with hierarchical structure based on a global policy for context-switching, and a local policy for formulating specific questions given a context.

We present two kinds of experiments. First, we compare the MDP policies for different games and different levels of informativeness of simulated dialogue partners. Results show how policies differ across games, and for different dialogue partners. For example, the agent asks more “*wh-*” questions when the dialogue partner is more forthcoming, and more “*yes/no*” questions when the dialogue partner is withholding. Second, we conduct an experiment with human dialogue partners to show that the agent can have successful dialogues with people, and that people vary in informativeness.

To add natural language capability, we developed a novel data collection method and used it to collect a dataset of 960 dialogues (12,885 turn exchanges) for Quarto, one of three *in-a-row* games that our agent can learn. The <MRL, English> pairs are then used to train NLU/NLG modules. The MRL has communicative action types that are functions from contexts to specific questions about that context. Thus the MRL combines utterance meaning with action type (similar to dialogue act type). Figure 1 illustrates one turn exchange in a graphical user interface developed for the data collection. Trained annotators presented with dialogues in MRL translated the MRL to colloquial English. We present initial results where we trained baseline natural language understanding and generation modules from this dataset to show that the agent can learn games in dialogues with people.

No other work we know of addresses the general

problem of agents learning through communication with respect to strategic knowledge, meaning knowledge about how to act when the state of the world can change through other agents’ actions or natural events. Our first main contribution is development of MDP dialogue policies for learning games through communication, based on our characterization of the learning goal in relative rather than absolute terms: to learn more and better about how to play a game. Specifically, policy training addresses the tradeoff between quality of knowledge acquired from the dialogue partner and length of the dialogue, so that the agent learns how to formulate advantageous questions. Our second main contribution is experimental evidence of the benefits of dialogue policies that are customized to the informativeness of the dialogue partner. Sensitivity to the informativeness of the dialogue partner is particularly important when the role of the dialogue partner is to provide knowledge, given that different dialogue partners can have different levels of expertise, and different communication skills.

2 Related Work

Recent work on deep reinforcement learning has made great progress in developing systems capable of learning Atari games and other games such as Chess, poker, and even Go (Silver and Hassabis, 2017; Silver and Sutskever, 2016; Dobrovsky and Hofmann, 2016). Although the agent does learn how to play the game with considerable accuracy, the process requires large amounts of data, time, and accurate perception. In contrast to this prior work, we seek an approach where an agent learns as much as it can by engaging in short, situated dialogues with human partners.

Most previous work that addresses agent learning through interaction with people, including games, involves agents learning by observing the world (learning from demonstration, or LfD). There can be some verbal input, but without significant knowledge of language or communication strategies. Virtual agents have learned games like Connect Four and Tic-Tac-Toe from demonstration videos, mapping observations to a fragment of first-order logic (Kaiser, 2012), or from sketches combined with natural language (Hinrichs and Forbes, 2013). The SOAR cognitive architecture has been applied to learning Tic-Tac-Toe and Tower of Hanoi (Kirk and Laird, 2014). In LfD, agents can also learn actions, such as how to hit a ping

pong ball (Mulling et al., 2013) or open a drawer (Rana et al., 2017). Active learning has been used for agents to ask clarification questions of a human who gives a fetching request (Whitney et al., 2017), to use pre-defined queries while learning task sequences (Racca and Kyrki, 2018), or to pose a specific question to learn a particular skill (Cakmak and Thomaz, 2012).

Previous work on learning through communication has addressed concept grounding or task learning, rather than learning how to act when the state changes due to other agents’ actions. In (Matuszek, 2018), machine-learned classifiers ground words and phrases provided by a human in an agent’s perception of the world. Language can also be grounded more directly in perception, by machine learning the relevant perceptual categories from data, rather than pre-specifying them in a formal semantics (Pillai et al., 2019). In (Liu et al., 2016), an agent learns cloth folding through rich verbal communication, based on AND-OR graphs. It can understand utterances with context dependencies common to human language but challenging for machines (e.g., descriptions of objects that evolve over several utterances). Language interaction via semantic parsing combined with deep reasoning is used in agents that explain their actions (Kasenberg et al., 2019b,a), using existing NLP tools for parsing into a logical form (Steedman and Baldridge, 2011), and a rule-based, broad-coverage toolkit for generating English from structured input (Gatt and Reiter, 2009). Other work that relies on rich, situated reasoning through multi-modal communication is based on an architecture for collaborative problem-solving (Galescu et al., 2018), with plan-based dialogue management (Perera et al., 2018a). These works either do not have distinct dialogue management modules, or rely on manually-engineered dialogue management rather than machine-learning. Our work presents machine-learned MDP policies using a method that generalizes across different games, and across differences in dialogue partners’ informativeness.

3 Game-learning Dialogues: Overview

Three games our agent learns through communication, in order of complexity, are Connect Four, Gobblet, and Quarto. In all three, players take turns placing pieces on a grid game board. The first player with four pieces in a row wins. There are different sets of possible actions per game due to

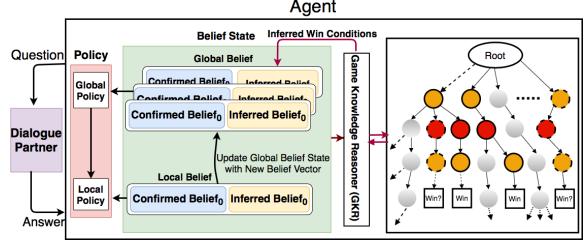


Figure 2: System Architecture.

different board sizes, numbers of game pieces, and properties that distinguish game pieces.

This paper focuses mostly on Quarto. Quarto has a 4×4 board and 16 game pieces, distinguished into two colors, two heights, two shapes, and whether they are solid or hollow. At each turn n of the game, there are $(4^2 - n) \times (4^2 - n)$ possible moves. In each turn, the opponent identifies a piece for the current player to place on the board. Four in a row wins if there is a property shared by all four pieces.

To engage in a game-learning dialogue, a Markov Decision Process (MDP) policy π chooses the agent’s dialogue actions, meaning an action a_t at time t depends on the current state s_t , which is fully observable. Reinforcement learning finds an optimal policy π to choose communicative actions that will maximize the expected total reward over time, $R_t = \mathbb{E}_\pi[\sum_{t=0}^T \gamma^t r_t]$. Here we give a brief sketch of the hierarchical policy π , dialogue actions a_t , states s_t , and reward r_t .

The multi-modal dialogues are structured as sequences of sub-dialogues, where each sub-dialogue starts with a visual demonstration of a game board showing a new way to win. The use of demonstrations of win conditions is based on observations from our previous work of how people start asking questions to learn a new game (Ayub and Wagner, 2018). As indicated below, each win condition corresponds to a path to a win state in an extensive-form game tree, where the opponent’s game actions are left unspecified. A global policy π_g chooses whether to continue the current subdialogue context, or initiate a new one, while a local policy π_l generates questions to prompt for additional win conditions based on the current demonstration, or additional information about what makes it a win. For example, the agent can ask whether the current configuration of pieces counts as a win due to the color of the pieces (see Figure 1). The use of game trees for knowledge representation is presented in section 4. We developed a meaning representation language (MRL) to represent specific communica-

tive actions a_t that are grounded in the actions and action properties of game trees (see section 5).

Game trees are a well-studied abstraction for representing game knowledge, and for executing play based on tree search. Game trees represent game states as nodes, actions as edges, with payoffs at relevant nodes (Kuhn, 1953). Each visual demonstration of a win condition presented to the agent updates the agent’s belief state s_t , as described in section 6. The belief state is also updated after a simulated or human dialogue partner (DP) responds to a question. In turn, the belief state is used to update the agent’s knowledge, represented as a game tree. For example, each visual demonstration of a win condition is interpreted as a path in a game tree from the game start to a finish in which the agent wins, and where the other player’s actions are unspecified. The agent receives a greater reward r_t when the questions it asks lead to more and better game knowledge, and receives a small penalty on each next turn to encourage efficiency. Dialogues vary in length, depending on the game and the informativeness of the DP, but most dialogues are around a dozen turn exchanges. The reward function and policy training are presented in section 7. An excerpt of a Quarto dialogue from our data collection appears in appendix A.

4 Game Trees as Knowledge

Game theory has been used to represent, reason about, and implement games (Goeree and Holt, 1999; Berlekamp et al., 1982; Ling et al., 2018). Our innovation is to use the game tree abstraction as a vehicle for 1) storing the agent’s persistent knowledge about a game, 2) reasoning about that knowledge for dialogue, and 3) providing a measurement of the quality of the game knowledge that the agent acquires in the dialogue.

We developed a game knowledge reasoner (GKR) shown in Figure 2 as an interface between the agent’s belief state during a dialogue, and its long-term knowledge store. The GKR assesses the strategic value of new win conditions that a DP has confirmed, and draws inferences about new ways to win that are added to the agent’s belief state as unconfirmed beliefs, as discussed further below.

After a dialogue, the agent’s final game tree, can be used to engage in play. In an extensive form game tree, each next depth in the tree represents action choices of alternate players. The well-known minimax algorithm (Osborne and Ru-

binstein, 1994) computes a player’s optimal action from a given node at depth d_i , on the assumption that at depth d_{i+1} the opponent always chooses its best action. The challenge of learning a new extensive form game is thereby reduced to learning enough of a game tree to engage in play. The quality of what the agent learned is reflected in how often it can win.

At the start of a dialogue, an empty extensive form game tree is initialized, and incrementally extended based on answers to the agent’s questions. Game-specific constraints specify how the game tree can grow, e.g. how many actions are available at each node. We use mapping functions from abstract actions in a game tree to physical actions, based on pre-defined information about the game-board and pieces.

The GKR computes a *strategic value* for a new win condition at a given dialog state as a function of the number of overlapping actions with existing win paths in the tree. Given a game tree with N win paths $\{W_1, W_2, \dots, W_n\}$ of length m ($W_i = \{a_{i1}, a_{i2}, \dots, a_{im}\}$), the Strategic Value (SV) for a new win path $W_j = \{a_{j1}, a_{j2}, \dots, a_{jm}\}$, $j > n$ is a conditional summation:

$$SV(W_j) = \sum_{i=1}^n \sum_{k=1}^m 1[a_{jk} \in W_i] \quad (1)$$

At a given depth in the game tree, sibling nodes represent the actions available to the corresponding player. In an incomplete game tree, some of these siblings are part of a set of win paths and some of them are not. If some of the actions at a given depth lead to win conditions, the agent infers that siblings of these actions might lead to similar win conditions. The GKR thus infers unseen board configurations based on the current game tree, and passes them to the dialogue manager as hypothesized win conditions. Formally, given a known win path $W_i = \{a_{i1}, a_{i2}, \dots, a_{im}\}$ and a sibling list of an action a_{il} of the win condition W_i ($sibling(a_{il}) = \{a'_1, a'_2, \dots, a'_k\}$) the GKR infers a maximum of k new win branches, for k remaining actions in the game, based on a sibling distance metric $SiblingDistance(a_{il}, a'_j) = d$, $j \in \{1, \dots, k\}$:

$$W_j = \{a_{i1} + d, a_{i2} + d, \dots, a'_j, \dots, a_{im} + d\} \quad (2)$$

For Connect Four and Gobblet we use a depth two game tree to make inferences about possible win conditions. For Quarto, we don’t set a depth limit. We also use the board positions of inferred

Communicative Actions of agent	
Action Type	Meaning
Conf(ChangeDisk)	Is D still a win after ChangeDisks?
Conf(ShiftBoard)	Is D still a win after ShiftBoard?
Conf(Property)	Is Property what makes D a win?
Req(ShiftBoard)	What ShiftBoard operations on D are also a win?
RequestOth()	Can the other player undo D ?
WinC(i)	Resume discussion of the i-th D .
NewWinC()	Request an unknown D .
Communicative Actions of Interlocutor	
Action Type	Meaning
Inform()	Provide some/all of the requested information
Affirm()	Positive answer to a yes/no question.
Negate()	Negative answer to a yes/no question.
Unknown()	Non-answer to a question.

Table 1: Communicative Action Types, apart from Conventional, for starting or ending dialogues.

win condition W_j to find any known win condition W_i at the same board positions, so as to infer that any feature f shared by all actions in W_j is the game piece feature that contributes to this win condition. The GKR returns this information to the dialog manager. In sum, if the agent sees a new win condition in a row where it has previously seen a win condition, and the color is what distinguishes this new win, it infers that the color is a win feature.

5 Meaning Representation

The communicative action generator takes as input the current context and the communicative action type selected by the dialogue policy, and generates a specific communicative action for the agent in an MRL we describe here. The meaning representation language is described in detail in our previous work (Zare et al., 2019). Here we explain the communicative action types of the agent and dialogue partner. The Action Types at the top of Table 1 show that the agent can ask *yes/no* questions (*Confirm*, *ConfirmOtherPlayer*), ask *wh*-questions (*Request*), resume a previous context (*WinC()*), or prompt the DP for a new demonstration (*NewWinC()*). These Action Types can be viewed as functions that return a complete MRL as a value. If no argument is shown, the current board D_i is the implicit argument. *Confirm* and *Request* can be used to ask questions about actions that can be taken on the current board (*ChangeDisks*, *ShiftBoard*) or about properties of the game pieces (*Property*).

The turn exchange in Figure 1 references a demonstrated win condition D_3 . It shows the MRL

for a *yes/no*- question asking about the contribution of color of the pieces in D_3 . Given an informative DP, a *yes/no*- question elicits a yes or no answer to an agent’s question. Here, however, the DP did not provide an answer. The kinds of answers that the agent currently understands are shown at the bottom of Table 1. A *wh*- question elicits an *Inform()* act, and a *yes/no* question elicits a positive (*Affirm()*) or negative (*Negate()*) answer, or *Unknown()*. Here we assume dialogue partners will be truthful, but may not always know the answers to questions, and may provide incomplete answers.

6 Belief State

The global belief space is a set of belief vectors B that represent beliefs acquired during a dialogue (see Figure 2). Each new demonstration D_i instantiates a new local belief vector B_i to represent confirmed information observed in D_i or acquired from responses to questions about D_i . Inferences the GKR makes about possible win conditions are also represented. A game board is represented as a vector representing each board position (e.g., 0 to 15 for Quarto), with a belief value in [0,1] for each vector position. Confirmed beliefs (B_C) and inferred beliefs (B_I) about ways to reconfigure a win condition are similar vectors with an additional position None. Formally, the game belief vector B is defined as concatenated vectors that each pertain to an observed property of game pieces (e.g., color) or a type of physical rearrangement of a configuration of pieces (e.g., rotate):

$$\begin{aligned} B_C &= b_{Color_c} \oplus \dots \oplus b_{Size_c} \oplus b_{Rotate_c} \\ &\quad \oplus b_{Translate_c} \oplus b_{OtherPlayer_c} \oplus b_{Board_c} \\ B_I &= b_{Translate_i} \oplus b_{Color_i} \oplus \dots \oplus b_{Quantity_i} \\ B &= B_C \oplus B_I \end{aligned} \tag{3}$$

Figure 1 illustrates a board demonstration D_3 for Quarto with a vertical sequence of four game pieces starting in position 2. The board D_3 is the implicit argument in the question. B_I is updated at the end of each turn with inferences derived by the GKR. For updating B_C , we rely on the baseline belief tracking method proposed in (Wang and Lemon, 2013). Given a response to a particular question, the component belief vector $vect_t$ gets updated if the turn exchange is a question and answer about a function (e.g. translate) or a property (e.g. game piece shape). When the response from the DP is positive or contains new information, the corresponding belief vectors get updated according to

equation (4). When the DP response is negative, the relevant sub-belief vectors are updated according to equation (5).

$$P_{vect_t} = 1 - (1 - P_{vect_{t-1}})(1 - P_{u_t}) \quad (4)$$

$$P_{vect_t} = (1 - P_{vect_{t-1}})(1 - P_{u_t}) \quad (5)$$

Currently, the confidence score P_{u_t} over the DP utterance is always 1.0, because there is no uncertainty in the interpretation of the MRL. (In future work, we plan to train Partially Observable MDP policies to accommodate the uncertainty in natural language interactions with humans.)

7 Policy Learning and Reward

Through simulation, we can control the informativeness of the DP’s responses, and thus investigate the impact of informativeness on policy learning. We train multiple policies, setting the DP informativeness to a value between 0 and 1. A 100% informative DP responds to all questions completely. For lower informativeness, we keep a list of all the possible winning conditions sorted by the number of times they have been presented by the DP in ascending order. When the agent asks a *NewWinC()* question, a DP with $x\%$ informativeness randomly chooses a win condition from the top $(100-x)\%$ of the sorted list. A $x\%$ informative DP responds with *Unknown()* to *Confirm()* queries with $100 - x\%$ probability, and provides only $x\%$ of a complete answer to *Request()* queries.

Gašić and Young (2014) achieved good results with less training for a Gaussian process approach to policy learning. The model has few hyperparameters and converges quickly to a local optimum ($< 20k$ epochs). We adopted their model and trained dialogue polices for $10k$ epochs. The policy gets updated at the end of each interaction.

The reward is designed to encourage the agent to acquire as many new win condition paths as possible, to prefer paths with higher strategic value, and to end the dialogue when the turn costs outweigh the gains in knowledge. Equation 6 shows the reward R for a turn exchange t as a function of the number of new win conditions in the DP’s response to a question, the strategic value SV of the response, and a turn cost C (through tuning, we found good performance from $\alpha = 0.2$, $\beta = 3$, and $C = 2$):

$$R = \lceil \frac{\#Ways to Win}{\beta} \rceil \times \alpha + SV - C \quad (6)$$

Quarto Policies with Six Dialogue Partner Levels						
Commun Act	100%	80%	60%	50%	40%	20%
NewWinC()	0.46	0.44	0.39	0.40	0.33	0.31
WinC()	0.45	0.49	0.52	0.51	0.56	0.60
Req(ShiftBoard)	0.17	0.22	0.23	0.21	0.16	0.20
Conf(ShiftBoard)	0.33	0.15	0.06	0.04	0.12	0.09
Conf(ChangeDisk)	0.01	0.01	0.03	0.04	0.09	0.15
Conf(Property)	0.47	0.62	0.65	0.68	0.58	0.47
RequestOth()	0.02	0.00	0.03	0.03	0.05	0.09
Dialogue Length	10.6	10.3	10.3	9.8	10.2	10.1

Table 2: Dialogue length and action type frequencies.

We progress here through five questions to investigate how considerations of DP informativeness can affect learning through communication.

Our first question is *how dialogue policy learning differs across levels of DP informativeness*. Figure 3 shows a sensitivity analysis of the training process over $10k$ epochs, using change in total reward, for six informativeness levels ranging from 100% to 20%. The informativeness conditions clearly differ, with lower reward for lower informativeness. We achieved similar results for Connect Four and Gobblet with much faster convergence for Connect Four, the simplest game.

Using the fully trained policies from Figure 3, we ask *how communicative actions differ during learning dialogues*. In each informativeness level, the agent engages in 100 dialogues. Table 2 reports the average frequencies of each communicative act type (except Conventional, which is always 9%, since every dialogue has an opening and a closing), and the average dialogue length in turn exchanges. *NewWinC()* and *WinC()* are equiprobable only for the 100% condition; in the other conditions, the latter is somewhat more frequent. More interestingly, the dialogue length is invariant as the agent can still learn from a low informative DP. The frequency of *Confirm(Property)* is highest for the 50% condition, the DP who is neither very informative, nor very uninformative. Similar trends were observed for Gobblet as well. However, for Connect Four, dialogues get shorter as informativeness decreases.

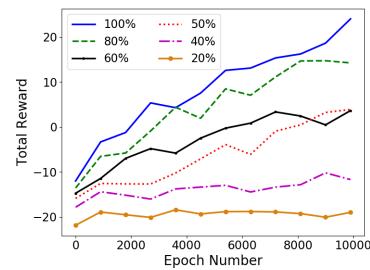


Figure 3: Total reward for six levels of DP informativeness.

Type	Policy-Dialogue Partner Condition				
	100-100	100-50	50-50	50-100	20-20
Row	40%	25%	40%	20%	10%
Col	50%	25%	45%	50%	0%
Diag	50%	0%	25%	20%	0%
AntiD	75%	25%	50%	0%	25%

Table 3: Final game knowledge under 5 dialogue conditions.

We next ask *how the policy affects what is learned in a given dialogue from a given DP type, and what happens if the agent’s learned policy for a DP level X is used when interacting with a DP of level Y*. Table 3 shows five policy-DP ($X-Y$) conditions we tested. Under each condition, one dialogue from a set of ten dialogues was randomly selected where we inspected the final game tree knowledge. Quarto has four win condition locations, labeling the table rows. The most interesting result common among all three games is that if the DP is neither informative nor uninformative (50%), the agent gains the most game knowledge from using a matching policy (50-50). Note that the agent learns less from a 100% DP using the wrong policy than from a 50% DP using the right policy.

We next ask *how well can the agent play after a learning dialogue*. For Connect Four and Gobblet, we recruited 16 students to play with the agent, using the same conditions and knowledge states from Table 3. Because the slow movements of our Baxter robot (Rethink robotics) resulted in tedious 20-minute games, we used a simulated agent at a terminal. Prior to data collection, each subject played a few practice games to become familiar with the game and the interface. Each subject played 10 games, randomly ordered among the 5 conditions. We set a time limit of 2.5 minutes for each game and used a Minimax algorithm with 2 step look-ahead. We observed that the quantity differences in knowledge acquired by the agent show up directly as quality differences for Connect Four. For Gobblet the proportion of outcomes for the agent were more or less the same across the conditions involving a 50% policy and/or a 50% DP. We attributed the uniform Gobblet results to the time limit for the play and to the need for greater look-ahead, given the many action choices.

Result	Policy-Dialogue Partner Condition				
	100-100	100-50	50-50	50-100	20-20
Wins	0.94	0.19	0.50	0.18	0.12
Losses	0.00	0.81	0.47	0.78	0.82
Draws	0.06	0.00	0.03	0.04	0.06

Table 4: Percentage of agent wins/losses/draws.

For Quarto, we altered the experiment by removing the restriction on length of play and depth of search. We also developed a graphical user interface to display game pieces in a more realistic way. We recruited 18 students to play Quarto. The game results in Table 4 show that the agent won games more often when it had learned the game from a more informative DP, as long as it used the corresponding policy.

Our final question was *whether the agent could use the same policy to continue learning over a sequence of dialogues*. Here we looked at three conditions: where the learned policy matched the DP informativeness of 100%, 50% and 20%. In each condition, the agent had four dialogues, starting with no knowledge. The agent began each next dialogue with the knowledge it had gained from its previous dialogue. We averaged the final reward at the end of each dialogue. Results show that the agent continues to learn more and more about the game, especially from the 100% informative DP. Results for Gobblet were very similar to Quarto. However for Connect Four, there is usually little reward (knowledge) left to gain after the first or second dialogue in higher informativeness levels, so the reward plateaus after two or three dialogues.

8 Dialogue Data Collection

To add natural language capability for the agent, we developed a novel data collection method to produce a corpus consisting of $\langle \text{Gameboard}, \text{MRL}, \text{NL} \rangle$ tuples for each utterance in 960 dialogues between an agent and simulated dialogue partner. The Quarto Dialogue corpus is distinctive in that it is agent-agent situated, multi-modal dialogue where agents’ utterances are in an MRL, then all dialogues translated by experts into English.

To our knowledge, this is the first corpus of its kind. Most previous dialogue corpora we know of fall into one of three other categories: human-Wizard-of-Oz, human-agent, or human-human. Corpora for human-Wizard-of-Oz are used either to in-

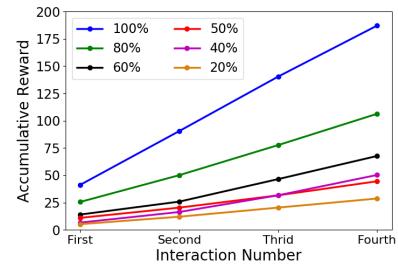


Figure 4: Consecutive dialogues reward trend.

form manually engineered dialogue management or as training data for machine learned dialogue managers. These corpora are collected for the purpose of restaurant reservation (Henderson et al., 2013), finding available vacation accommodations (Asri et al., 2017), or even open-domain information retrieval systems (Rosset and Petel, 2006). Human-agent corpora are often annotated with dialogue acts for applications such as travel booking systems (Bennett and Rudnicky, 2002). Human-human corpora are either collected under constrained settings where humans are instructed to follow a series of instructions (Brennan et al., 2013; Heeman and Allen, 1995), or are naturally occurring conversations between humans (Asher et al., 2016; Afantinos et al., 2012; Passonneau and Sachar, 2014). Distinctive characteristics of the Quarto corpus are that every utterance has an MRL and a natural language version where the MRL is a communicative act. The dialogues involve a shared multi-modal context, leading to deictic reference to the game board and with a known structure into sub-dialogues.

To collect our corpus, we developed two graphical user interfaces (GUIs) to display a schematic representation of the current board demonstration (cf. Figure 1), and to allow annotators to page through each turn exchange. One GUI was for the translation task, and a second was to collect ratings on the translations. Thirteen undergraduate students from a course in Artificial Intelligence participated as part of their course work. Students were first trained in the MRL, including comparisons with the first order logic translations of English that students had learned in class. Their instructions were to translate into colloquial English. Meetings were held where students discussed examples and asked questions. All translations were rated for correctness and naturalness on a five-point scale where 5 was the top. On average, correctness was 4.79 and naturalness was 4.72.

The 960 dialogues contain 12,885 turn exchanges. The English translations contain 229,641 word tokens, and 1,498 word types. The NLG data has 146,055 tokens and 1,102 types. The NLU data is somewhat less rich, with 83,586 tokens and 952 types. The 960 dialogues consist of 535 from a 60% informative simulator, 255 from a 100% informative simulator, and 170 from a 50% simulator. We are currently augmenting the data to synthesize new examples for Quarto, and to synthesize Connect Four and Gobblet data.

Because all turn exchanges are tied to a physical board, the corpus is rich in spatial references. The students referred to the pieces by specific attributes (e.g. *next to that green circular piece*), exact location on the board (e.g. *top corner piece*), relation with other pieces (e.g. *to the right of the square piece*), or deictic reference (e.g. *this piece here*). There are also many anaphoric references (e.g. *about that win you showed, the second win*).

9 Human-Agent Dialogues

The dataset described above provides training data for NLU and NLG modules to enable the agent to engage in dialogue with humans. Two other changes needed to support future human-agent dialogues are clarification sub-dialogues to handle misunderstandings or confusions, and modification of the policy training and belief updates to address uncertainty in the NLU. To preview our future challenges, we developed baseline NLU and NLG modules, and asked the 18 subjects who played Quarto with our agent to engage in text-based dialogues. Here we describe the dialogue interface, the baseline NLU and NLG modules, the dialogue outcomes, and the subjects' informativeness.

We developed a text-based GUI for subjects to engage in dialogues with an agent, similar to the GUI used for translating MRL into English. For NLG and NLU, we trained two sequence-to-sequence RNN models with two hidden layers and a Bahdanau attention layer (Bahdanau et al., 2015). The Adam optimizer was used for training (Kingma and Ba, 2014) (20 epochs for NLG, and 15 for NLU). The MDP policy for 100% informativeness was used, and belief updating remained the same.

Each subject engaged in two dialogues. Average dialogue length was 10.96 turn exchanges (min 9, max 15, std 2.15), which is similar to dialogues with the simulator. Subjects also completed a questionnaire. The questionnaire¹ asked subjects 1) whether they understood the agent's questions, 2) to list the confusing questions by turn number, 3) to rate the dialogues on a 5-point scale for the agent's command of English, and 4) to tell us how willing they would be to have another dialog with this agent. Fourteen of the subjects said they understood the agent most of the time. Inspection of the questions listed as confusing indicated they all had incomplete or incorrect NLG output. The average fluency rating was 2.93. Eleven subjects said they

¹See Appendix B for the complete list of questions.

Win Type	Mean	Min	Max	SDev
Row	20%	0%	35%	7.8
Col	20%	0%	40%	8.3
Diag	15%	0%	50%	13.3
Anti-Diag	5%	0%	10%	1.2

Table 5: Average final knowledge states for the 36 dialogues

would be willing to have more dialogues, one was neutral, and six were somewhat dissatisfied.

The overall quality of the NLG was good; two thirds of the agent questions were fluent and correct. Of 197 total turn exchanges, 58 were less than perfect. One of the co-authors rated all the generated questions on a five-point scale for correctness and intelligibility, yielding an average score of 4.19 (min 1, max 5, std 1.23). The NLU quality was less good. Subjects’ answers were translated to a gold-standard MRL by one of the co-authors, and compared with the NLU output; only 60% of the answers were interpreted correctly. Despite the agent’s frequent failure to understand subjects’ responses, the average total reward of 12.45 was comparable to the reward for an 80% informative simulator with a matching policy (cf. Figure 3). Table 5 gives the average final knowledge states for the 36 dialogues, which is in the same range as for dialogues with a 50% informative DP and matching policy (see Table 3). To assess the subjects’ informativeness, we examined the 139 turn exchanges that subjects understood well, comparing the subjects’ answers to 100% informative answers. Subjects’ answers were 100% informative only 41% of the time.

The comparison of baseline human-agent learning dialogues with those between an agent and simulated DP shows promise for reinforcement learning of policies that are trained offline in simulation. Subjects provided less than 100% informative answers, and the agent’s final knowledge states were similar to those where the agent interacted with a 50% informative simulator, using a matching policy. Even without the ability to engage in clarification sub-dialogues with a human to clear up confusions, the dialogues were all completed. The agent was completely understandable two thirds of the time. The agent learned as much about Quarto as in the 50%-50% simulator condition.

A question raised by these results is how an agent could benefit from having access to multiple dialogue policies. In robotics, a very similar question has been addressed for agents learning motor skills through simulation and deploying the learned

policies in real-world environments with unknown dynamics. Approaches include learning to linearly combine a family of policies (Zhang et al., 2018), learning a classifier for environment parameters to choose the correct policy (Yu et al., 2017), or searching directly within a family of policies using the current accumulated reward (Yu et al., 2019). Similar methods could be applied to exploit a family of dialogue policies to adapt questioning strategies in different ways, depending on the observed behavior of the dialogue partner.

10 Conclusion

Our results show that agents can learn MDP policies to learn board games through multi-modal dialogues using a relative knowledge goal, namely to increase the agent’s game knowledge as much as possible during a short dialogue. We also show that the agent learns different dialogue policies depending on the dialogue partner’s informativeness. This work exploits the benefits of a knowledge domain that has a very abstract representation in the form of game trees, where a novel meaning representation language is grounded in the game tree abstraction. This approach can generalize to a wide range of two-person board games, and provides a foundation for communication learning about other strategic activities. In addition, an agent that can learn new games and then engage in play has potential benefits in Socially Assistive Robotics (Feil-Seifer and Mataric, 2005). Board games have been used to delay the onset of dementia (Dartigues et al., 2013), and have been shown to help children learn computational concepts (Berland and Lee, 2011).

Additionally, we have demonstrated that MDP policies trained offline in simulation can lead to fairly effective human-robot learning dialogues, based on training data for natural language modules we collected through a novel procedure. Our future work will expand the communicative actions to include clarifications, will train POMDP policies, and will borrow ideas from reinforcement learning of robotic motor skills to close the reality gap between offline training of dialogue policies and engaging in real-world dialogues with humans.

Acknowledgements

This work was supported by a grant from Penn State’s Institute for Computational and Data Sciences. We thank the students from the Fall 2019 course in AI who contributed to the data collection.

References

- Stergos Afantinos, Nicholas Asher, Farah Benamara, Myriam Bras, Cécile Fabre, Lydia-Mai Ho-Dac, Anne Le Draoulec, Philippe Muller, Marie-Paule Pery-Woodley, Laurent Prévot, et al. 2012. An empirical resource for discovering cognitive principles of discourse organisation: the ANNODIS corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2727–2734.
- Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos Afantinos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 16)*.
- Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A corpus for adding memory to goal-oriented dialogue systems. *arXiv preprint arXiv:1704.00057*.
- Ali Ayub and Alan R Wagner. 2018. Learning to win games in a few examples: Using game-theory and demonstrations to learn the win conditions of a Connect Four game. In *Social Robotics*, pages 349–358. Springer International Publishing.
- Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *(International Conference on Learning Representations ICLR)*.
- Christina Bennett and Alexander I Rudnicky. 2002. The Carnegie Mellon Communicator corpus. In *Seventh International Conference on Spoken Language Processing*.
- Matthew Berland and Victor R. Lee. 2011. Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, 1(2):65–81.
- E. Berlekamp, J. H. Conway, and R. Guy. 1982. *Winning ways for your mathematical plays: Games in general*. Academic Press.
- Susan Brennan, Katharina Schuhmann, and Karla Bates. 2013. Entrainment on the move and in the lab: The walking around corpus. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 35.
- M. Cakmak and A. L. Thomaz. 2012. Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE International conference on Human-Robot Interaction*.
- Joyce Y. Chai, Qiaozi Gao, Lanbo She, Shaohua Yang, Sari Saba-Sadiya, and Guangyu Xu. 2018. *Language to action: Towards interactive task learning with physical agents*. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2–9. International Joint Conferences on Artificial Intelligence Organization.
- Jean François Dartigues, Alexandra Foubert-Samier, Mélanie Le Goff, Mélanie Viltard, Hélène Amieva, Jean Marc Orgogozo, Pascale Barberger-Gateau, and Catherine Helmer. 2013. Playing board games, cognitive decline and dementia: a French population-based cohort study. *BMJ open*, 3(8).
- Borghoff-U. M. Dobrovsky, A. and M. Hofmann. 2016. An approach to interactive deep reinforcement learning for serious games. In *7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*.
- David Feil-Seifer and Maja J. Mataric. 2005. Defining socially assistive robotics. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 465–468.
- Lucian Galescu, Choh Man Teng, James Allen, and Ian Perera. 2018. *Cogent: A generic dialogue system shell based on a collaborative problem solving model*. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 400–409, Melbourne, Australia. Association for Computational Linguistics.
- Milica Gašić and Steve Young. 2014. Gaussian processes for POMDP-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40.
- Albert Gatt and Ehud Reiter. 2009. *SimpleNLG: A realisation engine for practical applications*. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 90–93, Athens, Greece. Association for Computational Linguistics.
- Jacob K Goeree and Charles A Holt. 1999. Stochastic game theory: For playing games, not just for doing theory. *Proceedings of the National Academy of sciences*, 96(19):10564–10567.
- Peter A Heeman and James F Allen. 1995. The trains 93 dialogues. Technical report, Rochester University, Dept. of Computer Science.
- Matthew Henderson, Blaise Thomson, and Jason Williams. 2013. Dialog state tracking challenge 2 & 3 handbook. *camdial.org/mh521/dstc*.
- Thomas R. Hinrichs and Kenneth D. Forbus. 2013. X goes first: Teaching simple games through multimodal interaction. In *Proceedings of the Second Conference on Advances in Cognitive Systems*, pages 31–46.
- Lukasz Kaiser. 2012. Learning games from videos guided by descriptive complexity. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 963–69.

- Daniel Kasenberg, Antonio Roque, Ravenna Thielstrom, Meia Chita-Tegmark, and Matthias Scheutz. 2019a. Generating justifications for norm-related agent decisions. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 484–493, Tokyo, Japan. Association for Computational Linguistics.
- Daniel Kasenberg, Antonio Roque, Ravenna Thielstrom, and Matthias Scheutz. 2019b. Engaging in dialogue about an agent’s norms and behaviors. In *Proceedings of the 1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI)*.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.
- James R. Kirk and John E. Laird. 2014. Interactive task learning for simple games. In *Advances in Cognitive Systems*, pages 11–28.
- H.W. Kuhn. 1953. Extensive form games and the problem of information. In *Contributions to the Theory of Games II*, page 193–216. Princeton University Press.
- Chun Kai Ling, Fei Fang, and J. Zico Kolter. 2018. What game are we playing? end-to-end learning in normal and extensive form games. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 396–402. International Joint Conferences on Artificial Intelligence Organization.
- Changsong Liu, Shaohua Yang, Sari Saba-Sadiya, Nishant Shukla, Yunzhong He, Song-Chun Zhu, and Joyce Chai. 2016. Jointly learning grounded task structures from language instruction and visual demonstration. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1492, Austin, Texas. Association for Computational Linguistics.
- Cynthia Matuszek. 2018. Grounded language learning: Where robotics and nlp meet. In *IJCAI*, pages 5687–5691.
- K. Mulling, J. Kober, O. Kroemer, and J. Peters. 2013. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research (IJRR)*.
- Martin J Osborne and Ariel Rubinstein. 1994. *A course in game theory*. MIT press.
- Rebecca J. Passonneau and Evaneet Sachar. 2014. Loqui human-human dialogue corpus (transcriptions and annotations). Columbia University Academic Commons.
- Ian Perera, James Allen, Choh Man Teng, and Lucian Galescu. 2018a. Building and learning structures in a situated blocks world through deep language understanding. In *Proceedings of the First International Workshop on Spatial Language Understanding*, pages 12–20, New Orleans. Association for Computational Linguistics.
- Ian Perera, James Allen, Choh Man Teng, and Lucian Galescu. 2018b. A situated dialogue system for learning structural concepts in blocks world. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 89–98, Melbourne, Australia. Association for Computational Linguistics.
- Nisha Pillai, Cynthia Matuszek, and Francis Ferraro. 2019. Deep learning for category-free grounded language acquisition. In *Proc. of the NAACL Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics (NAACL-SpLU-RoboNLP)*, Minneapolis, MI, USA.
- Mattia Racca and Ville Kyrki. 2018. Active robot learning for temporal task models. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 123–131. ACM.
- M. A. Rana, M. Mukadam, S. R. Ahmadzadeh, S. Chernova, and B. Boots. 2017. Towards robust skill generalization: Unifying learning from demonstration and motion planning. In *Conference on Robot Learning (CoRL)*.
- Sophie Rosset and Sandra Petel. 2006. The ritel corpus—an annotated human-machine open-domain question answering spoken dialog corpus. In *LREC*, volume 6, pages 1640–1643.
- Huang-A. Maddison C. J. Guez A. Sifre L. Driessche G. v. d. Schrittwieser J. Antonoglou I. Panneershelvam V. Lanctot M. Dieleman S. Grewe D. Nham J. Kalchbrenner N. Silver, D. and I. Sutskever. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489.
- Hubert-T. Schrittwieser J. Antonoglou I. Lai M. Guez A. Lanctot M. L. Sifre Kumaran D. Graepel T. Lillicrap T. Simonyan K. Silver, D. and D. Hassabis. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv Reprint arXiv:1712.01815*.
- Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. In R.D. Borsley and K. Börjars, editors, *Non-Transformational Syntax*, pages 181–224. Wiley.
- Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432.
- D. Whitney, E. Rosen, J. MacGlashan, L. L. Wong, and S. Tellex. 2017. Reducing errors in object-fetching interactions through social feedback. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Wenhai Yu, C. Karen Liu, and Greg Turk. 2019. Policy transfer with strategy optimization. In *International Conference on Learning Representations (ICLR)*.

Wenhai Yu, Jie Tan, C. Karen Liu, and Greg Turk. 2017. Preparing for the unknown: Learning a universal policy with online system identification. In *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts.

Maryam Zare, Ali Ayub, Alan R. Wagner, and Rebecca J. Passonneau. 2019. Show me how to win: A robot that uses dialog management to learn from demonstrations. In *Fourth Games and Natural Language Processing Workshop (GAMNLP-19)*, San Luis Obispo, CA.

Chao Zhang, Yang Yu, and Zhi-Hua Zhou. 2018. Learning environmental calibration actions for policy self-evolution. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*.

A Appendix A: Excerpt of a Quarto Dialogue

Here is a sample Quarto dialog from the data collection, showing the MRL,NL pairs, and images of the relevant board contexts.

- **Q1) Start()**

NL: Hi there! Can you teach me how to play Quarto?

A1) Start()

NL: Sure thing, let's get started.

- **Q2) RequestNewWinCondition()**

NL: Can you show me an example of a win?

A2)

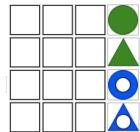


Figure 5: First Win Condition (D_1)

- **Q3) RequestNewWinCondition()**

NL: Alright, how about another?

A3)

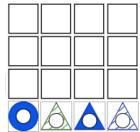


Figure 6: Second Win Condition (D_2)

- **Q4) ContextSwitch (D_1), Confirm(translate($col0$))**

NL: In the first example you showed me, would it still count if the pieces were placed down the first column instead of down the fourth?

A4) Affirm()

NL: Yeah, it would.

- **Q5) Same, Request(rotate())**

NL: Say I keep the blue triangle in the bottom-right corner, but try placing the remaining pieces along some other direction upward, to the left, or a combination of both. Could I still get a win?

A5) Inform([135, 180])

NL: You could get a win so long as you placed the remaining pieces along the anti-diagonal, or leftward along the bottom row.

- **Q6) Same, Confirm(shape)**

NL: Does this count because of the shape of the pieces?

A6) Negate()

NL: No, this win is not based on the shapes.

- **Q7) Same, Confirm(quantity)**

NL: Well, does this count because there are four and only four pieces?

A7) Affirm()

NL: Yes, it wouldn't count for any other number of pieces.

- **Q8) ContextSwitch (D_2), Confirm(removepiece((0,1))**

NL: Okay, in the second example you showed me, what would happen if I removed the green triangle?

A8) Negate()

NL: Then you would no longer have a win.

- **Q9) RequestNewWinCondition()**

NL: Can you show me another example of a win?

A9)

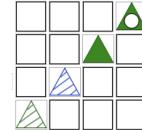


Figure 7: Third Win Condition (D_3)

- **Q10) Finish**

NL: Great, thanks for helping me get started!

A10) Finish

NL: I'm happy I could help!

B Appendix B: Questionnaire

Here we present the complete list of questions asked from students.

1. Were you able to understand what the agent was asking you even when it was not completely fluent English?
2. If there were questions you could not understand, please list them below by turn number.
3. For questions you could not understand please try to explain to us your confusion for each of the turns you listed above. (Please use the turn number again)
4. On a scale of 1 to 5 (5 is the best) how likely you would be think that this dialog was typed in by an English speaker?
5. What aspects of this dialog did you find interesting, if any?
6. How likely you would come back and have another dialog about a game with this agent?
7. What aspects of the GUI do you think can be improved?

Author Index

- Arimoto, Tsunehiro, 323
Ayub, Ali, 339

Baiju, Aiswarya, 246
Bao, Haogang, 257
Beaver, Ian, 79
Bergqvist, Amanda, 209
Bhattacharya, Indrajit, 136
Bryant, Caleb, 290

Cao, Yan, 329
Carne, Jake, 132
Cervone, Alessandra, 162
Chang, Jonathan P., 57
Chen, Xiaoping, 107, 329
Chiam, Caleb, 57
Choi, Jinho D., 236, 261

Danescu-Niculescu-Mizil, Cristian, 57
Di Eugenio, Barbara, 246
Dondrup, Christian, 45

Eric, Mihail, 278
Eskenazi, Maxine, 225

Fan, Wei, 61
Favre, Benoit, 203
Finch, James D., 261
Finch, Sarah E., 236
Fosler-Lussier, Eric, 196
Fu, Liye, 57
Fuscone, Simone, 203

Gálvez, Ramiro H., 215
Gao, Jianfeng, 297
Gasic, Milica, 35
Gauder, Lara, 215
Geishauser, Christian, 35
Gerber, Ben, 246
Gervits, Felix, 86
Ghosh, Subhasis, 136
Gindi, Aaron, 128
Goindani, Mahak, 74
Goldwasser, Dan, 74
Gopalakrishnan, Karthik, 278

Gravano, Agustín, 215
Gunson, Nancie, 45
Gupta, Itika, 246

Hakkani-Tur, Dilek, 21, 278
Harrison, Vrindavan, 21
Heck, Michael, 35
Hedayatnia, Behnam, 278
Hewitt, Timothy, 79
Higashinaka, Ryuichiro, 323
Hoffmann, Joerg, 53
Hsueh, Cheng-Hsun, 1
Huang, Minlie, 297

Inoue, Koji, 118
Ishiguro, Hiroshi, 323

Jain, Ayush, 74
Jhunjhunwala, Megha, 290

Kane, Benjamin, 128
Karkada, Deepthi, 209
Kawahara, Tatsuya, 118
Kawanishi, Takahito, 323
Kennington, Casey, 97, 132
Kim, Seokhwan, 278
Köhn, Arne, 53
Koller, Alexander, 53
Komatani, Kazunori, 147
Kundu, Arpita, 136

Lala, Divesh, 118
Lancette, Steven, 74
Lemon, Oliver, 45
Li, Jinchao, 297
Lin, Hsien-Chin, 35
Liu, Aishan, 339
Liu, Bing, 246
Liu, Yang, 278
Liu, Ye, 61
Lu, Keting, 107, 329
Lubis, Nurul, 35
Luque, Jordi, 215

Ma, Jun, 186

- Ma, Wei-Yun, 1
Maier, Wolfgang, 311
Manuvinakurike, Ramesh, 209
Marchand, Lucas, 132
McMahan, Brian, 175
McNeill, David, 97, 132
Mehri, Shikib, 225
Michael, Thilo, 49, 157
Mihalcea, Rada, 10
Möller, Sebastian, 157
Moresi, Marco, 35
Moro, Daniele, 132
Murray, R. Charles, 257

Nabulsi, Nadia, 246
Nakamura, Shizuka, 118
Nakano, Mikio, 147

Oraby, Shereen, 21

Pacheco, Maria, 74
Paetzel, Maike, 209
Passonneau, Rebecca, 339
Peng, Baolin, 297
Pérez-Rosas, Verónica, 10
Platonov, Georgiy, 128
Pramanick, Aniket, 136
Prévot, Laurent, 203

Reed, Lena, 21
Riccardi, Giuseppe, 162
Roque, Antonio, 86
Rose, Carolyn, 257

Sawada, Hiroshi, 323
Schäfer, Christine, 53
Scheutz, Matthias, 86
Schubert, Lenhart, 128
Shah, Pararth, 290
Sharp, Lisa, 246
Shen, Siqi, 10
Shi, Hongjie, 272
Sieińska, Weronika, 45
Smart, Mary, 246
Song, Qi, 196
Stiff, Adam, 196
Stone, Matthew, 175
Stone, Peter, 107
Sudhakara, Sweekar, 339
Sugiyama, Hiroaki, 323

Takanashi, Katsuya, 118
Takanobu, Ryuichi, 297

Tanaka, Kou, 323
Thielstrom, Ravenna, 86
Torralba, Álvaro, 53

Ultes, Stefan, 311

van Niekerk, Carel, 35

Wagner, Alan, 339
Walker, Marilyn, 21
Wang, Andrew, 57
Wang, Shaojun, 186
Wang, Yan, 186
Wang, Yansen, 257
Welch, Charles, 10
Wichlacz, Julia, 53

Xiao, Jing, 186

Yamamoto, Kenta, 118
Yang, Tao, 61
You, Zeyu, 61
Yu, Philip S., 61

Zare, Maryam, 339
Zhang, Jiayu, 186
Zhang, Justine, 57
Zhang, Shiqi, 107, 329
Zhu, Chenguang, 265
Zhu, Qi, 297
Ziebart, Brian, 246