Lecture Notes in Electrical Engineering 943

Svetlana Stoyanchev
Stefan Ultes
Haizhou Li *Editors*

Conversational AI for Natural Human-Centric Interaction

12th International Workshop on Spoken Dialogue System Technology, IWSDS 2021, Singapore

Springer

Lecture Notes in Electrical Engineering

Volume 943

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering—quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada:

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries:

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

Svetlana Stoyanchev · Stefan Ultes · Haizhou Li
Editors

Conversational AI for Natural Human-Centric Interaction

12th International Workshop on Spoken Dialogue System Technology, IWSDS 2021, Singapore

Springer

Editors
Svetlana Stoyanchev
Toshiba (United Kingdom)
Weybridge, UK

Stefan Ultes
Daimler (Germany)
Stuttgart, Germany

Haizhou Li
The Chinese University of Hong Kong
Shenzhen, China

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-19-5537-2 ISBN 978-981-19-5538-9 (eBook)
https://doi.org/10.1007/978-981-19-5538-9

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Organization

General Chair

Haizhou Li, The Chinese University of Hong Kong, Shenzhen, China

Program Chairs

Svetlana Stoyanchev, Toshiba Europe Ltd., UK
Stefan Ultes, Mercedes-Benz AG, Germany

Publication Chair

Minghui Dong, Institute for Infocomm Research, Singapore

Finance Chair

Yan Wu, Institute for Infocomm Research, Singapore

Sponsorship Chair

Lei Wang, COLIPS, Singapore

Senior Steering Committee

David Traum, University of Southern California, USA

Local Arrangement Committee

Grandee Lee, National University of Singapore, Singapore
Yan Zhang, National University of Singapore, Singapore
Siqi Cai, National University of Singapore, Singapore
Chen Zhang, National University of Singapore, Singapore
Bidisha Sharma, National University of Singapore, Singapore
Chitralekha Gupta, National University of Singapore, Singapore
Berrak Sisman, Singapore University of Technology and Design, Singapore
Celine Cheong, National University of Singapore, Singapore
Min Yuan, National University of Singapore, Singapore
Kun Zhou, National University of Singapore, Singapore
Yi Zhou, National University of Singapore, Singapore
Rui Liu, National University of Singapore, Singapore
Mingyang Zhang, National University of Singapore, Singapore

Program Committee

Alexandros Papangelis, Uber, USA
Andreea Niculescu, Institute for Infocomm Research, Singapore
Axel Buendia, CNAM, France
Bayan Abushawar, Arab Open University, Saudi Arabia
Carl Vogel, Trinity College Dublin, Ireland
David Griol, University of Granada, Spain
Dhivya Piraviperumal, Apple, USA
Dilek Hakkani-Tur, Amazon Alexa AI, USA
Emer Gilmartin, Trinity College Dublin, Ireland
Fabrice Lefèvre, Avignon Université, France
Felix Burkhardt, audEERING, Germany
Fernando Fernández-Martínez, Universidad Politécnica de Madrid, Spain
Gennaro Cordasco, Universita' della Campania "L. Vanvitelli", Italy
Harksoo Kim, Konkuk University, Korea
Hsin-Min Wang, Academia Sinica, Taiwan
Huiru Zheng, University of Ulster, UK
Jan Alexandersson, DFKI GmbH, Germany
Jonathan Ginzburg, Université Paris-Diderot (Paris 7), France

José David Lopes, Heriot Watt University, UK
Jose F Quesada, University of Seville, Spain
Joseph Mariani, LIMSI-CNRS, France
Julia Hirschberg, Columbia University, USA
Justine Cassell, Carnegie Mellon University, USA
Kallirroi Georgila, University of Southern California, USA
Kazunori Komatani, Osaka University, Japan
Khalid Choukri, ELRA/ELDA, France
Kheng Hui Yeo, Institute for Infocomm Research, Singapore
Kiyonori Ohtake, NICT, Japan
Korbinian Riedhammer, Technische Hochschule Nürnberg Georg Simon Ohm, Germany
Kristiina Jokinen, AIRC AIST Tokyo Waterfront, Japan
Leila Ben Letaifa, Universidad del País Vasco UPV/EHU, Spain
Leonardo Campillos Llanos, Consejo Superior de Investigaciones Científicas, Spain
Luis Fernando D'Haro, Universidad Politécnica de Madrid, Spain
Martin Heckmann, Aalen University, Germany
Matthew Henderson, Google, UK
Michael Mctear, University of Ulster, UK
Nigel Ward, The University of Texas at El Paso, USA
Norbert Braunschweiler, Toshiba Research Europe Ltd., UK
Pierre Lison, Norwegian Computing Center, Norway
Rafael Banchs, Intapp, USA
Rainer Gruhn, Harman Becker Automotive Systems, Germany
Ramon Lopez-Cozar Delgado, University of Granada, Spain
Ryuichiro Higashinaka, Nagoya University/NTT, Japan
Sakriani Sakti, Nara Institute of Science and Technology, Japan
Satoshi Nakamura, Nara Institute of Science and Technology, Japan
Simon Keizer, Toshiba Europe Ltd., UK
Sophie Rosset, LIMSI, CNRS, Université Paris-Saclay, France
Susanne Burger, Carnegie Mellon University, USA
Timo Baumann, University of Hamburg, Germany
Wolfgang Maier, Mercedes-Benz AG, Germany
Zoraida Callejas, University of Granada, Spain

Preface

The 12th International Workshop on Spoken Dialogue Systems (IWSDS 2021) was held remotely and on-site on 15–17 November 2021 in Singapore. This year's conference theme was "Conversational AI for natural human-centric interaction" putting an emphasis on the naturalness of the interaction with the user at its center.

The IWSDS conference series constitute a consolidated forum where international researchers, practitioners, and stakeholders working in the field of spoken dialogue systems and associated technologies can disseminate their current research and applications, discuss technological challenges, present their success stories, and share their complementary visions about the future of the technology. IWSDS 2021 was grounded on the experience and knowledge generated in the previous workshops:

- IWSDS'09 (Irsee, Germany),
- IWSDS'10 (Gotemba Kogen Resort, Japan),
- IWSDS'11 (Granada, Spain),
- IWSDS'12 (Paris, France),
- IWSDS'14 (Napa, USA),
- IWSDS'15 (Busan, Korea),
- IWSDS'16 (Saariselkä, Finland),
- IWSDS'17 (Farmington, PA, USA),
- IWSDS'18 (Singapore, Singapore),
- IWSDS'19 (Siracusa, Italy) and
- IWSDS'20 (Madrid, Spain/remote).

The conference invited and received paper submissions on the following topics:

- User engagement and emotion in dialogue systems,
- Proactive, anticipatory, or incremental interaction,
- Use of humor and metaphors in automatic dialogue systems,
- Multimodal and situated dialogue systems,
- Companions and personal assistant dialogue systems,
- Educational and healthcare applications,
- Big data and large scale spoken dialogue systems,

- Digital resources for interactive dialogue management,
- Domain Transfer and adaptation techniques for spoken dialogue systems,
- Spoken dialogue systems for low-resource languages and multilingual systems,
- Dialogue system evaluation.

However, submissions were not limited to these topics, and submission of papers in all areas related to spoken dialogue systems was encouraged. The contributions were grouped into four categories: (a) long research papers targeting reports on mature research results, (b) short research papers targeting smaller case studies or ongoing but interesting and original research efforts, (c) position papers to present novel research ideas or viewpoints which describe trends or fruitful starting points for future research and elicit discussion, and finally (d) demo submissions–system papers to demonstrate innovative or industrial-based research.

In addition, the IWSDS 2021 included the WOCHAT: **WO**rkshop on **CH**atbots and Conversational **A**gen**T**s special session. It was organized by Ryuichiro Higashinaka (Nagoya University, Japan), João Sedoc (New York University, USA), Luis F. D'Haro (Universidad Politécnica de Madrid, Spain), Rafael E. Banchs (Intapp Inc, USA), and Alexander Rudnicky (Carnegie Mellon University, USA). This was the eighth event of a Special Session Series on Chatbots and Conversational Agents. WOCHAT aims at bringing together researchers working on problems related to chat-oriented dialogue with the objective of promoting discussion and knowledge sharing about the state of the art and approaches in this field, as well as coordinating a collaborative effort to collect/generate data, resources, and evaluation protocols for future research in this area. The session invited original research contributions on all aspects of chat-oriented dialogue, including knowledge representation, reasoning, natural language generation, and understanding. The presented papers discussed data collection, user state detection, question answering, language generation, and evaluation of chat-oriented dialogue.

IWSDS 2021 received a total of 26 submissions, where each submission was reviewed by at least three Program Committee members. The committee decided to accept a total of 24 papers distributed as follows: 15 long papers and three short papers for the general track and four long papers and two short papers for the WOCHAT session. Finally, 21 papers are included in the proceedings.

The program included three keynotes by renowned international experts:

- Dr. Maxine Eskenazi, Carnegie Mellon University, Pittsburgh, PA, USA,
- Prof. Dr. Helen Hastie, Heriot Watt University, Edinburgh, UK, and
- Dr. Jinho D. Choi, Emory University, Atlanta, GA, USA.

The keynote speech by Dr. Maxine Eskenazi was entitled "User-centric dialog". In her talk, she focused on a paradigm shift of recent research turning from being agent-centric to being user-centric. She emphasizes that this shift is important if we are to create systems acceptable to the general population of users. Dr. Eskenazi's talk described the reasoning behind user-centric research and looked at concrete ways to apply this point of view to system training, such as using implicit feedback from the user's consecutive utterances as a signal for training a model. She highlighted the benefits of “real” motivated users over paid experiment subjects. Dr. Eskenazi pointed out that the “Turing test” may not be necessarily the right metric for evaluation of dialogue systems which can be effective without having to emulate human behavior. She concluded with addressing user-centric strategies when dealing with a malevolent user.

Professor Helen Hastie presented a keynote entitled “Trustworthy Interactive Robots”. She described trust as a multifaceted, complex phenomenon that is not well understood when it occurs between humans, let alone between humans and robots. For Prof. Hastie, robots that portray social cues, including voice, gestures, and facial expressions, are key tools in researching human-robot trust, specifically how trust is established, lost, and regained. In her talk, she discussed various aspects of trust for HRI including language, social cues, embodiment, transparency, mental models, and theory of mind. She presented a number of studies performed in the context of two large projects: the UKRI Trustworthy Autonomous Systems Programme, specifically the Node on Trust, and the EPSRC ORCA Hub for robotic and autonomous systems for remote hazardous environments.

Finally, the keynote speech by Jinho D. Choi (part of the WOCHAT) was entitled: “Alexa Prize and Beyond: the Future of Chatbot”. Dr. Choi started with discussing the challenges in developing a robust dialogue system for open-domain conversations and the lack of the “ground truth” approach for conducting open-domain conversations that would satisfy a wide range of users. He described the approaches that his team used to design the Alexa Prize winning dialogue system. He further pointed out that the subjective nature of the dialogue management evaluation adds another level of difficulty to the design and enhancement of open-domain dialogue systems. In his talk, he illustrated limitations of state-of-the-art transformer-based dialogue systems as well as top-ranked bots from the Alexa Prize Socialbot Grand Challenge. He then introduced the inference-driven dialogue management framework developed at Emory University and discussed its extension to deep learning-based dialogue models. Dr. Choi concluded with real-life applications of open-domain dialogue management for education and healthcare domains.

This year, the IWSDS Organizing and Steering Committee made an important effort in promoting the conference activities among different research, academic, and industrial partners. Thanks to this initiative, the conference received four gold sponsors, two silver sponsors and two local sponsors. Concretely, the gold sponsors were DataBaker (China), a company specialized in AI data acquisition, massive AU databases and one-stop data solutions, MagicData (China), a global AI data service provider, AiShell (China), a technology enterprise focusing on voice data and technical services, and Kriston AI (China), a company focusing on AI-powered customer services and market intelligence. Silver sponsors were Speechocean (China), an AI data resource provider, and Arcadia (Japan), a company doing research and development in the field of human interface such as voice recognition and speech synthesis. Local sponsors were the Chinese and Oriental Languages Information Processing Society (COLIPS) a non-profit professional organization that contributes to advance the research of computer processing and one of the IWSDS traditional supporters, and the National University of Singapore.

Last but not least, as editors and organizers of the conference, we would like to thank the IWSDS Steering Committee and the more than 100 members of the IWSDS 2021 Scientific Committee for their timely and efficient contributions and for completing the review process on time. In addition, we would like to express our gratitude to the members of the Local Committee who highly contributed to the success of the workshop, making it an unforgettable experience for all participants. Thank you all for your logistic support; without it IWSDS 2021 would not have been such a remarkable conference.

With our highest appreciation,

Weybridge, UK — Svetlana Stoyanchev
Stuttgart, Germany — Stefan Ultes
Shenzhen, China — Haizhou Li
April 2022

Contents

Natural Language Understanding

Out-of-Scope Domain and Intent Classification through Hierarchical Joint Modeling . . . 3
Pengfei Liu, Kun Li, and Helen Meng

Segmentation-Based Formulation of Slot Filling Task for Better Generative Modeling . . . 17
Kei Wakabayashi, Johane Takeuchi, and Mikio Nakano

Can We Predict How Challenging Spoken Language Understanding Corpora Are Across Sources, Languages, and Domains? . . . 33
Frederic Bechet, Christian Raymond, Achraf Hamane, Rim Abrougui, Gabriel Marzinotto, and Géraldine Damnati

Personalisation and Generation

Personalized Extractive Summarization with Discourse Structure Constraints Towards Efficient and Coherent Dialog-Based News Delivery . . . 49
Hiroaki Takatsu, Ryota Ando, Hiroshi Honda, Yoichi Matsuyama, and Tetsunori Kobayashi

Empathetic Dialogue Generation with Pre-trained RoBERTa-GPT2 and External Knowledge . . . 67
Ye Liu, Wolfgang Maier, Wolfgang Minker, and Stefan Ultes

Towards Handling Unconstrained User Preferences in Dialogue . . . 83
Suraj Pandey, Svetlana Stoyanchev, and Rama Doddipatla

Jurassic is (Almost) All You Need: Few-Shot Meaning-to-Text Generation for Open-Domain Dialogue . . . 99
Lena Reed, Cecilia Li, Angela Ramirez, Liren Wu, and Marilyn Walker

Spoken and Multimodal Interaction

Comparison of Automatic Speech Recognition Systems 123
Joshua Y. Kim, Chunfeng Liu, Rafael A. Calvo, Kathryn McCabe,
Silas C. R. Taylor, Björn W. Schuller, and Kaihang Wu

Multimodal Dialogue Response Timing Estimation Using Dialogue Context Encoder 133
Ryota Yahagi, Yuya Chiba, Takashi Nose, and Akinori Ito

Eliciting Cooperative Persuasive Dialogue by Multimodal Emotional Robot 143
Sara Asai, Koichiro Yoshino, Seitaro Shinagawa, Sakriani Sakti,
and Satoshi Nakamura

Dialogue Evaluation

Design Guidelines for Developing Systems for Dialogue System Competitions 161
Kazunori Komatani, Ryu Takeda, Keisuke Nakashima,
and Mikio Nakano

Understanding How People Rate Their Conversations 179
Alexandros Papangelis, Nicole Chartier, Pankaj Rajan,
Julia Hirschberg, and Dilek Hakkani-Tur

Dialogue Modelling and Applications

A WoZ Study for an Incremental Proficiency Scoring Interview Agent Eliciting Ratable Samples 193
Mao Saeki, Weronika Demkow, Tetsunori Kobayashi,
and Yoichi Matsuyama

SUPPLE: A Dialogue Management Approach Based on Conversation Patterns 203
Florian Kunneman and Koen Hindriks

Dialogue Management as Graph Transformations 219
Nicholas Thomas Walker, Torbjørn Dahl, and Pierre Lison

Chatbots and Conversational Agent Technologies

Data Collection for Detecting Unwillingness to Answer Questions in Dialogue 231
Kazumi Nagao, Ryuichiro Higashinaka, and Kazuto Ataka

Enhancing Self-disclosure In Open-Domain Dialogue By Candidate Re-ranking 243
Mayank Soni, Benjamin R. Cowan, and Vincent Wade

On the Impact of Self-efficacy on Assessment of User Experience in Customer Service Chatbot Conversations 253
Yuexin Cao, Vicente Ivan Sanchez Carmona, Xiaoyi Liu, Changjian Hu, Neslihan Iskender, André Beyer, Sebastian Möller, and Tim Polzehl

Learning to Ask Specific Questions Naturally in Chat-Oriented Dialogue Systems 263
Sota Horiuchi and Ryuichiro Higashinaka

Fine-Tuning a Pre-trained Transformer-Based Encoder-Decoder Model with User-Generated Question-Answer Pairs to Realize Character-Like Chatbots 277
Koh Mitsuda, Ryuichiro Higashinaka, Hiroaki Sugiyama, Masahiro Mizukami, Tetsuya Kinebuchi, Ryuta Nakamura, Noritake Adachi, and Hidetoshi Kawabata

Investigating the Impact of Pre-trained Language Models on Dialog Evaluation 291
Chen Zhang, Luis Fernando D'Haro, Yiming Chen, Thomas Friedrichs, and Haizhou Li

Natural Language Understanding

Out-of-Scope Domain and Intent Classification through Hierarchical Joint Modeling

Pengfei Liu, Kun Li, and Helen Meng

Abstract User queries for a real-world dialog system may sometimes fall outside the scope of the system's capabilities, but appropriate system responses will enable smooth processing throughout the human-computer interaction. This paper is concerned with the user's intent, and focuses on out-of-scope intent classification in dialog systems. Although user intents are highly correlated with the application domain, few studies have exploited such correlations for intent classification. Rather than developing a two-stage approach that first classifies the domain and then the intent, we propose a hierarchical multi-task learning approach based on a joint model to classify domain and intent simultaneously. Novelties in the proposed approach include (1) sharing supervised out-of-scope signals in joint modeling of domain and intent classification to replace a two-stage pipeline and (2) introducing a hierarchical model that learns the intent and domain representations in the higher and lower layers respectively. Experiments show that the model outperforms existing methods in terms of accuracy, out-of-scope recall, and F_1. Additionally, threshold-based post-processing further improves performance by balancing precision and recall in intent classification.

1 Introduction

Intent classification [1] is one of the core components for NLU in dialog systems, where NLU needs to recognize the domain, intent, and slots of a user query to make an appropriate response. Out-of-scope user queries are inevitable in a task-oriented

P. Liu (✉) · K. Li
SpeechX Limited, Shenzhen, China
e-mail: pfliu@speechx.cn

K. Li
e-mail: kli@speechx.cn

H. Meng
The Chinese University of Hong Kong, Hong Kong, China
e-mail: hmmeng@se.cuhk.edu.hk

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_1

dialog system, because it is difficult, if not impossible, to convey precisely and comprehensively to the system the range of capabilities of the system, especially in terms of the supported intents [2]. However, the problem of out-of-scope intent classification, which aims to find out the queries not belonging to any of the system-supported intents, is not so actively investigated due to lack of publicly available datasets. This problem is similar to out-of-distribution intent classification [3, 4], but poses new challenges since the out-of-scope queries are often similar with the in-scope queries, in terms of topics and/or styles [2]. Existing approaches to out-of-scope intent classification include (1) two-step approaches which first perform binary classification of in-scope versus out-of-scope and in the former case further classify the specific in-scope intent [2, 4]; (2) classifier-based approaches that place out-of-scope query as an additional intent category [2, 5]; and further extend this with (3) a threshold for classification probabilities for each in-scope intent and optionally augmented with an out-of-scope intent [2, 6, 7].

As can be seen, out-of-scope intent classification has not yet been studied from the perspective of joint modeling or multi-task learning. Intent classification in dialog systems is highly dependent on supported domains, such as banking, restaurant, shopping, etc., which means that domain information is useful for recognizing the intent of a user query. Although there has been studies on joint models for the tasks of intent classification and slot filling [8–16], multi-task joint modeling of domains and intents are rarely studied [12, 17, 18]. Furthermore, there still lacks deep understanding of the settings in which multi-task learning may bring significant benefits [19], in other words, how to effectively model the correlation between domain and intent classification in a multi-task learning framework. Remarkably, [19] introduced a hierarchical multi-task learning model for a set of carefully selected semantic tasks, aiming to supervise lower-level tasks (e.g., NER) at the bottom layers and more complex tasks (e.g., relation extraction) at the top layers of the model.

This paper presents a hierarchical joint model for out-of-scope domain and intent classification, where the two tasks of domain and intent classification share the same out-of-scope supervised signals through joint modeling, and a hierarchical structure is introduced in the network to learn the intent representation on top of the domain representation. The major benefits of joint modeling and hierarchical structure are *information sharing and inheritance* between domain and intent classification, which may present advantages over a two-stage pipeline approach of domain classification followed by intent classification. The motivation to introduce the hierarchical structure in the network are two-fold: (1) there are generally a larger number of intents than domains and consequently intent classification may need a more refined semantic understanding of the user's query than domain classification, and (2) intent classification can generally benefit from additional domain-related information.

For example, in the user query of "*My credit card was swallowed by ATM when I tried to withdraw some money. How can I get back my card?*"—it is easy to determine the domain as *banking* based on the words like *credit card* or *ATM*, but requires a model of more refined understanding to determine that the intent is "*report card swallowed*" instead of "*withdraw money*". Besides, knowing that the domain of the query is in banking gives additional information for intent classification. From the perspective of representation learning, the proposed joint model introduces a hierarchical bias whereby the higher layers represent intent information, while the lower layers represent domain information. Such an organization offers a better knowledge representation than a flat structure shared between domain and intent. The major contributions of this paper are:

(1) We propose a novel multi-task joint model for out-of-scope domain and intent classification, which outperforms state-of-the-art methods by a large margin;
(2) We introduce a hierarchical structure in the model to allow for hierarchical representation learning and information inheritance from domain to intent;
(3) We show that a threshold-based post-processing method improves the performance further by balancing precision and recall in out-of-scope intent classification.

2 Related Work

The problem of out-of-scope intent classification is not as actively studied due to lack of publicly available datasets [2, 20–23], but is nonetheless very important especially in real-world dialog systems [24]. The out-of-scope problem encompasses cases where the intent of a user query is not supported by the dialog system but the query is similar in style and or topic to the in-scope queries, as is reflected by the term *out-of-scope* [2, 5]. It also encompasses cases where the user query originates from another dataset and is substantially different from the in-distribution queries, which is literally an *out-of-distribution* problem [3, 4]. For out-of-scope intent classification, [2] introduced a 150-intent dataset for evaluating out-of-scope prediction performance of intent classification systems, and presented BERT-based methods which are however poor at recognizing out-of-scope intents. In line with this formulation, [5] introduced a pre-trained language model named ToD-BERT which is learned from a bunch of task-oriented dialogue datasets and obtained better performance than BERT in terms of accuracy and out-of-scope recall on a downstream intent classification task. By contrast, [3] studied the out-of-distribution problem by forming out-of-distribution examples from another dataset and found that classification with *softmax* distribution probabilities offer good performance on out-of-distribution detection. Similarly, [4] considered the intents excluded from the training set as out-of-distribution intents and adopted a novelty detection algorithm named *local outlier factor* to detect the unknown intents.

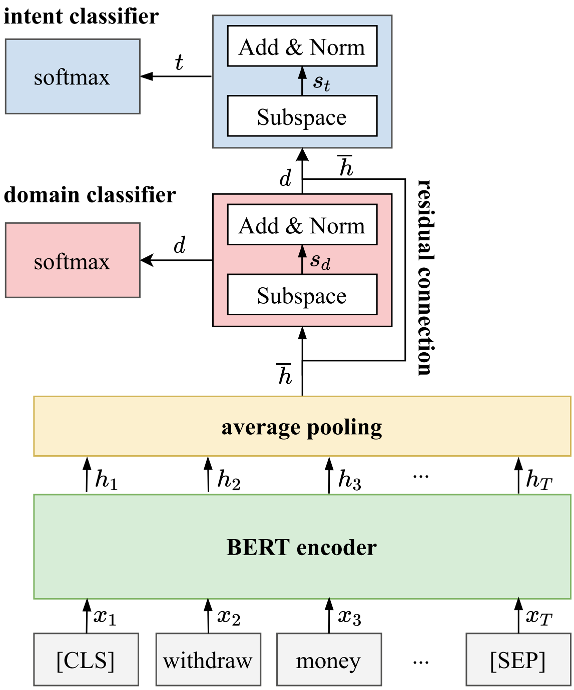
This paper presents a novel approach for out-of-scope intent classification based on joint modeling of domain and intent, together with hierarchical representation fine-tuning from the BERT model for the correlated tasks of domain and intent classification. Hierarchical multi-task learning has been introduced for semantic tasks such as named entity recognition, entity mention detection, coreference resolution, and relation extraction [19]. Similarly, hierarchical modeling has also been applied to syntactic and semantic tasks in chunking, dependency parsing, semantic relatedness, and textual entailment [25]. To the best of our knowledge, the present work is the first to apply hierarchical joint modeling to out-of-scope domain and intent classification.

3 Hierarchical Joint Modeling

The proposed hierarchical joint model, named BERT-Joint, is illustrated in Fig. 1, where a token sequence is fed into a BERT encoder to obtain a sequence of hidden states that are averaged by a pooling operation to obtain the BERT representation $\bar{h}$. The following modules are a domain encoder in red and an intent encoder in blue, as well as the subsequent `softmax` layers for domain and intent classification respectively. Particularly, the intent encoder is fed with the domain representation d to model the hypothesis that intent classification needs additional domain information and requires more layers than domain classification to learn the intent representation t.

BERT Representation. For a given utterance, BERT [26] takes the word sequence $x = (x_1, x_2, ..., x_T)$ as input, and outputs a sequence of hidden states $h = (h_1, h_2, ...,$ $h_T)$ after a few Transformer layers. Following the training schema in pre-trained BERT models, a special token `[CLS]` is added to the start of every sequence

Fig. 1 The architecture of BERT-Joint model, where a pre-trained BERT model is adopted as the encoder, `[CLS]` and `[SEP]` are the two special tokens adding to the start and the end of each sequence respectively



for aggregating the sequence representation and another special token `[SEP]` is appended to the end of every sequence for differentiating the sentences [26]. We used the average pooling vector $\bar{h} = \texttt{average-pooling}(h)$, as the BERT representation of an utterance, which gives slightly better performance than h_1 corresponding to `[CLS]`.

Domain Representation. Given an utterance representation $\bar{h}$ from the BERT encoder, we first obtain a representation subspace s_d from $\bar{h}$ using a non-linear transformation with the weight matrix W_d and the additive bias b_d, and then apply residual connection [27] and layer normalization [28] to obtain the domain representation vector d, as illustrated in Eqs. (1) and (2), which are inspired from the Transformer model [29].

$$s_d = \texttt{ReLU}(W_d\bar{h} + b_d) \tag{1}$$

$$d = \texttt{LayerNorm}(s_d + \bar{h}) \tag{2}$$

Intent Representation. Similarly, we obtain a representation subspace s_t, as in Eq. (3), for intent transformed from the summation of the domain representation d and the BERT representation $\bar{h}$, where W_t is the weight matrix and b_t is the additive bias. We then apply residual connection and layer normalization to get the intent representation t in Eq. (ch1eq4).

$$s_t = \texttt{ReLU}(W_t(d + \bar{h}) + b_t) \tag{3}$$

$$t = \texttt{LayerNorm}(s_t + d) \tag{4}$$

The intent representation t is built on top of the domain representation d to introduce a hierarchical structure in the network. Such a hierarchical structure aims to capture the dependency between a domain and the corresponding intents, and model the hypothesis that additional domain information is useful for intent classification. Besides, we believe that intent classification needs a model with more layers than domain classification due to a larger number of intent classes and the requirement for a deeper understanding of the utterance semantic.

Joint Learning. We learn domain and intent classification jointly using two separate `softmax` layers on top of the corresponding representations, as illustrated in Eqs. (5)–(6), where p^d is the predicted domain distribution and p^t is the predicted intent distribution. We adopt the cross entropy loss for model training. L_d is the loss between the predicted domain distribution p^d and the true domain y^d, where p^d_m means the predicted probability of being domain m, and y^d_m is 1 if the true domain is m else 0. Similarly, L_t measures the loss between the predicted intent distributions p^t and the true intent y^t, where p^t_n means the predicted probability that of being intent n. y^t_n is also a binary indicator which is 1 if the true intent is n and else 0. We optimize domain and intent classification jointly using a linear combination of their corresponding cross entropy loss, as shown in Eq. (9), where the weight λ is also a learnable parameter, jointly learned with the other model parameters.

$$p^d = \texttt{softmax}(W^d d + b^d) \tag{5}$$

$$p^t = \texttt{softmax}(W^t t + b^t) \tag{6}$$

$$L_d = -\sum_{m=1}^{M} y_m^d \log(p_m^d) \tag{7}$$

$$L_t = -\sum_{n=1}^{N} y_n^t \log(p_n^t) \tag{8}$$

$$L = \lambda L_d + (1 - \lambda) L_t \tag{9}$$

Note that the number of domains M is much smaller than the number of intents N in real-world dialog systems, which means that it is easier to determine the domain of an utterance than the intent. As each utterance has both labels of domain and intent, the advantage of joint learning is that the discrimination capability learned by the domain classifier, particularly on out-of-scope user queries, is also shared with the intent classifier by feeding the domain representation to the subsequent intent representation layers.

Threshold-based Post-processing. Since out-of-scope examples are frequently misclassified as in-scope intents at low probabilities, we propose a threshold-based method to post-process the predicted probabilities, and consider an example as out-of-scope if the predicted probability is below the pre-specified threshold τ, (i.e., $p^t < \tau$ for intent classification). It is interesting to observe that setting a threshold value generally improves both in-scope and out-of-scope accuracy. More importantly, the threshold-based post-processing method provides an effective way to balance *precision* and *recall* for out-of-scope intent classification.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate the proposed model using the OOS dataset [2], which consists of 150 intents across 10 domains and a number of out-of-scope examples belonging to none of the domains or intents.[1] The dataset is different from conventional intent datasets in the sense that it focuses on out-of-scope intent classification. The task is particularly challenging since the out-of-scope examples are similar in topics or styles to the in-scope examples but are not within any of the 150 in-scope intents. There are three variants of the OOS dataset, namely *Small*, *Imbalanced*, and *OOS+*, where *Small* has the smallest number of total examples, and *OOS+* has the largest number of out-of-scope examples. In contrast, *Imbalanced* has the imbalanced number of in-scope examples. The number of examples in each variant of the OOS dataset is

[1] https://github.com/clinc/oos-eval.

Table 1 Number of examples in variants of the OOS dataset

		Full	Small	Imbalanced	OOS+
Train	Total Examples	15100	7600	10625	15250
	#Out-of-scope Examples	100	100	100	250
	#Examples per In-scope Intent	100	50	25, 50, 75, 100	100
Valid	Total Examples	3100	3100	3100	3100
	#Out-of-scope Examples	100	100	100	100
	#Examples per In-scope Intent	20	20	20	20
Test	Total Examples	5500	5500	5500	5500
	#Out-of-scope Examples	1000	1000	1000	1000
	#Examples per In-scope Intent	30	30	30	30

shown in Table 1. Note that all the variants have the same test set which has 1000 out-of-scope examples and 150 * 30 in-scope examples.

Metrics. We adopt *accuracy* as the metric for evaluating the overall accuracy (all) on all the examples, and the in-scope accuracy (in) on the in-scope examples, for the OOS test set. For out-of-scope examples, we report the metrics of precision (P), recall (R), and F_1.

Settings. We adopt the pre-trained BERT model of *bert-base-uncased* for an initial utterance representation. For fine-tuning, we used the AdamW [30] optimizer and set the proportion of warm-up steps as 0.1, the learning rate as `4E-5`. The maximum number of epochs is set as 10 on all the experiments except on OOS+, which has the largest number of training examples and obtains the best performance using 5 epochs. We adopted early stopping on condition that the intent classification accuracy does not improve for 3 epochs. We implemented the models using the PyTorch framework [31] and kept the random seed fixed on all the experiments for reproducible results. We released the code as open source at `https://github.com/ppfliu/oos-intent-recognition`.

4.2 Results and Discussion

Comparisons with Existing Methods. Table 2 presents the experimental results from [2] including the methods of FastText, SVM, CNN, and BERT, and [5] covering GPT2, DialogGPT, and ToD-BERT, as well our methods of BERT and BERT-Joint. It can be seen that the proposed BERT-Joint model obtains the best performance in terms of overall accuracy, and out-of-scope precision (P), recall (R) and F_1, and is further outperformed by applying a threshold-based post-processing method.

Error Analysis. We analyzed a few examples from the OOS test set, which are misclassified by either BERT or BERT-Joint, as shown in Table 3. Although Examples 1–4 are out-of-scope, users may naturally ask these questions as they do not

Table 2 Performance comparisons with existing methods for intent classification on the OOS test dataset (Full)

	Model	Accuracy		P	R	F_1
		All	In	Out	Out	Out
Larson et al. [2]	**FastText**	–	0.890	–	0.097	–
	SVM	–	0.910	–	0.145	–
	CNN	–	0.912	–	0.189	–
	BERT	–	**0.969**	–	0.403	–
Wu et al. [5]	**GPT2**	0.830	0.941	–	0.320	–
	DialoGPT	0.839	0.955	–	0.321	–
	BERT	0.849	0.958	–	0.356	–
	ToD-BERT-mlm	0.859	0.961	–	0.463	-
	ToD-BERT-jnt	0.866	0.962	–	0.436	–
This Work	**BERT**	0.855	0.962	0.981	0.370	0.537
	BERT-Joint	**0.876**	0.964	**0.984**	**0.484**	**0.649**
	+Threshold	**0.920**	0.955	0.902	**0.761**	**0.825**

precisely know about the system's knowledge scope and capabilities. BERT-Joint makes correct predictions for Examples 3 and 4 but fails to reject Examples 1 and 2. Examples 5–7 are quite challenging, as they need a deeper semantic understanding of the sentences such as semantic inference (Example 6), discourse structure (Example 7). BERT-Joint classifies Examples 5 and 6 correctly on both domain and intent but not on Example 7, which actually consists of two sentences and the second sentence delivers the real intent. We notice that there are some annotation errors on *intent* in Examples 8-10. However, the predicted intents by both BERT and BERT-Joint are reasonable.

Performance on Dataset Variants. We further verified the performance of BERT-Joint on the OOS variants, as shown in Table 4. We observe that BERT-Joint consistently outperforms BERT on all the dataset variants in terms of overall accuracy, out-of-scope recall, and F_1. Particularly, it improves the F_1 score of out-of-scope examples by an absolute increase of more than **10%** on *Full* and *Small*, **3%** on *Imbalanced* and **6%** on *OOS+*.

Effect of Hierarchical Structure. The proposed approach of joint modeling of domain and intent is flexible to support various *flat* or *hierarchical* model structures. Here, *flat* means the domain representation and the intent representation are put side by side in the network, such as $F(\bar{h}; \bar{h})$ which directly uses the same BERT output $\bar{h}$ for domain and intent classification respectively, and $F(s_d; s_t)$ which adopts the subspace vectors s_d and s_t for the corresponding domain and intent classification. For *hierarchical* model structures, we consider both $H(s_t \rightarrow s_d)$ and $H(s_d \rightarrow s_t)$. The former structure means that we get the intent representation s_t first and then feed it to the subsequent layers to get the domain representation s_d, while the latter first learns the domain representation s_d which is then fed to the subsequent layers to get the intent representation s_t.

Table 5 presents the performance comparisons between BERT and the variants of BERT-Joint covering the four different model structures. We have the following observations:

(1) All variants of BERT-Joint outperform the BERT model, which is not surprising since BERT-Joint takes advantage of additional domain information for intent classification;
(2) The flat structure of $F(s_d; s_t)$ consistently outperforms $F(\bar{h}; \bar{h})$ on all the datasets, which may indicate that s_d and s_t can capture effective features from the BERT representation $\bar{h}$ for the corresponding domain and intent classification;
(3) The hierarchical structures generally outperform flat structures in terms of accuracy (all) on all the datasets except OOS+ where the structure of $F(s_d; s_t)$ obtains the best accuracy (all), as well as out-of-scope recall and F_1;
(4) BERT-Joint is particularly effective in dealing with out-of-scope intent classification. For example, $H(s_d \rightarrow s_t)$ outperforms BERT in terms of F_1 by an absolute increase of more than **10%** on both *Full* and *Small*. This may be attributed to the domain classification task which also needs to learn how to classify the out-of-domain examples. Such capability is inherited by the intent classifier through feeding the domain representation to the subsequent intent layers and thus the out-of-scope intent classification performance is improved further.

Threshold-based Post-processing. Figure 2 presents the performance comparisons on the validation dataset (V) and and the testing dataset (T) of OOS with $\tau \in \{0.1, \ldots, 0.9\}$. It is clear to see that the threshold value τ affects all the metrics on both V and T, and thus the threshold-based post-processing method provides an effective way to *balance* precision and recall for out-of-scope intent classification.

Table 3 Examples from the OOS test set for error analysis, where the labels in red are misclassified. Note that BERT only predicts the intent whereas BERT-Joint predicts both the domain and intent simultaneously

(a) Testing examples

ID	Example	Domain
1	Give me the weather forecast for today	oos
2	How much data does my phone have left this month	oos
3	How many homeless people are there	oos
4	How do i learn more about linguistics	oos
5	I would like to know my vacation days balance	work
6	Does bank of America give credit cards to people like me	credit_cards
7	I'm trying to raise my credit score can you tell me what it is now	credit_cards
8	Someone used my chase card without my authorization	credit_cards
9	Can you call the help desk line for my credit card company	credit_cards
10	How can i request a new credit card	credit_cards

(continued)

Table 3 (continued)

(b) Error analysis

ID	Ground truth	BERT	BERT-Joint	
	Intent	**Intent**	**Domain**	**Intent**
1	oos	weather	utility	weather
2	oos	balance	utility	find_phone
3	oos	traffic	oos	oos
4	oos	translate	oos	oos
5	pto_balance	balance	work	pto_balance
6	new_card	international_fees	credit_cards	new_card
7	credit_score	improve_credit_score	credit_cards	improve_credit_score
8	report_lost_card	report_fraud	banking	report_fraud
9	replacement_card_duration	make_call	credit_cards	make_call
10	replacement_card_duration	new_card	credit_cards	new_card

Table 4 Performance comparisons between BERT and BERT-Joint using different dataset variants of OOS

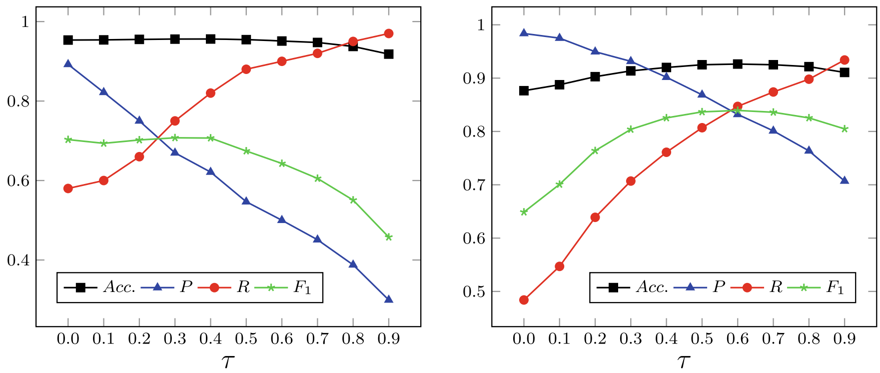
	Model	Accuracy		P	R	F_1
		All	**In**	**Out**	**Out**	**Out**
Full	BERT	0.855	0.962	0.981	0.370	0.537
	BERT-Joint	0.876	0.964	**0.984**	**0.484**	**0.649**
Small	BERT	0.845	0.953	0.975	0.357	0.523
	BERT-Joint	0.865	0.954	**0.981**	**0.464**	**0.630**
Imbalanced	BERT	0.855	0.952	**0.981**	0.423	0.591
	BERT-Joint	0.869	0.960	0.979	**0.462**	**0.628**
OOS+	BERT	0.882	0.959	**0.983**	0.536	0.694
	BERT-Joint	0.897	0.959	0.969	**0.621**	**0.757**

In Fig. 2a, with the increase of τ, the accuracy (*Acc.*) improves first and then drops when $\tau > 0.4$, since the low-probability ($< \tau$) in-scope examples are now misclassified as out-of-scope. As illustrated in Fig. 2a, b for out-of-scope intent classification, R keeps increasing at the expense of decreasing in P, whereas the highest F_1 is obtained at $\tau = 0.3$ on V and at $\tau = 0.6$ on T. Note that T has a much larger number of out-of-scope examples than V and thus requires a larger τ for better recall.

Representation Visualization. To deepen our understanding of the out-of-scope classification problem, we further visualized the domain and intent representations from the test set of OOS using t-SNE [32], which visualizes high-dimensional vectors in a two or three-dimensional map. As illustrated in Fig. 3, each color represents a domain (a) or an intent (b). The 10 in-scope domains are well separated in Fig. 3a, so does the 150 in-scope intents in Fig. 3b. Note that some points are overlapped in Figure 3 due to too many examples and domains/intents, best viewed when enlarged.

Table 5 Performance comparisons between BERT and BERT-Joint with different structures (F: *Flat*, H: *Hierarchical*)

	Model	Structure	Accuracy		P	R	F$_1$
			All	In	Out	Out	Out
Full	BERT	–	0.8545	0.9622	0.9814	0.3700	0.5374
	BERT-Joint	$F(\bar{h};\bar{h})$	0.8689	0.9622	0.9825	0.4490	0.6163
		$F(s_d;s_t)$	0.8727	0.9604	**0.9856**	0.4780	0.6438
		$H(s_t \rightarrow s_d)$	0.8715	0.9611	0.9770	0.4680	0.6329
		$H(s_d \rightarrow s_t)$	**0.8764**	**0.9636**	0.9837	**0.4840**	**0.6488**
Small	BERT	–	0.8447	0.9531	0.9754	0.3570	0.5227
	BERT-Joint	$F(\bar{h};\bar{h})$	0.8529	0.9460	0.9731	0.4340	0.6003
		$F(s_d;s_t)$	0.8538	0.9500	0.9768	0.4210	0.5884
		$H(s_t \rightarrow s_d)$	0.8651	**0.9573**	**0.9890**	0.4500	0.6186
		$H(s_d \rightarrow s_t)$	**0.8653**	0.9544	0.9810	**0.4640**	**0.6300**
Imbalanced	BERT	–	0.8555	0.9516	0.9814	0.4230	0.5912
	BERT-Joint	$F(\bar{h};\bar{h})$	0.8569	0.9544	**0.9882**	0.4180	0.5875
		$F(s_d;s_t)$	0.8673	0.9536	0.9796	**0.4790**	**0.6434**
		$H(s_t \rightarrow s_d)$	0.8689	0.9587	0.9873	0.4650	0.6322
		$H(s_d \rightarrow s_t)$	**0.8693**	**0.9598**	0.9788	0.4620	0.6277
OOS+	BERT	–	0.8820	0.9589	0.9835	0.5360	0.6939
	BERT-Joint	$F(\bar{h};\bar{h})$	0.9005	0.9600	0.9649	0.6330	0.7645
		$F(s_d;s_t)$	**0.9053**	0.9609	**0.9762**	**0.6550**	**0.7840**
		$H(s_t \rightarrow s_d)$	0.8973	0.9611	0.9744	0.6100	0.7503
		$H(s_d \rightarrow s_t)$	0.8985	**0.9613**	**0.9762**	0.6160	0.7554



(a) validation set (V) (b) testing set (T)

Fig. 2 Overall accuracy (*Acc.*) and out-of-scope intent classification performance (P, R, and F_1) on V and T

The out-of-scope examples are mainly located in the same *blue* cluster in both (a) and (b), but quite a few out-of-scope examples are distributed across different domains or intents. This explains why it is difficult to classify the out-of-scope examples, and why the simple threshold-based method gives better performance on out-of-scope intent classification.

5 Conclusion

This paper presents a novel hierarchical joint model based on BERT for out-of-scope domain and intent classification. The proposed model allows sharing of supervised signals between both classification tasks and introduces a structural bias to enable hierarchical representation learning from the pre-trained BERT representations. We empirically show that the model outperforms existing methods in terms of accuracy as well as out-of-scope recall and F_1 by a large margin on all the variants of the OOS dataset. These observations serve to illustrate the effectiveness of joint modeling and hierarchical structure of the model particularly in out-of-scope intent classification. Furthermore, we show that a threshold-based post-processing method improves the performance further and allows to effectively balance precision and recall in out-of-scope intent classification.

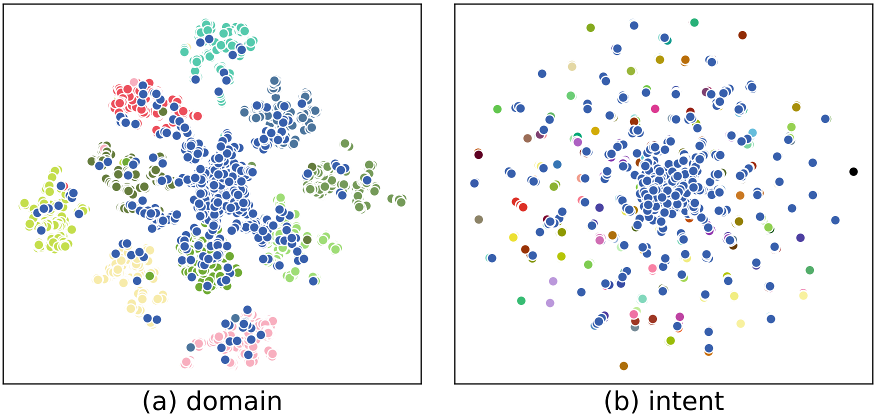


Fig. 3 Visualization of domain and intent representations using t-SNE, where each color indicates a domain (intent) and the out-of-scope examples are colored in blue, best viewed when enlarged

References

1. Liu J, Li Y, Lin M (2019) Review of intent detection methods in the human-machine dialogue system. J Phys: Conf Ser 1267:012059. IOP Publishing
2. Larson S, Mahendran A, Peper JJ, Clarke C, Lee A, Hill P, Kummerfeld JK, Leach K, Laurenzano MA, Tang L, Mars J (2019) An evaluation dataset for intent classification and out-of-scope prediction. 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). https://www.aclweb.org/anthology/D19-1131
3. Hendrycks D, Gimpel K (2017) A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: Proceedings of international conference on learning representations
4. Lin TE, Xu H (2019) Deep unknown intent detection with margin loss. In: Proceedings of the 57th annual meeting of the association for computational linguistics, Association for Computational Linguistics, Florence, Italy, pp 5491–5496. https://www.aclweb.org/anthology/P19-1548
5. Wu CS, Hoi S, Socher R, Xiong C (2020) Tod-bert: Pre-trained natural language understanding for task-oriented dialogues. arXiv:2004.06871
6. Meng HM, Lam W, Wai C (1999) To believe is to understand. In: Proceedings of the 6th European conference on speech communication and technology
7. Lin TE, Xu H (2019) A post-processing method for detecting unknown intent of dialogue system via pre-trained deep neural network classifier. Knowl-Based Syst 186:104979
8. Xu P, Sarikaya R (2013) Convolutional neural network based triangular CRF for joint intent detection and slot filling. In: 2013 IEEE workshop on automatic speech recognition and understanding. IEEE, pp 78–83
9. Guo D, Tur G, Yih Wt, Zweig G (2014) Joint semantic utterance classification and slot filling with recursive neural networks. In: 2014 IEEE spoken language technology workshop (SLT). IEEE, pp 554–559
10. Liu B, Lane I (2016) Attention-based recurrent neural network models for joint intent detection and slot filling. In: Interspeech 2016. pp 685–689. http://dx.doi.org/10.21437/Interspeech.2016-1352
11. Zhang X, Wang H (2016) A joint model of intent determination and slot filling for spoken language understanding. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence, pp 2993–2999
12. Kim YB, Lee S, Stratos K (2017) Onenet: joint domain, intent, slot prediction for spoken language understanding. In: 2017 IEEE automatic speech recognition and understanding workshop (ASRU). IEEE, pp 547–553
13. Goo CW, Gao G, Hsu YK, Huo CL, Chen TC, Hsu KW, Chen YN (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, Vol 2 (Short Papers), pp 753–757
14. Wang Y, Shen Y, Jin H (2018) A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In: NAACL-HLT (2)
15. Chen Q, Zhuo Z, Wang W (2019) Bert for joint intent classification and slot filling. arXiv:1902.10909
16. Zhang C, Li Y, Du N, Fan W, Yu P (2019) Joint slot filling and intent detection via capsule neural networks. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy, pp 5259–5267. https://www.aclweb.org/anthology/P19-1519
17. Hakkani-Tür D, Tur G, Celikyilmaz A, Chen YN, Gao J, Deng L, Wang YY (2016) Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. Interspeech 2016:715–719
18. Kim JK, Kim YB (2018) Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisficing false acceptance rates. arXiv:1807.00072

19. Sanh V, Wolf T, Ruder S (2019) A hierarchical multi-task approach for learning embeddings from semantic tasks. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 6949–6956
20. Braun D, Mendez AH, Matthes F, Langen M (2017) Evaluating natural language understanding services for conversational question answering systems. In: Proceedings of the 18th annual SIGdial meeting on discourse and dialogue, pp 174–185
21. Coucke A, Saade A, Ball A, Bluche T, Caulier A, Leroy D, Doumouro C, Gisselbrecht T, Caltagirone F, Lavril T, et al (2018) Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv:1805.10190
22. Liu X, Eshghi A, Swietojanski P, Rieser V (2019) Benchmarking natural language understanding services for building conversational agents. In: Proceedings of the tenth international workshop on spoken dialogue systems technology (IWSDS)
23. Yilmaz EH, Toraman C (2020) KLOOS: KL divergence-based out-of-scope intent detection in human-to-machine conversations. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp 2105–2108
24. Aliannejadi M, Zamani H, Crestani F, Croft WB (2019) Asking clarifying questions in open-domain information-seeking conversations. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp 475–484
25. Hashimoto K, Xiong C, Tsuruoka Y, Socher R (2017) A joint many-task model: Growing a neural network for multiple NLP tasks. In: Proceedings of the 2017 conference on empirical methods in natural language processing. Association for Computational Linguistics, Copenhagen, Denmark, pp 1923–1933. https://www.aclweb.org/anthology/D17-1206
26. Devlin J, Chang MW, Lee K, Toutanova K (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, Vol 1 (Long and Short Papers). pp 4171–4186
27. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
28. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv:1607.06450
29. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
30. Loshchilov I, Hutter F (2017) Decoupled weight decay regularization. arXiv:1711.05101
31. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al (2019) PyTorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems, pp 8026–8037
32. Maaten Lvd, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(Nov):2579–2605

Segmentation-Based Formulation of Slot Filling Task for Better Generative Modeling

Kei Wakabayashi, Johane Takeuchi, and Mikio Nakano

Abstract Slot filling is a fundamental task in spoken language understanding that is usually formulated as a sequence labeling problem and solved using discriminative models such as conditional random fields and recurrent neural networks. One of the weak points of this discriminative approach is robustness against incomplete annotations. For obtaining a more robust method, this paper leverages an overlooked property of slot filling tasks: Non-slot parts of utterance follow a specific pattern depending on the user's intent. To this end, we propose a generative model that estimates the underlying pattern of utterances based on a segmentation-based formulation of slot-filling tasks. The proposed method adopts nonparametric Bayesian models that enjoy the flexibility of the phrase distribution modeling brought by the new formulation. The experimental result demonstrates that the proposed method performs better in a situation that the training data with incomplete annotations in comparison to the BiLSTM-CRF and HMM.

1 Introduction

Slot filling is a task that estimates the speaker's intent in the form of slot representation. For example, the utterance "Remind me to call John at 10 to 9 am tomorrow" contains two pieces of information that the system is required to extract for setting a reminder; {**time**: "10 to 9 am tomorrow"} and {**subject**: "call John"}. We use the

K. Wakabayashi (✉)
University of Tsukuba, Tsukuba, Japan
e-mail: kwakaba@slis.tsukuba.ac.jp

J. Takeuchi
Honda Research Institute Japan Co., Ltd., Saitama, Japan
e-mail: johane.takeuchi@jp.honda-ri.com

M. Nakano
Honda Research Institute Japan Co., Ltd. (Currently with C4A Research Institute, Inc.), Saitama, Japan
e-mail: mikio.nakano@c4a.jp

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_2

O	O	O	B-subject	I-subject	O	B-time	I-time	I-time	I-time	I-time
remind	me	to	call	john	at	10	to	9	am	tomorrow

Fig. 1 Sequence labeling formulation of slot filling

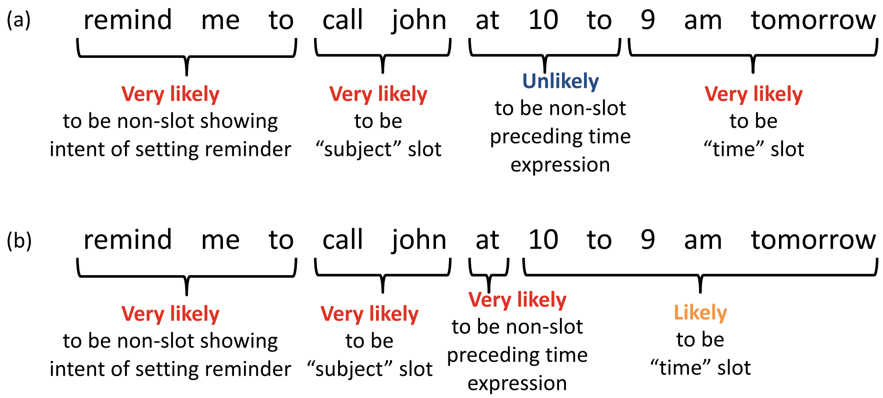


Fig. 2 The segmentation-based approach. Detecting slot parts are formulated as a task that finds the best partition in terms of the joint probability of both the slot and non-slot phrases. The proposed method prefers the segmentation in case (**b**) than in (**a**) because it gives a higher joint likelihood

term *slot* to refer to variables such as **time** and **subject** that are filled by substrings in an utterance.

Slot filling is usually formulated as a sequence labeling task with IOB tagging scheme, which is generally used in phrase extraction tasks such as named entity recognition [10]. Figure 1 shows an example of the sequence labels. On the basis of this formulation, the existing studies have applied discriminative models including conditional random fields (CRFs) [18] and neural networks [13, 17, 29].

One of the weak points in the supervised learning approach is the robustness against incomplete annotations [7]. In practice, obtaining high-quality annotation for phrase extraction is expensive and not scalable [24]. When there is a missing annotation on a substring, the model will be trained to assign O tags for the substrings since we have no way to know if it is missing or truly a non-slot part.

In this paper, we explore an approach that leverages an overlooked characterization of slot filling that is not shared with other phrase extraction tasks: Non-slot substrings also follow a specific pattern. For example, when a user has an intent of setting a reminder, her utterance likely starts with "remind me" to show her intent. On this idea, the slot filling can be formulated as a task that splits an utterance into segments and estimates the role of each segment as Fig. 2 shows.

To this end, we propose a generative model that allows us to induce the roles of non-slot substrings in a similar way to unsupervised grammar induction methods [8, 16]. The proposed model adopts Pitman-Yor Chinese restaurant processes (PYCRPs), which reflects the power-law property inhered in natural language phrases [4], for defining phrase probabilities. Instead of the word-by-word generative process assumed by conventional models such as hidden Markov models (HMMs), the proposed generative model fully enjoys the flexibility of the phrase distribution modeling brought by the new formulation. In the experiment, we will show that the proposed model is capable of capturing the latent structure of utterances, and therefore, more robust against missing annotations in training data.

The contribution of this paper is summarized as follows:

- We propose a new representation of slot filling task, particularly for a better generative modeling. We also propose a Bayesian model that leverages the flexibility of modeling provided by this formulation by adopting a nonparametric Bayesian model for phrase distribution.
- We empirically show that the generative methods on this formulation are more robust than neural networks when the annotation is highly incomplete. We also show the proposed model has good interpretability thanks to this formulation based on segment-wise pattern recognition.

2 Related Work

Although the sequence labeling formulation is the dominant approach in recent years, there have been different ways to formalize slot filling tasks, including formulation as machine translation task [12], decoding problem with finite-state transducers [3, 9], parsing task with context-free grammar [22]. When we can fix the set of possible values that can be put in each slot, we can formulate a slot filling task as a value-based classification task [6]. Wakabayashi et al. [27] extend the classification-based slot filling method by considering the likelihood of non-slot phrases. To define the likelihood of phrases, they proposed a probabilistic model based on nonparametric Bayesian models that are similar to ours. However, their formulation is classification-based; therefore, it requires candidates of slot filling output fed by the N-best results of another discriminative model such as CRF.

While the discriminative modeling based on neural networks is extensively studied [18, 30], the generative approach still has an advantage when we have incomplete and noisy annotated sentences as training data. When we use crowdsourcing to obtain labeled sentence, we need to handle the training data that includes erroneous annotation [15, 19]. In this situation, HMM-based generative models achieve a better accuracy compared to methods that are based on discriminative models [14]. Simpson et al. [23] further improve the accuracy by integrating prior distribution into the generative models. These methods treat the true labels as latent variables and estimate them in a Bayesian estimation manner in the representation of word-by-word sequence labeling formulation. Applying the proposed segmentation-based formulation to these models for crowdsourced annotations will be a subject of future work.

The proposed method can be viewed as a kind of grammar induction [8, 16] since the method attempts to induce the roles of non-slot parts in an unsupervised manner. From this viewpoint, the proposed method is characterized as follows: (i) Flat (non-hierarchical) latent structure is assumed. (ii) Partial supervision on slot part is available instead of inducing fully model-driven grammatical units. Ponvert et al. [16] proposed an unsupervised shallow parsing (i.e., chunking) method that induces labeled segmentation, which is compatible with the characteristics (i) of ours, In comparison to Ponvert's method, our proposed method uses nonparametric Bayesian language modeling to handle longer phrases than grammatical units with a partially supervised training algorithm. Some language models developed for unsupervised morphological analysis [4, 26] adopt the Pitman-Yor process, which inspires our model. However, these models are designed to find the morphological units by embedding n-gram probability over segments. Our proposed method is designed to capture patterns of non-slot phrases supposing that partial supervision on slot part is available, which is novel even in the context of grammar induction.

3 Segmentation-Based Formulation of Slot Filling Task

The proposed formulation regards a slot filling task as a labeled segmentation of a given sentence. For example, the case (b) in Fig. 2 divides the sentence into four segments, "remind me to", "call john", "at" and "10 to 9 am tomorrow", and attaches labels "non-slot 2", "subject slot", "non-slot 4" and "time slot", respectively. Let $x_{1:T} = x_1, \ldots, x_T$ be a sequence of tokens[1] and $b_{1:K} = b_1, \ldots, b_K$ be indices of the last token of each segment where $b_k < b_{k+1}$. The segmentation in Fig. 2b is represented as $b_1 = 3$, $b_2 = 5$, $b_3 = 6$, and $b_4 = 11$. We denote the sequence of the segment labels by $y_{1:K} = y_1, \ldots y_K$. The subsequence of tokens that represents the k-th segment is denoted by $s_k = x_{b_{k-1}+1:b_k}$. b_K equals T because the last segment ends with the last token. We also define $b_0 = 0$ for convenience.

For slot filling tasks, a set of slots $\mathscr{X}$ (e.g., {**time**, **subject**}) and a set of training data are given. The instance of the training data is a pair of a sentence and a slot annotation. For example, the annotation for the sentence in Fig. 2 is { **subject**: "call john", **time**: "10 to 9 am tomorrow" }. In the proposed method, we assume that the non-slot parts also have latent segments. For these segments, we assign a non-slot label that reflects a pattern of non-slot parts. We denote a set of the non-slot labels by $\mathscr{U}$ and assume that each non-slot label in $\mathscr{U}$ is associated with its particular phrase distribution. Consequently, the set of labels is defined as $\mathscr{Y} = \mathscr{X} \cup \mathscr{U}$. We emphasize that the training data only have slot annotations so that the non-slot labels are latent variables. In the proposed method, the non-slot labels are estimated by Gibbs sampling as we present later.

[1] In the experiment, we used word as a token for English and character as a token for Japanese.

3.1 Definition of Generative Models

We consider a generative model in the following form.

$$p(x_{1:T}, y_{1:K}, b_{1:K}) = p(x_{1:T}, b_{1:K}|y_{1:K})p(y_{1:K}) \tag{1}$$

We assume that $p(y_{1:K})$ follows a Markov model with a parameter $\Theta = \theta_1, \ldots, \theta_{|\mathscr{Y}|}$.

$$p(y_{1:K}) = \prod_{k=1}^{K} p_{cat}(y_k|\theta_{y_{k-1}}) \tag{2}$$

$p_{cat}(y_k|\theta_{y_{k-1}})$ is a transition probability that follows a categorical distribution with parameter $\theta_{y_{k-1}}$. θ_y follows a Dirichlet distribution of a hyperparameter γ. The joint distribution $p(x_{1:T}, b_{1:K}|y_{1:K})$ is assumed to be decomposable into segments.

$$p(x_{1:T}, b_{1:K}|y_{1:K}) = \prod_{k=1}^{K} \mathscr{P}_{y_k}(x_{b_{k-1}+1:b_k}) \tag{3}$$

Given the label y_k, a sequence of characters in the k-th segment is generated by a *slot model* $\mathscr{P}_y$, which we present in the following subsections. In the slot model, we represent a phrase as a sequence of characters $s = c_1, \ldots, c_L$ where c_l is a character,[2] instead of the sentence-dependent representation $x_{b_{k-1}+1:b_k}$ [31]. $\mathscr{P}_y$ is a probabilistic model over the infinite set of token sequences $\mathscr{V}$ represented below.

$$\mathscr{V} = \{c_1, \ldots, c_L|c_l \in \mathscr{C}, L \geq 0\}$$

where $\mathscr{C}$ is the set of characters that potentially appear in the input sentences, including the whitespace character. We call an element of $\mathscr{V}$ as a phrase.

3.1.1 N-Gram Slot Model

One of the simplest ways to define a distribution on $\mathscr{V}$ is to adopt an N-gram model. We also explicitly formulate the probability of the phrase length to define a distribution such that the sum of the probability is 1 over $\mathscr{V}$ [31]. The probability of phrase $s = c_1, \ldots, c_L$ is defined as the product of the n-gram probability of the character sequence and the probability that the phrase length is L.

[2] We can formulate the language models for phrases based on token sequence representation, but we prefer the character sequence modeling because the model can get more flexibility. This choice does not affect the overall framework of the proposed method.

$$p_{ngsm}(s = c_1, \ldots, c_L|\psi, \xi) = p_{cat}(L|\psi) \prod_{l=1}^{L} p_{cat}(c_l|\xi_{c_{l-n+1:l-1}})$$

$p_{cat}(L|\psi)$ is defined as a L_{max}-dimensional categorical distribution. $p_{cat}(c_l|\xi_{c_{l-n+1:l-1}})$ is a categorical distribution of a character depending on the n-gram context $c_{l-n+1:l-1}$. Dirichlet distributions with parameter η_1, η_2 are assumed as the priors of ψ and ξ, respectively. We call this model n-gram slot model (NGSM).

3.1.2 Pitman-Yor Process Slot Model

In slot filling tasks, users tend to use specific common phrases. For example, the **time** slot only takes the expressions of time and date, and as a result, a small number of expressions are expected to be used repeatedly. To reflect this observation, we present Pitman-Yor process slot model (PYPSM) for modeling the phrase distribution. PYPSM adopts a Pitman-Yor Chinese restaurant process (PYCRP) that entails power-law distributions over $\mathcal{V}$ [11, 20].

PYPSM is a model that generates phrases $s_1, \ldots, s_N$ where $s_i = c_{i1}, \ldots, c_{iL_i}$ based on the generative process shown in Fig. 3 **(Left)**. The PYPSM has two latent variables; $\boldsymbol{\phi} = \{\phi_1, \ldots, \phi_M\}$ ($\phi_m \in \mathcal{V}$) that is a series of phrases that have been seen before[3] and $a_{1:N} = a_1, \ldots, a_N$ ($1 \le a_i \le M$) that associates each observation s_i to one of the elements of $\boldsymbol{\phi}$. Initially, $\boldsymbol{\phi}$ is empty and $M = 0$. For each step to generate a phrase s_i, the process draws a_i depending on $a_{1:i-1}$ from the following distribution.

$$p(a_i = m|a_{1:i-1}) = \begin{cases} \frac{n_m - \beta}{i-1+\alpha} & 1 \le m \le M \\ \frac{M\beta+\alpha}{i-1+\alpha} & m = M+1 \end{cases} \tag{4}$$

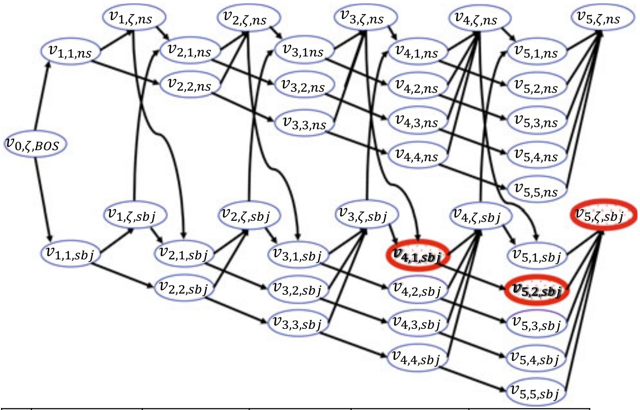
n_m is the frequency of m in $a_{1:N}$, i.e., $n_m = \sum_{i=1}^{N} \delta_{a_i=m}$ where δ_p is an indicator function that returns 1 if the proposition p is true and 0 otherwise. α and β are hyper-parameters of PYCRP that controls the strength of the power-law property. If $a_i = M + 1$ is drawn from the distribution above, the process generates a new phrase for s_i from the NGSM p_{ngsm}, which is known as base distribution [31]. If $a_i \le M$, the process generates s_i as the same phrase generated before, ϕ_{a_i}.

The PYPSM assigns a large probability to "memorized" phrases in ϕ but does not fix a set of possible phrases predefined in advance, which matches the tendency of slot filling tasks. When all the latent variables $a_{1:N}$ and $\phi_{1:M}$ generated up to the N-th observation are given, the predictive distribution of the next phrase s_{N+1} can be described as follows by marginalizing out a_{N+1} and ϕ_{M+1}.

$$p_{pypsm}(s_{N+1}|a_{1:N}, \phi_{1:M}, \psi, \xi) = \sum_{m=1}^{M} \frac{n_m - \beta}{N+\alpha} \delta_{\phi_m = s_{N+1}} + \frac{M\beta+\alpha}{N+\alpha} p_{ngsm}(s_{N+1}|\psi, \xi) \tag{5}$$

[3] In contrast to the major usage of CRP that constitutes an infinite mixture model [25], ϕ_{a_i} is not a parameter for another distribution but an observable phrase ($s_i = \phi_{a_i}$).

$M \leftarrow 0$
For $i = 1$ to N:
  Draw a_i given $a_{1:i-1}$ from Eq. (4)
  If $a_i = M + 1$:
    Draw $\phi_{M+1} \sim p_{ngsm}$
    $M \leftarrow M + 1$
  $s_i \leftarrow \phi_{a_i}$



t	1	2	3	4	5
x_t	remind	me	to	call	john

Fig. 3 **(Left)** Generative process of PYPSM. **(Right)** Lattice for forward-backward sampling and Viterbi algorithm on the segmentation-based formulation. The emphasized nodes correspond to a slot annotation in training data (**subject**: "call john") that is available in the training phase

This distribution predicts the next phrase as either (i) a phrase that has been observed at least once in $s_{1:N}$ for probability $\frac{N-M\beta}{N+\alpha}$, or (ii) a phrase that is newly generated by the NGSM for probability $\frac{M\beta+\alpha}{N+\alpha}$. We use p_{pypsm} as the language model $\mathscr{P}_y$.

3.2 Training of PYPSMs by Collapsed Gibbs Sampling

The annotation provided in training data consists of multiple pairs of (slot, value) to be extracted from a given utterance. For example, the annotation for the sentence in Fig. 2 is { **subject**: "call john", **time**: "10 to 9 am tomorrow" }. This supervision partially determines y and b in the proposed model, but the boundaries and the labels for non-slot parts are still hidden. In this paper, we present a collapsed Gibbs sampling method to make an inference on these latent variables.

Let $X = x^{(1)}_{1:T_i}, \ldots, x^{(N)}_{1:T_N}$ be a set of training sentences and $Z = z^{(1)}, \ldots, z^{(N)}$ be the corresponding annotation. The set of latent variables that the collapsed Gibbs sampler draws is $\{y, b, a\}$. When y, b, and $a_{\backslash i}$ are given,[4] the sample of a_i can be obtained from Eq. (4) easily. However, y and b involve the sequence structure, so that we need an efficient sampler. The conditional distribution of y and b that is required to compose the sampler is below.

[4] The index $\backslash i$ indicates a set of the variables except for the ith variable.

$$
\begin{aligned}
& p(b^{(i)}, y^{(i)} | X, Z, b^{(\backslash i)}, y^{(\backslash i)}, a, \phi) \\
& \propto p(x^{(i)}, b^{(i)} | y^{(i)}, x^{(\backslash i)}, y^{(\backslash i)}, b^{(\backslash i)}, a, \phi) p(y^{(i)} | y^{(\backslash i)}) \delta_{S(z^{(i)}, y^{(i)}, b^{(i)})} \\
& \approx \prod_{k=1}^{K} \mathscr{P}_{y_k^{(i)}}(x^{(i)}_{b^{(i)}_{k-1}+1:b^{(i)}_k} | z^{(i)}, x^{(\backslash i)}, y^{(\backslash i)}, b^{(\backslash i)}, a, \phi) \prod_{k=1}^{K} p(y_k^{(i)} | y_{k-1}^{(i)}, y^{(\backslash i)}) \delta_{S(z^{(i)}, y^{(i)}, b^{(i)})} \quad (6)
\end{aligned}
$$

$S(z^{(i)}, y^{(i)}, b^{(i)})$ is a proposition that checks if the labeled segment $(y^{(i)}, b^{(i)})$ is consistent with the supervision $z^{(i)}$. The approximation we applied above ignores the non-Markov dependency between the local random variables in the ith sentence.[5] Each factor in Eq. (6) can be calculated as follows.

$$
\begin{aligned}
\mathscr{P}_{y_k^{(i)}}(x^{(i)}_{b^{(i)}_{k-1}+1:b^{(i)}_k} | x^{(\backslash i)}, y^{(\backslash i)}, b^{(\backslash i)}, a, \phi) &= p_{pypsm}(s_{N+1} = x^{(i)}_{b^{(i)}_{k-1}+1:b^{(i)}_k} | a, \phi, \hat{\psi}_{y_k^{(i)}}, \hat{\xi}_{y_k^{(i)}}) \\
p(y_k^{(i)} | y_{k-1}^{(i)}, y^{(\backslash i)}) &= p_{cat}(y_k^{(i)} | \hat{\theta}_{y_{k-1}^{(i)}})
\end{aligned}
$$

$\hat{\theta}$, $\hat{\psi}_{y_k^{(i)}}$ and $\hat{\xi}_{y_k^{(i)}}$ are respectively the expected value of the variable with the posterior distribution given $x^{(\backslash i)}, y^{(\backslash i)}, b^{(\backslash i)}$.[6] The sample from the distribution of Eq. (6) can be obtained by using a sequence-structured sampling method called forward-backward sampling [21] based on a lattice illustrated in Fig. 3 **(Right)**. Unlike the dynamic programming for HMMs, the proposed model requires restoring the range that corresponds to the phrase for calculating the phrase probability. For this reason, the state in the lattice retains the number of tokens contained in the current segment.

The nodes $v_{t,\tau,y}$ in Fig. 3 **(Right)** such as $v_{1,1,ns}$ and $v_{5,2,sbj}$ indicate a combination of position and label (t, τ, y) that means x_t is the τ-th token of a segment having label y. For example, $v_{1,1,ns}$ indicates x_1 ("remind") is interpreted as the first token for ns (**non-slot**) segment, and $v_{5,2,sbj}$ indicates x_5 ("john") is considered the second token for sbj (**subject slot**) segment. The node $v_{t,\zeta,y}$ $(\tau = \zeta)$ indicates that a segment with the label y is terminated at x_t. Any possible labeled segmentation $(b_{1:K}, y_{1:K})$ has a one-to-one relationship with a path from the node $v_{0,\zeta,BOS}$ to a node $v_{T,\zeta,y}$. Slot annotations in training data can be represented as a set $\mathscr{L}$ of nodes that a path needs to visit. When we denote a slot annotation by a tuple of label and range $(y, i : j)$, $\mathscr{L}$ contains $\{v_{t,\tau,y}\}_{i \le t \le j, 1 \le \tau \le j-i}$ and $\{v_{j,\zeta,y}\}$. For example, the elements in $\mathscr{L}$ corresponding to a slot annotation (**subject**, $4:5$) are the red nodes in Fig. 3 **(Right)**.

For the forward-backward sampling, we first compute forward probabilities $\alpha(v_{t,\zeta,y}) \equiv \sum_k p(x_{1:t}, b_k = t, y_k = y)$ and $\alpha(v_{t,\tau,y}) \equiv \sum_k p(x_{1:t-\tau}, b_{k-1} = t - \tau, y_k = y)$ by using the following recursive formulas with the base $\alpha(v_{1,\zeta,BOS}) = 1$.

[5] As described in [28], the effect of this approximation that ignores the local count is sufficiently small when there are many short sentences. This case applies to the slot filling task.

[6] We can substitute the variables with the expected values because the predictive distribution of a Dirichlet-categorical distribution with $p_{dir}(\theta|\alpha)$ and $p_{cat}(x|\theta)$ equals $p(x_N = k | x_{1:N-1}) = \int p(x_N = k|\theta) p(\theta|x_{1:N-1}) d\theta = \frac{\alpha_k + \sum_{i=1}^{N-1} \delta(x_i = k)}{\sum_k \alpha_k + N - 1} = p_{cat}(x|\theta = E_{p(\theta|x_{1:N-1})}[\theta])$.

$$\alpha(v_{t,\tau,y}) = \left(\delta_{\tau=1} \sum_{y' \in \mathscr{Y}} \alpha(v_{t-1,\zeta,y'}) p(y|\theta_{y'}) + \delta_{\tau>1} \alpha(v_{t-1,\tau-1,y})\right) \delta_{\neg excluded(t,\tau,y)} \tag{7}$$

$$\alpha(v_{t,\zeta,y}) = \sum_{\tau=1}^{t} \alpha(v_{t,\tau,y}) \mathscr{P}_y(x_{t-\tau+1:t}) \tag{8}$$

To exclude a path that does not follow the training annotations, we define a predicate $excluded(t, \tau, y) \Leftrightarrow \exists \tau', y'[v_{t,\tau',y'} \in \mathscr{L} \wedge (\tau', y') \neq (\tau, y)]$ and use it in (7). The backward sampling starts with drawing a sample of the label of the last segment denoted by $\tilde{y}_{\mathscr{K}}$. Then, the segment lengths $\tilde{\tau}_\kappa$ and the segment labels $\tilde{y}_{\kappa-1}$ are sampled recursively in a backward order for $\kappa = \mathscr{K}, \mathscr{K}-1, \mathscr{K}-2, \ldots$. Let $\tilde{b}_{\mathscr{K}} = T$ and $\tilde{b}_{\kappa-1} = \tilde{b}_\kappa - \tilde{\tau}_\kappa$. The sampling repeats until $\tilde{b}_{\kappa-1} = 0$ is obtained. The conditional distributions that the sampler draws from are represented by using the forward probabilities as a straightforward extension of the backward sampling for HMMs [2].

After the sampling iterations, we obtain a sample of the latent variables for all sentences. While Monte Carlo estimation generally takes an average of multiple samples, we simply use a single sample of segmentation to estimate the posterior of the model parameters [1]. The computational complexity of the algorithm that processes one sentence is $\mathscr{O}(TL_{max}^2)$ because the dominant factor Eq. (8) requires computations for $t = 1$ to T and $\tau = 1$ to $min(L_{max}, t)$, and each computation of $\mathscr{P}_y(x_{t-\tau+1:t})$ involves the calculation of N-gram probability that requires τ iteration.

3.3 Finding the Most Likely Labeled Segmentation

To complete the slot filling task, we need to find the most likely labeled segmentation regarding the trained PYPSMs. Such segmentation can be obtained by an algorithm to find the shortest path on the lattice. We define a cost function $f : E \to \mathbb{R}$ to make the sum of the costs in a path to be equivalent to the negative log likelihood of the corresponding labeled segmentation. For an initialization edge in E_I, the cost is the negative log probability of the corresponding label transition, $f(v_{t,\zeta,y} \to v_{t+1,1,y'}) = -\log p(y'|\theta_y)$. No cost is imposed to cross a continuation edge in E_C, i.e., $f(v_{t,\tau,y} \to v_{t+1,\tau+1,y}) = 0$. For a termination edge in E_T, the cost is the negative log probability of the phrase in the current segment, which is calculated by using a PYPSM, $f(v_{t,\tau,y} \to v_{t,\zeta,y}) = -\log \mathscr{P}_y(x_{(t-\tau+1):t})$. Under this definition of the cost function, the shortest path that minimizes the sum of the costs is guaranteed to correspond to the labeled segmentation that maximizes Eq. (1).

4 Experiment

4.1 Datasets

We use two datasets called DSTC corpus and weather corpus to evaluate the proposed methods. The DSTC corpus is a collection of English utterances for restaurant search provided at the dialog state tracking challenge 3 [5]. We extracted the first utterance in each dialog session, which typically describes the preference of restaurant. The sentences that contain no slot information are excluded. The corpus consists of 1,441 sentences with five possible slot types, **area**, **food**, **price range**, **type**, and **children allowed**. We manually identified the substring that expresses the slot value if the slot value does not match with any substring in the sentence.

The weather corpus is an in-house dataset of Japanese utterances that ask about the weather (for example, "Tell me the amount of precipitation in Tokyo tomorrow." in Japanese). The utterances are collected from users who accessed to a prototype dialog system that can reply about weather information. We manually annotated slot information to all the utterance for three slot types, **when**, **where**, and **what**. The weather corpus consists of 1,442 sentences.

For each dataset, we splitted the set of sentences into evaluation (90%) and validation (10%) subsets for hyperparameter search. From the evaluation set, we organized train/test subsets in a 10-fold cross validation manner.

4.2 Settings

The proposed method has two variants: **PYPSM** presented in Section 3, and **NGSM** that uses n-gram slot models instead of PYPSMs. By using the validation set, hyperparameter search is conducted for deciding the PYCRP parameters α, β, the number of non-slot labels $|\mathscr{U}|$, and the context length in N-gram N, for each dataset. The best configuration was $\alpha = 1.0$, $\beta = 0.1$, $|\mathscr{U}| = 3$, and $N = 4$ for the DSTC dataset and $\alpha = 1.0$, $\beta = 0.1$, $|\mathscr{U}| = 5$, and $N = 3$ for weather dataset. The Gibbs sampling and Viterbi decoding are applied to both method for training and inference. We compared the accuracy of the proposed method with the following existing methods:

- **BiLSTM-CRF** Neural network proposed in [10] with word and character embeddings and bidirectional LSTM. We set the hidden dimension of LSTM to 128 and the number of LSTM layers to 2 based on the hyperparameter search.
- **HMM** Hidden Markov model, which is a generative model based on sequence labeling formulation with IOB2 tagging scheme. The HMM is trained in a fully supervised manner by associating the sequence labels to the hidden states.

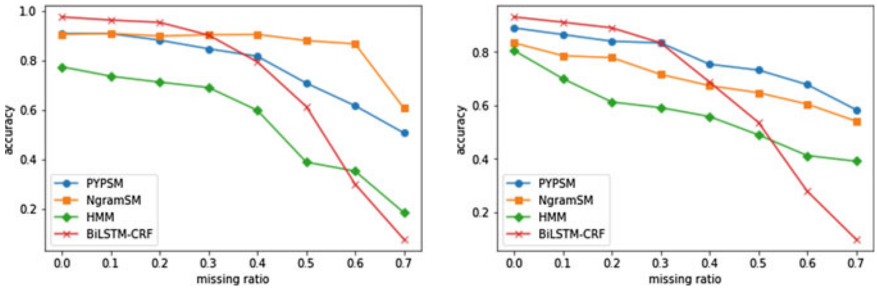


Fig. 4 Accuracy of slot estimation in DSTC corpus **(Left)** and weather corpus **(Right)**

We implemented the proposed method in Java and the BiLSTM-CRF in Python with Anago library.[7] We set $L_{max} = 32$. We treat a word as a unit of token for the DSTC corpus and a character as a token for the weather corpus. This is because Japanese sentences do not contain whitespace letters that indicate word boundaries. The n-gram models in the proposed method (NGSM and PYPSM) are based on characters including the whitespace letters for both datasets.

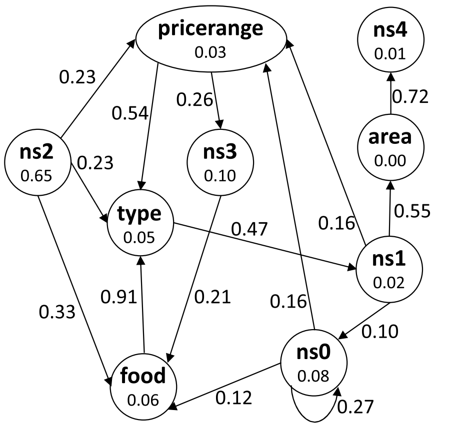
We calculated the slot estimation accuracy with 10-fold cross validation. The accuracy is defined as the ratio of the number of utterances that have the exactly correct slot estimation against the number of all test utterances. For the experiment on incomplete annotation, we simulate the missing annotation by dropping the annotated slot information randomly in various missing ratio from 0.0 (complete annotation) to 0.7 (highly incomplete annotation).

4.3 Results

Figure 4 shows the estimation accuracy. The horizontal axis indicates the missing ratio of annotation and the vertical axis indicates the averaged accuracy of the 10-fold cross validation. The accuracy of BiLSTM-CRF is the highest among all the methods when the missing rate is low. However, the performance of BiLSTM-CRF apparently degrades as the missing rate is higher. The generative models including the proposed method and HMM seem to be more tolerant against the missing annotation. Compared with the HMM, the proposed method significantly improves the estimation performance. This implies that the proposed segmentation-based formulation is more suitable than sequence labeling formulation for slot filling tasks.

Another advantage of the proposed method is the interpretability of model parameters. Figure 5 is a diagram representing the PYPSM model parameters we obtained on DSTC dataset with missing ratio 0.0 (annotated completely). The left side of the figure represents the transition parameters among labels. The values in the nodes

[7] https://github.com/Hironsan/anago.



ns0	a (28), an(12), I would like (10), to find a (10), the (9), uh (8), a tv (7), and (6), trying (5)
ns1	in the (101), in (59), with (11), I (11), with a (10), and (8), that (5), can I (5), should (4), hello (4)
ns2	I'm looking for a (161), I'm looking for an (29), I want to find a (22), a (21), looking for a (12)
ns3	price range (57), I am looking for a (20), yes I'm looking for a (13), unintelligible (8), about the (7)
ns4	area (112), and it should be in the (26), and it should have (10), be (5), tv(5), please (4), it (4), a television (4)

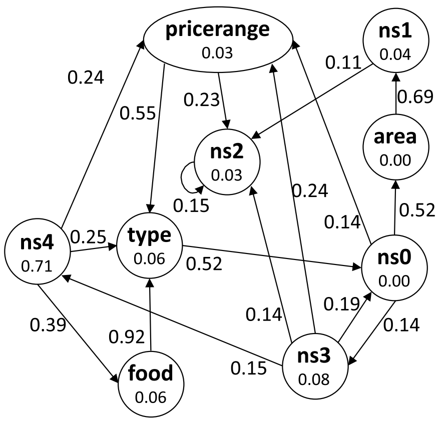
Fig. 5 Parameters of PYPSM obtained by training on DSTC dataset with missing ratio 0.0. (**Left**) Transition parameters among slot labels and non-slot labels (ns0, ..., ns4). The numbers by the edges are the transition probability $p(y_k|y_{k-1})$. The numbers in the circles are the initial label probability $p(y_1)$. (**Right**) Substrings recognized as a non-slot part in the test dataset. The numbers in parentheses are the frequency

indicate the initial probability for the corresponding label. The right side of the figure shows the list of top phrases assigned to the non-slot labels in the test data (the parenthesized numbers are the frequency). We can reconstruct typical utterances by examining this diagram. For example, one of the likely paths is **ns2** (e.g., "I'm looking for a"), **food** (e.g., "Italian"), **type** (e.g., "restaurant"), **ns1** (e.g., "in the"), **area** (e.g., "new chesterton"), **ns4** (e.g., area).

The robustness of the proposed method against a high missing ratio can be observed also in the invariance of the extracted pattern. A diagram for the model trained on DSTC with missing ratio 0.7 is shown in Fig. 6. The structure of the transition pattern resembles the diagram in Fig. 5, and the path we observed in Fig. 5 can be found in this diagram too. This indicates that the proposed model is capable of capturing the structural pattern of utterances from the partial annotations.

Table 1 shows examples of the prediction by models trained on the DSTC dataset with missing ratio 0.7. For the first example, the BiLSTM-CRF failed to detect the area slot even though it is a typical way to mention area information. We believe the area slot could be detected if the BiLSTM-CRF is trained on the perfectly annotated dataset. Contrarily, the PYPSM could detect the slot information probably because of the robust modeling of the non-slot segments.

The second example is the case that demonstrates the effectiveness of explicit probabilistic modeling on the phrases. The PYPSM is less likely to misrecognize phrases that are observed during the training because of the "memorizing" property [4]. On the other hand, on imperfect training data, discriminative models tend to be uncertain about the label for the phrase "chinese" should be **food** or not.



ns0	in the (127), in (60), and (35), with (13), and it should have (10), I (8), please (4), should have (4)
ns1	area (110), about the (10), ah (9), can I (6), you (3), me (3), allow children (3), I would like (3)
ns2	price range (60), a tv (9), unintelligible (8), internet connection and (7), the (7), be (5)
ns3	it should be (30), a (19), an(12), hi (10), with a (10), uh (8), and I (8), to (6), and (6), I need to find a (4)
ns4	I'm looking for a (163), I'm looking for an (29), a (27), I want to find a (22), I am looking for a (20)

Fig. 6 Parameters of PYPSM obtained by training on DSTC dataset with missing ratio 0.7

Table 1 Examples of the prediction by models trained on the DSTC dataset with missing ratio 0.7. Asterisk (*) indicates misrecognition

Utterance	BiLSTM-CRF	PYPSM
Expensive restaurant in the trumington area	**pricerange**: expensive,	**pricerange**: expensive,
	type: restaurant,	**type**: restaurant,
	area: (None) (*)	**area**: trumington
I'm looking for a chinese and it should be in the cherry hinton area	**type**: chinese (*),	**food**: chinese,
	area: cherry hinton	**area**: cherry hinton
I want to find a chinese take away	**food**: chinese take away	**food**: chinese (*),
		food: take away (*)

The third example shows the downside of the memorizing property of the proposed method. While "chinese take away" is another genre of food than "chinese", the PYPSM discretely assigns high probability to "chinese" and recognizes it as an independent slot value. For this example, "take away" is also recognized as another **food** slot value. This kind of generalization error might be mitigated by introducing a constraint that prevents such a split recognition of the same slot values.

5 Conclusion

In this study, we proposed a new formulation of slot filling tasks that is based on an inference of the most likely labeled segmentation. The proposed method considers the probabilities of both slot segments and non-slot segments by a Bayesian model that produces the power-law distribution of phrases. The experimental results show that the proposed method is more accurate than neural network methods when the missing ratio of annotation is high. We empirically showed the proposed model has good interpretability thanks to the formulation based on segment-wise pattern recognition. Future work includes the exploration of more accurate models that are based on the segmentation-based formulation.

Acknowledgements This work was partially supported by JSPS KAKENHI Grant Number 19K20333.

References

1. Bishop C (2006) Pattern recognition and machine learning. Springer
2. Chib S (1996) Calculating posterior distributions and modal estimates in markov mixture models. J Econom 75:79–97
3. Fukubayashi Y, Komatani K, Nakano M, Funakoshi K, Tsujino H, Ogata T, Okuno HG (2008) Rapid prototyping of robust language understanding modules for spoken dialogue systems. In: Proceedings of IJCNLP, pp 210–216
4. Goldwater S, Griffiths TL, Johnson M (2011) Producing power-law distributions and damping word frequencies with two-stage language models. J Mach Learn Res 12:2335–2382
5. Henderson M (2015) Machine learning for dialog state tracking: a review. In: Proceedings of international workshop on machine learning in spoken language processing
6. Henderson MS (2015) Discriminative methods for statistical spoken dialogue systems. PhD thesis, University of Cambridge
7. Jie Z, Xie P, Lu W, Ding R, Li L (2019) Better modeling of incomplete annotations for named entity recognition. In: Proceedings of NAACL: HLT, pp 729–734
8. Jin L, Schwartz L, Doshi-Velez F, Miller T, Schuler W (2021) Depth-bounded statistical PCFG induction as a model of human grammar acquisition. Comput Linguist Assoc Comput Linguist 47(1):181–216
9. Komatani K, Katsumaru M, Nakano M, Funakoshi K, Ogata T, Okuno HG (2010) Automatic allocation of training data for rapid prototyping. In: Proceedings of COLING
10. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. arXiv:1603.01360 [cs.CL]
11. Lim KW, Buntine W, Chen C, Du L (2016) Nonparametric Bayesian topic modelling with the hierarchical Pitman-Yor processes. Int J Approx Reason 78(C):172–191
12. Macherey K, Och FJ, Ney H (2001) Natural language understanding using statistical machine translation. In: Proceedings of EUROSPEECH, pp 2205–2208
13. Mesnil G, Dauphin Y, Yao K, Bengio Y, Deng L, Hakkani-Tur D, He X, Heck L, Tur G, Yu D, Zweig G (2015) Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Trans Audio, Speech, Lang Process 23(3):530–539
14. Nguyen AT, Wallace BC, Li JJ, Nenkova A, Lease M (2017) Aggregating and predicting sequence labels from crowd annotations. In: Proceedings ACL, pp 299–309
15. Niu J, Penn G (2019) Rationally reappraising ATIS-based dialogue systems. In: Proceedings ACL, pp 5503–5507

16. Ponvert E, Baldridge J, Erk K (2011) Simple unsupervised grammar induction from raw text with cascaded finite state models. In: Proceedings ACL, pp 1077–1086
17. Qin L, Liu T, Che W, Kang B, Zhao S, Liu T (2021) A co-interactive transformer for joint slot filling and intent detection. In: Proceedings ICASSP, pp 8193–8197
18. Raymond C, Riccardi G (2007) Generative and discriminative algorithms for spoken language understanding. In: Proceedings of Interspeech
19. Rodrigues F, Pereira F, Ribeiro B (2014) Sequence labeling with multiple annotators. Mach Learn 95(2):165–181
20. Sato I, Nakagawa H (2010) Topic models with power-law using Pitman-Yor process. In: Proceedings KDD
21. Scott SL (2002) Bayesian methods for hidden markov models: recursive computing in the 21st century. J Am Stat Assoc 97:337–351
22. Seneff S (1992) TINA: a natural language system for spoken language applications. Comput Linguist 18(1):61–86
23. Simpson ED, Gurevych I (2019) A bayesian approach for sequence tagging with crowds. In: Proceedings EMNLP, pp 1093–1104
24. Snow R, O'Connor B, Jurafsky D, Ng AY (2008) Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In: Proceedings EMNLP, pp 254–263
25. Teh YW, Jordan MI, Beal MJ, Blei DM (2005) Hierarchical dirichlet processes. J Am Stat Assoc 101:1566–1581
26. Uchiumi K, Tsukahara H, Mochihashi D (2015) Inducing word and part-of-speech with pitman-yor hidden semi-markov models. In: Proceedings ACL-IJCNLP
27. Wakabayashi K, Takeuchi J, Funakoshi K, Nakano M (2016) Nonparametric Bayesian models for spoken language understanding. In: Proceedings EMNLP
28. Wang P, Blunsom P (2013) Collapsed variational Bayesian inference for hidden Markov models. In: Proceedings AISTATS, pp 599–607
29. Xu P, Sarikaya R (2013) Convolutional neural network based triangular CRF for joint intent detection and slot filling. In: Proceedings of IEEE workshop on automatic speech recognition and understanding
30. Yadav V, Bethard S (2018) A survey on recent advances in named entity recognition from deep learning models. In: Proceedings COLING
31. Zhai K, Boyd-graber J (2013) Online latent dirichlet allocation with infinite vocabulary. In: Proceedings of ICML

Can We Predict How Challenging Spoken Language Understanding Corpora Are Across Sources, Languages, and Domains?

Frederic Bechet, Christian Raymond, Achraf Hamane, Rim Abrougui, Gabriel Marzinotto, and Géraldine Damnati

Abstract State-of-the art Spoken Language Understanding models of Spoken Dialog Systems achieve remarkable results on benchmark corpora thanks to the winning combination of pretraining on large collection of out-of-domain data with contextual Transformer representations and fine-tuning on in-domain data. On average, performances are almost perfect on benchmark datasets such as ATIS. However some phenomena can affect greatly these performances, like unseen events or ambiguities. They are the major sources of errors in real-life deployed systems although they are not necessarily equally represented in benchmark corpora. This paper aims to predict and characterize error-prone utterances and to explain what makes a given corpus more or less challenging. After training such a predictor on benchmark corpora from various languages and domains, we confront it to a new corpus collected from a French deployed vocal assistant with different distributional properties. We show that the predictor can highlight challenging utterances and explain the main complexity factors even though this corpus was collected in a completely different setting.

F. Bechet (✉) · A. Hamane
Aix Marseille University, CNRS, LIS UMR 7020, Marseille, France
e-mail: frederic.bechet@lis-lab.f

A. Hamane
e-mail: achraf.hamane@etu.univ-amu.fr

C. Raymond
INSA Rennes - IRISA, Rennes, France
e-mail: christian.raymond@irisa.fr

R. Abrougui · G. Marzinotto · G. Damnati
Orange Labs, Lannion, France
e-mail: rim.abrougui@orange.com

G. Marzinotto
e-mail: gabriel.marzinotto@orange.com

G. Damnati
e-mail: geraldine.damnati@orange.com

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_3

1 Introduction

In the Transformer era, Spoken Language Understanding models of Spoken Dialog Systems have achieved remarkable results on a wide range of benchmark tasks. State-of-the-art models involve contextual embeddings trained on a very large quantity of out-of-domain text, usually with a Transformer approach, followed by a fine-tune training process on in-domain data to generate the semantic representation required, often made of intent+concept/value labels [10].

This winning strategy gives a boost in performance compared to previous models, mostly because of the generalization power of pretrained contextual embeddings. However, if on some SLU benchmark corpora like ATIS, such models have reached almost perfect performance, other corpora remain challenging and performance can be greatly affected by the amount and the quality of data available for training or by the complexity and ambiguity of the semantic annotation scheme.

But how can we characterize how challenging a corpus is? What are the factors that explain why some utterances still resist to Transformer-based models? And can we predict automatically this complexity when dealing with a new corpora in order to partition data into several sets representing different sources and levels of difficulty?

Moreover, it was noticed in [3, 9] that standard benchmark datasets don't contain enough *difficult* examples that can be found in real-life deployed services, giving a false impression that there are no margin of improvement in current models. Furthermore, the distribution of utterances in benchmark corpora doesn't necessarily reflect real-life usage. Distributions in corpora collected from deployed services are more likely to be imbalanced, with on one hand possibly more *easy* utterances that researchers may not consider interesting to integrate in benchmark corpora and on the other hand a larger variety of complex phenomena that are under-represented in benchmark corpora.

This paper aims to give some answers to these questions on benchmark SLU corpora as well as a new dataset collected from a deployed voice assistant in order to verify if knowledge extracted on *artificial* data can generalize to *real* human-machine interactions.

2 Predicting Corpus Complexity

To predict corpus complexity, we follow the approach proposed in [1, 2] inspired by the NIST *Recognizer output voting error reduction* [7] method for scoring Automatic Speech Recognition (ASR) performance. In this method, multiple recognizers output are combined by voting on each decision, the most probable one being the output with most votes. This method acknowledges the fact that there is some kind of uncertainty in the output produced by statistically trained models, therefore using multiple decisions can help increasing robustness in the decision process. This phenomenon is particularly true for current deep learning models which involve some randomness in parameter initialization, leading to produce different performance on different runs of the same model.

In [1], it was proposed to use a modified version of the ROVER method in order to qualify each utterance of an evaluation corpus for an SLU task of semantic concept recognition seen as a sequence labeling problem. By running multiple SLU models on the same data, we obtain several concept recognition hypotheses at the word level. According to the agreement between hypotheses, a cluster label is given to each word: **AC** means that all models agree, and the output is correct; **AE** means that all models agree, and the output is incorrect; **NC** means that some models disagree but at least one of them is correct; **NE** means that some models disagree but none of them is correct. It was hypothesized in [2] that cluster **AC** corresponds to the easy samples, **NC** to the difficult ones, **NE** to the very difficult ones, and finally **AE** to the *problematic* ones, often corresponding to annotation errors.

In this study, we want to go further than just qualifying a sample as *easy* or *difficult* by understanding the reason behind this qualification. Moreover, we want to uncover generic principles, that can be applied to any SLU task, independently from the language, the topic, or the semantic model related to a given corpus. For this purpose, we propose the following method based on a 2-step process:

First step:

1. Select a set of L SLU corpora, with concept annotation at the word level (with B,I,O info if multi-word concepts), partitioned into train, development, and test.
2. Select a set of N Deep Neural Network (DNN) sequence tagger implementing different DNN architectures and using different kinds of word pretraining.
3. Train the N sequence taggers separately on each train partition of the L corpora, and evaluate the performance on their corresponding development and test sets.
4. Label each word in the development and test corpora with the **AC**, **AE**, **NC** and **NE** labels according to the agreement and the correctness of the concept label predicted by the N concept sequence taggers;

An example of such process is given in Table 1 for two SLU concept taggers. Since this utterance contains at least one word labeled **NCE**, it will belong to the **NCE** cluster containing the *difficult* utterances.

The second step of the process aims at understanding what makes a sample *easy* or *difficult*. **AC** samples stand for the *easy* one while labels **AE**, **NC** and **NE** are grouped into a new label **NCE** for *difficult* samples.

Second step:

1. Describe each word in the development and test corpora of each SLU corpus with language independent, topic independent, and concept independent features (*Generic Features—GF*), such as syntactic features and features related to the coverage of the training corpus (e.g., *how many times this word has been seen with this label in the training corpus?*).
2. Train a *glass-box* classifier such as Adaboost on the union of the L development corpora described by *GF* features to predict the complexity labels **AC** and **NCE**

Table 1 Example of annotation of utterance u with two SLU models (m_1, m_2) and the resulting cluster for each word

i	*word* w_i	*label(ref,u,i)*	*label(m_1,u,i)*	*label(m_2,u,i)*	*cluster*
1	find	O	O	O	**AC**
2	flights	O	O	O	**AC**
3	from	O	O	O	**AC**
4	new-york	B-from-city	B-from-city	**B-from-city**	**AC**
5	new-york	O	**B-from-city**	**B-to-city**	**NE** → **NCE**
6	next	B-date-dep	**B-date-dep**	**O**	**NC** → **NCE**
7	saturday	I-date-dep	**I-date-dep**	**B-date-arr**	**NC** → **NCE**

and evaluate its performance on the SLU test corpora also labeled with **AC** and **NCE** labels as in step 1.4.

3. Analyze the classification model obtained by uncovering the rules and their weights automatically learned on *GF* features to predict label **NCE** in order to qualify the major complexity factors on all the SLU corpora considered.

At the end of this 2-step process we obtain a *complexity* classifier that can process any new SLU corpus, regardless of its language, topic, and semantic model, as long as each word is described by *GF* features, without the need to train and evaluate any SLU system. This classifier labels each word with a complexity label (**AC** or **NCE**), a score, and an *explanation* about this complexity, obtained by analyzing the **NCE** rules learned and their weights. This kind of *explanation* is obtained by characterizing each feature type in the *GF* set. This is presented in the next section.

3 Analyzing Complexity Factors

To analyze utterance complexity with respect to an SLU task such as concept tagging, we make the following assumption, following previous work done on Named Entity Recognition [3, 9]: the two main sources of complexity that can affect an SLU model are *ambiguity* and lack of *coverage* of the training corpus.

- **ambiguity**: an utterance can be ambiguous if a word or a sequence of words can correspond to multiple labels in the semantic model and if either there is not enough context to help removing the ambiguity, or if the underlying structure of the utterance is complex (long utterance, multiple verbs, disfluencies, …);
- **coverage**: this source of complexity comes from a lack of coverage between the training and the evaluation data. The most obvious phenomenon is *Out-Of-Vocabulary* words, but it can also come from a new or a rare association between a known word and a label, or a new n-gram of known words.

Table 2 The Generic Feature (GF) set

Ambiguity
# of semantic labels acceptable for W
# of Part-Of-Speech (POS) acceptable for W + POS label
# of possible syntactic dependency for W + dependency label
distance between W and the sentence syntactic root.
utterance length (in words)
% of words in S belonging to a concept
Coverage
# of occurrences of W in train
# of occurrences of (W, l) in train
is bigrams $(W-1, W)$ and $(W, W+1)$ occurring in train?

The features we use in the *GF* set to describe a word W with label l in a sentence S are either related to ambiguity or coverage. They are defined in Table 2.

All the syntactic features are obtained through a parsing process on the train, dev, and test partitions of each corpus. In order to be language independent, we use parsers [12] based on the *Universal Dependency* syntactic model [13]. Hence, syntactic features are shared across languages. Once a corpus is projected into the *GF* feature set, there is no lexical information and no semantic labels left, therefore corpora on different languages, topics, and semantic models can be merged in order to train the *complexity* classifier for producing the **AC** or **NCE** labels.

We use a glass-box classifier called *Bonzaiboost*[1] [8] based on boosting [14] where a set of weak classifiers made of small decision trees on the features of *GF* are weighted in order to predict the output labels. When processing a sentence, the set of rules matching the input features are selected and the label chosen is the one maximizing the score according to the rules weights. When the **NCE** label is predicted, we can check in the selected rules which ones have contributed positively to predict the *difficult* label. Since each rule belongs either to the *ambiguity* or *coverage* set, we can estimate the % of weight in the **NCE** score that belongs to either set, and thus *explain* if this difficulty comes from an ambiguity issue or lack of coverage in the training data.

The classifier outputs decision at the word level, however they can be projected at the sentence level with this simple rule: the *easy* utterances are those where all words have been labeled as **AC**; the *difficult* utterances are those containing at least one word labeled as *NCE*. Therefore, we can use the *complexity* classifier output in order to select utterances with a certain level of difficulty, expressed by the **NCE** score, and belonging either to the *ambiguity* or *coverage* category.

[1] http://bonzaiboost.gforge.inria.fr/.

4 Experiments on Benchmark Corpora

The method presented in the two previous sections has been implemented on 4 SLU benchmark corpora described in Table 3 split into train, dev, and test partitions:

1. M2M: this corpus is a fusion of two datasets containing dialogues for restaurant and movie ticket booking. It has been released by [15] and collected using their M2M framework (Machines Talking To Machines) that combines dialogue self-play and crowd sourcing to generate dialogues.
2. ATIS: The Air Travel Information System (ATIS) task [6] is dedicated to provide flight information.
3. MEDIA: this corpus is made of 1250 French dialogue, dedicated to provide tourist information. It has been collected by ELDA, following a Wizard of Oz protocol: 250 speakers have followed 5 hotel reservation scenarios. This corpus has been transcribed manually and annotated with concepts from a rich semantic ontology [4].
4. SNIPS: this corpus has been collected by the SNIPS company. It is dedicated to 7 in-house tasks, SearchCreativeWork, GetWeather, BookRestaurant,PlayMusic, AddToPlaylist, RateBook, SearchScreeningEvent [5].

In order to obtain the complexity labels **AC** and **NCE**, we developed 6 SLU sequence tagger models (M1…M6) in order to predict concept labels at the word level on our 4 corpora. These 6 systems differ either by the pretraining condition (BERT or random initialization) and the DNN architecture (GRU, BIGRU, or self-attention) as described in Table 4 . These systems follow state-of-the-art architectures for SLU concept tagging [10]. If BERT pretraining outperforms by a large margin random initialization, it is interesting to keep this option for detecting easy utterance that does not need any generalization capabilities outside the training data. Table 5 shows F-measure results obtained by all systems on the four corpora.

Table 3 Corpora characteristics

Corpora	ATIS	MEDIA	SNIPS	Djingo_Spk
#word	8333	25977	6595	34938
#sent	893	3005	700	9984
vocabulary	485	1219	1752	2637
#concept	84	70	39	34
#intent	–	–	7	109
%OOD sentences	0	0	0	6.6%
%sent $\in$ *train* $\cap$ *test*	1.9	44.6%	0.9%	76.9%
%sent+concept	99.3%	86.5%	100%	59.3%
av. sent length	10.3	7.6	9.16	4.2

Table 4 Description of models M1 to M6 in terms of pretraining conditions and DNN architecture

Pretraining	bigru	gru	Self-attention
BERT	M1	M3	M5
random	M2	M4	M6

Table 5 Concept detection performance (F-measure) for models M1…M6 on the 4 benchmark corpora

Model/F-measure	ATIS	MEDIA	SNIPS	M2M
M1	94.6	85.7	95.4	91.5
M2	93.8	81.7	69.6	91.7
M3	94.7	85.8	95.2	93.6
M4	79.0	60.1	69.0	91.0
M5	94.8	85.3	95.9	93.0
M6	77.4	59.8	68.9	91.0

Table 6 Repartition into *easy* (AC) and *difficult* (NCE) samples at the word and sentence levels

Label/%	ATIS (%)	MEDIA (%)	SNIPS (%)	M2M (%)
AC (word)	89.8	70.1	83.1	96.1
NCE (word)	10.2	29.9	16.9	3.9
AC (sent)	46.2	54.3	35.1	84.2
NCE (sent)	53.8	45.7	64.9	15.8

As we can see models without pretraining (M2, M4, and M6) obtain much worst performance on all corpora except M2M, first indication that this corpus does not need generalization capabilities.

From the automatic labeling with models M1 to M6, we can compute labels AC and NCE at the word and sentence levels as presented in Sect. 2. The repartition between *easy* (AC) and *difficult* (NCE) utterances is presented in Table 6. We can see that the amount of *difficult* tokens and sentences differ greatly from one corpus to another, giving more insights about the complexity of a given corpus than just looking at the average SLU performance. For example, although the M2M corpus seems more challenging that ATIS and SNIPS according to the best model (M1) in Table 5, we can see in Table 6 that it contains a lot more of *easy* tokens and sentences than the other corpora.

Table 7 clearly indicates the relevance of the AC/NCE clustering since performance obtained with a state-of-the-art model such as M1 obtain much worse results on NCE utterances compared to AC utterances.

Following the method presented in Sect. 3, we trained a *Bonzaiboost* classifier to predict the complexity labels **AC** and **NCE** on the union of the 4 development corpora. The results are presented in Table 8. As we can see, if the classification results

Table 7 Performance of model M1 on AC and NCE sentences

Label/Fmes	ATIS	MEDIA	SNIPS	M2M
AC	98.7	98.5	99.7	99.0
NCE)	91.7	82.3	93.1	68.6

Table 8 Classification performance on AC/NCE labels with the GF feature set. Training on the union of all corpora

ATIS	**Precision**	**Recall**	**F-measure**
AC	91.75	98.26	94.89
NCE	60.61	23.26	33.61
MEDIA	**Precision**	**Recall**	**F-measure**
AC	82.55	87.82	85.11
NCE	63.03	52.80	57.46
SNIPS	**Precision**	**Recall**	**F-measure**
AC	92.54	96.04	94.26
NCE	58.93	42.31	49.25
M2M	**Precision**	**Recall**	**F-measure**
AC	98.08	99.89	98.98
NCE	97.00	65.10	77.91
All corpora			
All	**Precision**	**Recall**	**F-measure**
AC	**91.58**	**95.57**	**93.53**
NCE	**68.42**	**52.21**	**59.23**
All	**88.83**	**88.83**	**88.83**

vary according to the corpus considered, we obtain an F-measure over 93% for label AC and almost 60% on label NCE. These are encouraging results considering that no lexical nor semantic labels are used as features to predict utterance complexity and that we mix in the training and test conditions very different SLU corpora on different languages, topics and semantic models.

Table 9 shows the analysis of the NCE decisions in terms of the respective weights of the *ambiguity* and *coverage* features as described in Sect. 3. As we can see it is interesting to notice that, depending on the corpus considered, the complexity can come mostly because of coverage issues (ATIS and M2M), ambiguity issues (MEDIA) or a mix of both (SNIPS). The distribution obtained on partitions obtained with predicted labels, rather than reference ones are very similar. This is also encouraging showing that even if the complexity classifier makes errors (60% Fmeasure), it can still be used to accurately partition a corpus according to criteria linked to the utterance complexity and the sources of this complexity.

Table 9 % of weight for boosting rules belonging to the **ambiguity** (AMBIG) category *versus* the **coverage** (COVER) category

ATIS	**weight(NCE,AMBIG)**	**weight(NCE,COVER)**
reference	13.1%	86.9%
prediction	19.9%	80.1%
MEDIA	**weight(NCE,AMBIG)**	**weight(NCE,COVER)**
reference	84.4%	15.6%
prediction	84.3%	15.7%
SNIPS	**weight(NCE,AMBIG)**	**weight(NCE,COVER)**
reference	37.2%	62.8%
prediction	23.5%	76.5%
M2M	**weight(NCE,AMBIG)**	**weight(NCE,COVER)**
reference	4.1%	95.9%
prediction	2.3%	97.7%
all	**weight(NCE,AMBIG)**	**weight(NCE,COVER)**
reference	**65.8%**	**34.2%**
prediction	**68.0%**	**32.0%**

5 Application to Deployed SLU System Data

In addition to the previous experiments on benchmark corpora obtained either through a *Wizard-Of-Oz* paradigm (ATIS, MEDIA), or through an automatic process with human supervision (SNIPS, M2M), we decided to test the genericity of our approach on a corpus collected through a deployed service by *Orange* in France.

Orange, the French telco company, has experimented towards the general public the *Djingo* vocal domestic assistant with a set of skills centered on interactions with corporate services (Orange TV, music with its partner Deezer, Orange Radio, telephony), general services (weather, shopping, calendar, news) and general interaction with the speaker (small talks, global commands). According to the customer agreement, and in respect of the French GDPR law, log data have been anonymously collected and annotated in terms of intents and concept slots. The annotated corpus is built on a weekly basis, and corresponds to a random sub-sampling of a whole week logs. The sub-sampling strategy is guided by the annotation capacity for a given week (the average amount of annotations produced by annotators, denoted N_a) and is motivated by the objective of preserving the original distribution of utterances in the test set. Note that the annotations are not produced by crowd sourcing but by expert annotators. Let L be the set of logs gathered during a week, L can be divided into L_s, the subset of already seen utterances, present in the annotation database and $L_u = \overline{L_s}$, the subset of unseen utterances that constitute the pool of candidates for annotation. In a first step, L_u is randomly down sampled to N_a samples, and the corresponding random sampling probability is applied to L_s in order to derive a down-sampled subset from already annotated samples. The corpus also contains out-of-domain utterances that are labeled as “NoIntent”. The data distribution strategy and the presence of out-of-domain utterances constitute the most significant differences between this dataset and public benchmark datasets.

Semantic annotations are directly performed on ASR transcriptions and annotated automatic transcriptions are used both for training and testing the NLU model.

For these experiments, the test set is composed of 9984 utterances randomly sub-sampled from a full week of logs. The training corpus is composed of a set of anterior utterances, respecting the usage distribution except that the number of duplicate occurrences for a given utterance is notched to a maximum value of 50 in order to avoid over representation of some very common commands. Overall, the training corpus contains 279375 utterances (with 52132 different utterances). The model ontology is composed of 233 intents and 42 concepts. As can be seen in Table 3, the characteristics of the *Djingo* corpus are different from benchmark corpora from several perspectives.

The distribution of utterances reflects the usage and we observe for instance a larger proportion of utterances that are observed in the training corpus, but also a set of out-of-domain utterances and a significant amount of utterances without any concepts.

The SLU model used for this study is a Camembert Transformer [11] fine-tuned on the task of jointly predicting the concept slots with a BIO encoding and the sample’s intent, with the intent label set on the $[CLS]$ first token, as in the example below.

[CLS]	put	france	info
Set_Radio_Channel	O	B-channel	I-channel

In early experiments, we tested different pretrained models and different output layer configurations. As they had similar performances we settle for the fine-tuned Camembert baseline with a simple linear output layer. The model was trained using Pytorch and hyperparameters were chosen using an internal architecture hyper parameter completion toolbox (batch size of 10, learning rate of 5.0e-05, samples padded to a maximum of 50 word pieces, Adam optimizer and 5 epochs).

The evaluation of this SLU model on the *Djingo* corpus is given in the last column of Table 10. We show 3 metrics: token accuracy, F-measure on concepts, and sentence accuracy where a sentence is correct only if both the intent and the concept sequence are correct. As can be seen, the performance is in line with those obtained in Table 5.

We applied our complexity classifier on the *Djingo* corpus without any retraining or adaptation. We partitioned the corpus into an *easy* set and a *difficult* one according to the label predicted by the classifier. As we can see in Table 10, 86.5% of the sentences were labeled as **AC** sentences are 13.5% as **NCE**. By measuring the SLU performance on these 2 subsets, we can check if the AC/NCE prediction is indeed predicting sentence complexity. Results in Table 10 show that the predicted labels are meaningful since there is a drop of an absolute 16% between results on partition **AC** (95.7) compared to the **NCE** (79.7) partition.

By looking at the distribution of the weights between the *ambiguity* rules and the *coverage* ones, we observed that if issues linked to a lack of coverage in the training

Table 10 Evaluation of *easy* (AC) and *difficult* (NCE) partitions of the Djingo corpus thanks to the AC/NCE labels predicted by the complexity classifier

Partition	AC	NCE	All
coverage	86.5%	13.5%	100%
token accuracy	98.6	92.4	97.3
F1 concepts	95.6	83.8	92.2
intent+concepts OK	95.7	79.7	93.5
weight(AMBIG)	–	28.9	–
weight(COVER)	–	71.1	–

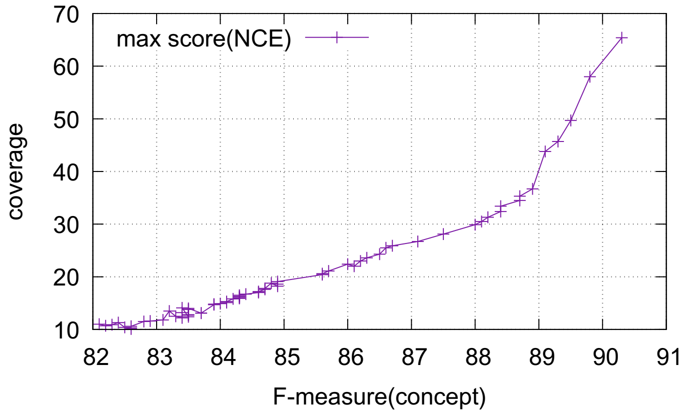


Fig. 1 F-measure versus coverage for different partitions of the eval corpus according to thresholds applied on the predicted **difficulty** (NCE) score

date represent 71.1% of the weights, nearly 30% come from ambiguity issues, making this corpus more challenging than ATIS or M2M where a very large majority of rules came from a lack in the training data.

In addition to the use of the AC/NCE prediction, we wanted also to check if the confidence scores given by *Bonzaiboost* on the **NCE** label predictions, could be used to partition further this corpus into sets of different complexity. To this purpose we tested a very simple approach consisting of fixing a threshold δ, then selecting all sentences containing at least one word labeled **NCE** with a score above threshold δ.

By varying δ we obtain the curve of Fig. 1 which plots the F-measure on concept with respect to the coverage of the corresponding partition. This curve clearly indicates that the NCE label scores are meaningful as they allow to select sentences of various complexity.

6 Conclusion

We have shown in this study that it was possible to predict sentence complexity without running an SLU system on the data. Just by defining very generic features that could be related either to ambiguity issues, or lack of coverage in the training data, we can process corpora in different languages, topics, and semantic models without adaptation. Furthermore, the complexity classification model can be analyzed to explain the major complexity factors on the corpus considered, leading to a better characterization of corpora. Finally, the model was successfully applied on a new corpus collected from a deployed vocal assistant with real-usage distributions, enabling to predict and explain complex utterances.

References

1. Béchet F, Raymond C (2018) Is ATIS too shallow to go deeper for benchmarking Spoken Language Understanding models? In: InterSpeech 2018, pp 1–5. Hyderabad, India. https://hal.inria.fr/hal-01835425
2. Béchet F, Raymond C (2019) Benchmarking benchmarks: introducing new automatic indicators for benchmarking spoken language understanding corpora. In: InterSpeech. Graz, Austria. https://hal.archives-ouvertes.fr/hal-02270633
3. Bernier-Colborne G, Langlais P (2020) HardEval: Focusing on challenging tokens to assess robustness of NER. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp 1704–1711. European Language Resources Association, Marseille, France. https://www.aclweb.org/anthology/2020.lrec-1.211
4. Bonneau-Maynard H, Rosset S, Ayache C, Kuhn A, Mostefa D (2005) Semantic annotation of the French media dialog corpus. In: InterSpeech. Lisbon. ftp://tlp.limsi.fr/public/IS052010.PDF
5. Coucke A, Saade A, Ball A, Bluche T, Caulier A, Leroy D, Doumouro C, Gisselbrecht T, Caltagirone F, Lavril T, Primet M, Dureau J (2018) Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. CoRR. http://arxiv.org/abs/1805.10190
6. Dahl DA, Bates M, Brown M, Fisher W, Hunicke-Smith K, Pallett D, Pao C, Rudnicky A, Shriberg E (1994) Expanding the scope of the ATIS task: the ATIS-3 corpus. In: HLT, pp 43–48
7. Fiscus JG (1997) A post-processing system to yield reduced word error rates: recognizer output voting error reduction (rover). In: 1997 IEEE workshop on automatic speech recognition and understanding proceedings, pp 347–354. IEEE
8. Laurent A, Camelin N, Raymond C (2014) Boosting bonsai trees for efficient features combination : application to speaker role identification. In: Interspeech. Singapour, Singapore. https://hal.inria.fr/hal-01025171
9. Lin H, Lu Y, Tang J, Han X, Sun L, Wei Z, Yuan NJ (2020) A rigorous study on named entity recognition: can fine-tuning pretrained model lead to the promised land? In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 7291–7300. Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.emnlp-main.592. https://www.aclweb.org/anthology/2020.emnlp-main.592
10. Louvan S, Magnini B (2020) Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: a survey. arXiv:2011.00564
11. Martin L, Muller B, Ortiz Suárez PJ, Dupont Y, Romary L, de la Clergerie É, Seddah D, Sagot B (2020) CamemBERT: a tasty French language model. In: Proceedings of the 58th

annual meeting of the association for computational linguistics, pp 7203–7219. Association for Computational Linguistics, Online. https://www.aclweb.org/anthology/2020.acl-main.645
12. Nasr A, Dary F, Bechet F, Favre B (2020) Annotation syntaxique automatique de la partie orale du CÉFC. Langages. https://hal.archives-ouvertes.fr/hal-02973242
13. Nivre J, De Marneffe MC, Ginter F, Goldberg Y, Hajic J, Manning CD, McDonald R, Petrov S, Pyysalo S, Silveira N, et al (2016) Universal dependencies v1: a multilingual treebank collection. In: Proceedings of the tenth international conference on language resources and evaluation (LREC'16), pp 1659–1666
14. Schapire RE, Singer Y (2000) Boostexter: a boosting-based system for text categorization. Mach Learn 39(2):135–168
15. Shah P, Hakkani-Tür D, Tür G, Rastogi A, Bapna A, Nayak N, Heck L (2018) Building a conversational agent overnight with dialogue self-play. arXiv:1801.04871

Personalisation and Generation

Personalized Extractive Summarization with Discourse Structure Constraints Towards Efficient and Coherent Dialog-Based News Delivery

Hiroaki Takatsu, Ryota Ando, Hiroshi Honda, Yoichi Matsuyama, and Tetsunori Kobayashi

Abstract In this paper, we propose a method to generate a personalized summary that may be of interest to each user based on the discourse structure of documents in order to deliver a certain amount of coherent and interesting information within a limited time, primarily via a spoken dialog form. We initially constructed a news article corpus with annotations of the discourse structure, users' profiles, and interests in sentences and topics. The proposed summarization model solves an integer linear programming problem with the discourse structure of each document and the total utterance time as constraints and extracts sentences that maximize the sum of the estimated degree of user's interest. The degree of interest in a sentence is estimated based on the user's profile obtained from a questionnaire and the word embeddings of BERT. Experiments confirm that the personalized summaries generated by the proposed method transmit information more efficiently than generic summaries generated based solely on the importance of sentences.

H. Takatsu (✉) · Y. Matsuyama · T. Kobayashi
Waseda University, Tokyo, Japan
e-mail: takatsu@pcl.cs.waseda.ac.jp

Y. Matsuyama
e-mail: matsuyama@pcl.cs.waseda.ac.jp

T. Kobayashi
e-mail: koba@waseda.jp

R. Ando
Naigai Pressclipping Bureau, Ltd., Tokyo, Japan
e-mail: ando@naigaipc.co.jp

H. Honda
Honda Motor Co., Ltd., Tokyo, Japan
e-mail: hiroshi_01_honda@jp.honda

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_4

1 Introduction

As people's interests and preferences diversify, the demand for personalized summarization technology has increased [1]. Summaries are classified as generic or user-focused, based on whether they are specific to a particular user [2]. Unlike generic summaries generated by extracting important information from the text, user-focused summaries are generated based not only on important information but also on a user's interests and preferences. Most user-focused summarization methods rank sentences based on a score calculated considering user's characteristics and subsequently generate a summary by extracting higher-ranked sentences [3–5]. However, such conventional user-focused methods tend to generate incoherent summaries. Generic summarization methods, which consider the discourse structure of documents, have been proposed to maintain coherence [6–8]. To achieve both personalization and coherence simultaneously, we propose a method to extract sentences that may be of interest according to a user's profile and generate a personalized summary for each user while maintaining coherence based on the discourse structure of documents.

As mobile personal assistants and smart speakers become ubiquitous, the demand for spoken dialog technology has increased. However, dialog-based media is more restrictive than textual media. For example, when listening to an ordinary smart speaker, users can not skip unnecessary information or skim only for necessary information. Thus, it is crucial for future dialog-based media to extract and efficiently transmit information that the users are particularly interested in without excess or deficiencies.

We utilize the proposed personalized summarization method for a spoken dialog system that delivers news as a realistic application [9]. This news dialog system proceeds the dialog according to a primary plan to explain the summary of the news article and subsidiary plans to transmit supplementary information through question answering. As long as the user is listening passively, the system transmits the content of the primary plan. The personalized primary plan generation problem can be formulated as follows:

> From N documents with different topics, sentences that may be of interest to the user are extracted based on the discourse structure of each document. Then the contents are transmitted by voice within T seconds.

Specifically, this problem can be formulated as an integer linear programming problem, which extracts sentences that maximize the sum of the degree of user's interest in the sentences of each document with the discourse structure of documents and the total utterance time T as constraints. The degree of interest in a sentence is estimated based on the user's profile obtained from a questionnaire and the word embeddings of bidirectional encoder representations from transformers (BERT) [10]. To evaluate the effectiveness of the proposed method, we construct a news article corpus with annotations of the discourse structure, users' profiles, and interests in sentences and topics.

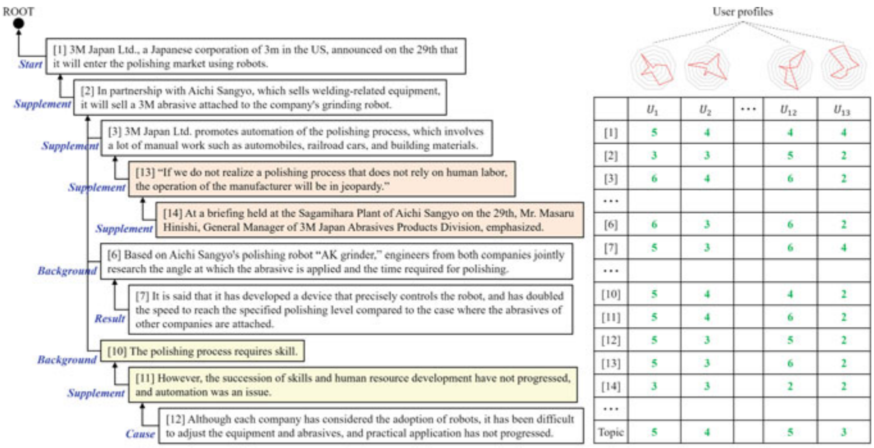
The rest of this paper is organized as follows. Section 2 overviews the discourse structure annotation and interest data collection. Section 3 describes the proposed method. Section 4 evaluates its performance. Section 5 provides the conclusions and future prospects.

2 Datasets

We constructed a news article corpus with annotations of the discourse structure, users' profiles, and interests in sentences and topics. Figure 1 shows an example of the annotation results. Experts annotated the inter-sentence dependencies, discourse relations, and chunks for the Japanese news articles. The users' profiles and interests in the sentences and topics of news articles were collected via crowdsourcing.

2.1 Discourse Structure Dataset

Two web news clipping experts annotated the dependencies, discourse relations, and chunks for 1,200 Japanese news articles. Each article contained between 15–25 sentences. The articles were divided into six genres: sports, technology, economy, international, society, and local. In each genre, we manually selected 200 articles to minimize topic overlap. The annotation work was performed in the order of depen-



	U_1	U_2	⋯	U_{12}	U_{13}
[1]	5	4		4	4
[2]	3	3		5	2
[3]	6	4		6	2
⋯					
[6]	6	3		6	2
[7]	5	3		6	4
⋯					
[10]	5	4		4	2
[11]	5	4		6	2
[12]	5	3		5	2
[13]	5	3		6	2
[14]	3	3		2	2
⋯					
Topic	5	4		5	3

(b) Interest level

Fig. 1 Example of the annotation results dencies, discourse relations, and chunks. The discourse unit was a sentence, which represents a character string separated by an ideographic full stop.

2.1.1 Dependency Annotation

The conditions in which sentence j can be specified as the parent of sentence i are as follows:

- In the original text, sentence j appears before sentence i.
- The flow of the story is natural when reading from the root node in order according to the tree structure and reading sentence i after sentence j.
- The information from the root node to sentence j is the minimum information necessary to understand sentence i.
- If it is possible to start reading from sentence i, the parent of sentence i is the root node.

2.1.2 Discourse Relation Annotation

A discourse relation classifies the type of semantic relationship between the child sentence and the parent sentence. We defined the following as discourse relations: *Start*, *Result*, *Cause*, *Background*, *Correspondence*, *Contrast*, *Topic Change*, *Example*, *Conclusion*, and *Supplement*. An annotation judgment was made while confirming whether both the definition of the discourse relation and the dialog criterion were met. The dialog criterion is a judgment based on whether the response is natural according to the discourse relation. For example, the annotators checked whether it was appropriate to present a child sentence as an answer to a question asking the cause, such as "Why?" after the parent sentence.

2.1.3 Chunk Annotation

A chunk is a highly cohesive set of sentences. If a parent sentence should be presented with a child sentence, it is regarded as a chunk.

A *hard chunk* occurs when the child sentence provides information essential to understand the content of the parent sentence. Examples include when the parent sentence contains a comment and the child sentence contains the speaker's information or when a procedure is explained over multiple sentences.

A *soft chunk* occurs when the child sentence is useful to prevent a biased understanding of the content of the parent sentence, although it does not necessarily contain essential information to understand the parent sentence itself. An example is explaining the situation in two countries related to a subject, where the parent sentence contains one explanation and the child sentence contains another.

2.1.4 Annotation Quality

A one-month training period was established, and discussions were held until the annotation criteria of the two annotators matched. To validate the inter-rater reliability, the two annotators annotated the same 34 articles after the training period. The Cohen's kappa of dependencies, discourse relations, and chunks were 0.960, 0.943, and 0.895, respectively. To calculate kappa of the discourse relations, the comparison was limited to the inter-sentence dependencies in which the parent sentence matched. To calculate kappa of the chunks, we set the label of the sentence selected as the hard chunk, soft chunk, and other to "1, 2, and 0," respectively. Then we compared the labels between sentences. Given the high inter-rater reliability, we concluded that the two annotators could cover different assignments separately.

2.2 Interest Dataset

Participants were recruited via crowdsourcing. They were asked to answer a profile questionnaire and an interest questionnaire. We used 1,200 news articles, which were the same as those used in the discourse structure dataset. We collected the questionnaire results of 2,507 participants. Each participant received six articles, one from each genre. The six articles were distributed so that the total number of sentences was as even as possible across participants. Each article was reviewed by at least 11 participants.

2.2.1 Profile Questionnaire

The profile questionnaire collected the following information: gender, age, residential prefecture, occupation type, industry type, hobbies, frequency of checking news (daily, 4–6 days a week, 1–3 days a week, or 0 days a week), typical time of day news is checked (morning, afternoon, early evening, or night), methods to access the news (video, audio, or text), tools used to check the news (TV, newspaper, smartphone, etc.), newspapers, websites, and applications used to check the news (Nihon Keizai Shimbun, LINE NEWS, SNS, etc.), whether a fee was paid to check the news, news genre actively checked (economy, sports, etc.), and the degree of interest in each news genre (not interested at all, not interested, not interested if anything, interested if anything, interested, or very interested).

2.2.2 Interest Questionnaire

After reading the text of the news article, participants indicated their degree of interest in the content of each sentence. Finally, they indicated their degree of interest in the topic of the article. The degree of interest was indicated on six levels: 1, not interested at all; 2, not interested; 3, not interested if anything; 4, interested if anything; 5, interested; or 6, very interested.

3 Methods

3.1 Inter-Sentence Dependency Parsing

Figure 2 schematically diagrams the proposed model. First, the sentences are inputted with the title added as ROOT: $S = (s_0 = \text{ROOT}, s_1, \ldots, s_n)$. The words of the sentence are given to BERT to generate the embedding of the top layer corresponding to the [CLS] token. Next, the sentence vector and the embedding of the auxiliary features of the sentence are concatenated and given to the bidirectional model [11] of the gated recurrent unit (GRU) [12]. Here, the auxiliary features are the sentence and paragraph positions in the document and the sentence position in the paragraph. $\boldsymbol{h}_i$ is a vector that concatenated the outputs of the hidden layers in the forward and backward directions of the GRU corresponding to the i-th sentence. Based on the head selection model [13], the probability $P_{head}\left(s_j|s_i, S\right)$ that s_j is the parent of s_i is calculated as

$$P_{head}\left(s_j|s_i, S\right) = \frac{\exp\left(g\left(\boldsymbol{h}_j, \boldsymbol{h}_i\right)\right)}{\sum_{k=0}^{N} \exp\left(g\left(\boldsymbol{h}_k, \boldsymbol{h}_i\right)\right)} \quad (1)$$

$$g\left(\boldsymbol{h}_j, \boldsymbol{h}_i\right) = \boldsymbol{v}_h^{\mathsf{T}} \tanh\left(\boldsymbol{U}_h \boldsymbol{h}_j + \boldsymbol{W}_h \boldsymbol{h}_i\right) \quad (2)$$

where $\boldsymbol{v}_h, \boldsymbol{U}_h, \boldsymbol{W}_h$ are weight parameters.

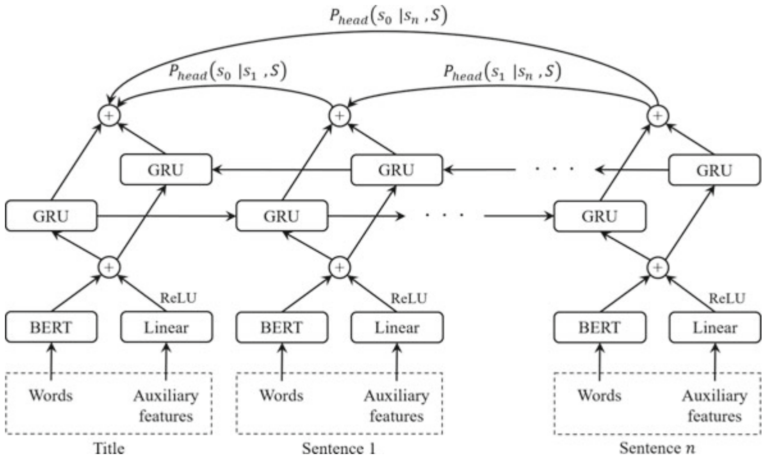


Fig. 2 Inter-sentence dependency parser

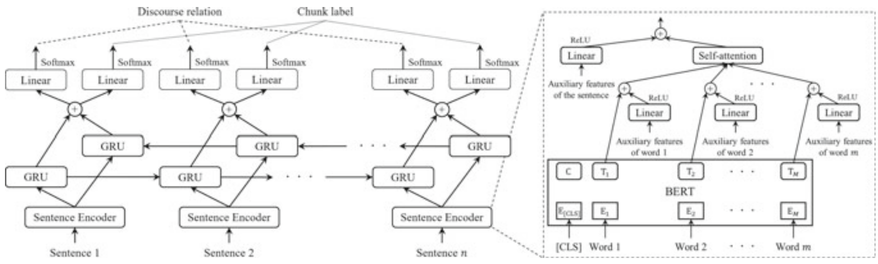


Fig. 3 Discourse relation and chunk estimator

3.2 *Discourse Relation Classification and Chunk Detection*

The discourse dependency tree is decomposed into sentence sequences from the root node to the leaf nodes (hereafter, root-to-leaf). Discourse relations and chunk labels are estimated by sequence labeling for the root-to-leaf sentences. Figure 3 schematically diagrams the proposed model. The total loss function $\mathcal{L}_{total}$ of multi-task learning is defined by the weighted sum of the loss function $\mathcal{L}_r$ of the discourse relation classification task and the loss function $\mathcal{L}_c$ of the chunk detection task. $\mathcal{L}_{total}$ is given as

$$\mathcal{L}_{total} = \lambda_r \times \mathcal{L}_r + \lambda_c \times \mathcal{L}_c \tag{3}$$

where λ_r and λ_c are the weight coefficients of each task.

The discourse relations of the ten labels explained in Sect. 2.1.2 and chunk labels are identified by softmax. The chunks do not distinguish between hard and soft chunks because the number of hard chunks was smaller than the number of soft chunks in the dataset. Chunk labels are defined as “B” for the start of the chunk, “I” for the inside of the chunk, “E” for the end of the chunk, and “O” for the outside of the chunk.

Word information such as a conjunction is an effective clue to identify discourse relations. The sentence encoder calculates self-attention [14] for a combination of word embeddings of BERT and the embedding of the auxiliary features of the word. The obtained vector and the embedding of the auxiliary features of the sentence are concatenated and given to the bidirectional GRU. The sentence auxiliary features are the sentence and paragraph positions in the document, the sentence position in the paragraph, and the depth in the discourse dependency tree. Since the cause of negative events is often negative and the cause of positive events is often positive [15], emotional polarity information can also effectively determine discourse relations. Hence, the word auxiliary features include sentiment polarity information in addition to part of speech and inflected form.

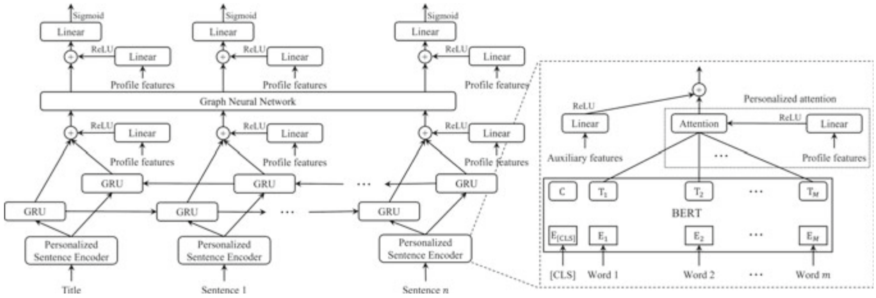


Fig. 4 Interest estimator: BERT_PA_BGRU+_GNN+

3.3 Interest Estimation

Figure 4 overviews the proposed model to estimate the degrees of interest in the topic of a document and each sentence based on the user's profile. The title is inputted before the first sentence because the degree of interest in the title is considered to be the degree of interest in the document's topic. In the personalized sentence encoder, the words of the sentence are given to BERT. Then the personalized attention [16] is calculated for the word embeddings using the profile features as a query. The auxiliary features are the sentence and paragraph positions in the document, the sentence position in the paragraph, and the depth in the discourse dependency tree. Next the sentence vector is given to the bidirectional GRU and the output of the GRU is given to the graph neural network (GNN) [17]. The GNN was introduced based on the analysis of the dataset that the depth of the discourse dependency tree influences the degree of interest. Information is propagated through the dependency structure between sentences with the title as the root node. Profile features are given again before and after the GNN to enhance the effect of the user's profile. Finally, the sentence vectors, which reflect the discourse structure and the user's profile, are given to the output layer with a sigmoid activation function, and the user's interest in each sentence is estimated.

3.4 Interesting Document Selection

The problem of selecting N documents that the user may be interested in from $|D|$ documents with different topics is formulated as an integer linear programming problem as

$$\max. \quad \sum_{k<l\in D} a_k^u a_l^u \left(1 - r_{kl}^d\right) y_{kl}^d \tag{4}$$

Table 1 Variable definitions in the interesting document selection method

x_k^d	Whether document d_k is selected
y_{kl}^d	Whether both d_k and d_l are selected
r_{kl}^d	Similarity between d_k and d_l
N	Maximum number of documents to select
D	Document IDs

s.t.

$$\forall k, l : x_k^d \in \{0, 1\}, \quad y_{kl}^d \in \{0, 1\}$$

$$\sum_{k \in D} x_k^d \leq N \tag{5}$$

$$\forall k, l : \quad y_{kl}^d - x_k^d \leq 0 \tag{6}$$

$$\forall k, l : \quad y_{kl}^d - x_l^d \leq 0 \tag{7}$$

$$\forall k, l : \quad x_k^d + x_l^d - y_{kl}^d \leq 1 \tag{8}$$

Table 1 explains each variable. a_k^u is the degree of user u's interest in the topic of the document d_k estimated by the interest estimator. r_{kl}^d represents the cosine similarity between the bag-of-words constituting d_k and d_l. Equation 5 is a constraint restricting the number of selecting documents is N or less. Equations 6–8 are constraints that set $y_{kl}^d = 1$ when d_k and d_l are selected.

3.5 *Interesting Sentence Extraction*

We considered a summarization problem, which extracts sentences that user u may be interested in from the selected N documents and then transmits them by voice within T seconds. The summary must be of interest to the user, coherent, and not redundant. Therefore, we formulated the summarization problem as an integer linear programming problem in which the objective function is defined by the balance between a high degree of interest in the sentences and a low degree of similarity between the sentences with the discourse structure as constraints.

$$\text{max.} \quad \sum_{k \in D_N^u} \sum_{i < j \in S_k} b_{ki}^u b_{kj}^u \left(1 - r_{kij}^s\right) y_{kij}^s \tag{9}$$

s.t.

$$\forall k, i, j : \; x_{ki}^s \in \{0, 1\}, \quad y_{kij}^s \in \{0, 1\}$$

$$\sum_{k \in D_N^u} \sum_{i \in S_k} t_{ki}^s x_{ki}^s \leq T \tag{10}$$

$$\forall k < l : -L \leq \sum_{i \in S_k} x_{ki}^s - \sum_{i \in S_l} x_{li}^s \leq L \tag{11}$$

$$\forall k, i : \quad j = f_k(i), \quad x_{ki}^s \leq x_{kj}^s \tag{12}$$

$$\forall k, m, i \in C_{km} : \quad \sum_{j \in C_{km}} x_{kj}^s = |C_{km}| \times x_{ki}^s \tag{13}$$

$$\forall k, i, j : \quad y_{kij}^s - x_{ki}^s \leq 0 \tag{14}$$

$$\forall k, i, j : \quad y_{kij}^s - x_{kj}^s \leq 0 \tag{15}$$

$$\forall k, i, j : \quad x_{ki}^s + x_{kj}^s - y_{kij}^s \leq 1 \tag{16}$$

Table 2 explains each variable. Here, the i-th sentence of the k-th document is expressed as s_{ki}. b_{ki}^u is calculated based on the degree of user u's interest in the sentence p_{ki}^u which is estimated by the interest estimator and the utterance time t_{ki}^s as $b_{ki}^u = p_{ki}^u \times t_{ki}^s$ to avoid preferential extraction of short sentences. r_{kij}^s represents the cosine similarity between the bag-of-words constituting s_{ki} and s_{kj}. Equation 10 is a constraint restricting the utterance time of the summary to T seconds or less. Equation 11 is a constraint restricting the bias of the number of extracting sentences between documents to L sentences or less. Equation 12 is a constraint in which the parent s_{kj} of s_{ki} in the discourse dependency tree must be extracted when s_{ki} is extracted. Equation 13 is a constraint requiring other sentences in the chunk to be extracted when extracting s_{ki} in a chunk. Equations 14–16 are the constraints that set $y_{kij}^s = 1$ when s_{ki} and s_{kj} are selected.

The maximum bias in the number of extracting sentences between documents L is calculated by the following formulas based on the maximum summary length T, the number of documents N, and the average utterance time of the sentences $\bar{t}$.

Table 2 Variable definitions in the interesting sentence extraction method

x_{ki}^s	Whether sentence s_{ki} is selected
y_{kij}^s	Whether both s_{ki} and s_{kj} are selected
r_{kij}^s	Similarity between s_{ki} and s_{kj}
t_{ki}^s	Utterance time of s_{ki} (seconds)
T	Maximum summary length (seconds)
L	Maximum bias in the number of extracting sentences between documents
$f_k(i)$	Function that returns the parent ID of s_{ki}
D_N^u	IDs of the selected N documents for user u
S_k	Sentence IDs contained in document d_k
C_{km}	Sentence IDs contained in chunk m of d_k

$$L = \left\lfloor \frac{\bar{n}}{\sqrt{N}} + 0.5 \right\rfloor \tag{17}$$

$$\bar{n} = \frac{T}{\bar{t} \times N} \tag{18}$$

$\bar{n}$ represents the expected number of sentences to be extracted from one document. L is the value obtained by dividing $\bar{n}$ by the square root of the number of documents and rounding the result.

4 Experiments

4.1 Discourse Analysis

4.1.1 Experimental Setup

We used the pre-trained BERT model published by the National Institute of Information and Communications Technology[1] (NICT-BERT). This model was trained using $\mathrm{BERT_{BASE}}$ [10] with a vocabulary size of 100,000, which was inputted with text that MeCab[2] [18] morphologically analyzed using the Juman dictionary for all Japanese Wikipedia articles. The dimensions of the GRU hidden layer and linear layer were 128. Adam [19] was used as the optimizer. The evaluation was performed by a tenfold cross-validation, where 9/10 of the articles in each genre were used as training data (1080 articles) and the remaining 1/10 was used as test data (120 articles).

Since the number of discourse relations in the dataset was biased, the classification performance of infrequent discourse relations deteriorated when all the data were used. To suppress the influence of this bias, the sequences of the root-to-leaf sentences of the articles containing at least one target discourse relation were used as a dataset for each discourse relation. *Start* and *Supplement* were excluded as evaluation targets because *Start* is automatically given to sentences whose parents are the root node and *Supplement* is given to those not classified into other discourse relations.

We used articles that contained at least one chunk. The evaluation was performed based on two viewpoints: chunk detection performance and chunk sentence detection performance. The chunk detection performance is F_1 [20] when all the chunk ranges match. The chunk sentence detection performance is F_1 of the I label when the B and E labels are aggregated into the I label.

[1] https://alaginrc.nict.go.jp/nict-bert/index.html.

[2] https://taku910.github.io/mecab/.

Table 3 Inter-sentence dependency parsing (accuracy)

w/ Sentence position features	0.768
w/o Sentence position features	0.717
Parent is the previous sentence	0.618

Table 4 Chunk detection (F_1)

	Single-task	Multi-task ($\lambda_r = 0.2, \lambda_c = 0.8$)
Chunk	0.605	0.629
Chunk sentence	0.720	0.737

Table 5 Discourse relation classification (F_1)

	Single-task	Multi-task
Result	0.465	0.497 ($\lambda_r = 0.8, \lambda_c = 0.2$)
Cause	0.615	0.640 ($\lambda_r = 0.9, \lambda_c = 0.1$)
Background	0.505	0.510 ($\lambda_r = 0.9, \lambda_c = 0.1$)
Correspondence	0.406	0.417 ($\lambda_r = 0.9, \lambda_c = 0.1$)
Contrast	0.888	0.896 ($\lambda_r = 0.5, \lambda_c = 0.5$)
Topic Change	0.678	0.696 ($\lambda_r = 0.6, \lambda_c = 0.4$)
Example	0.410	0.466 ($\lambda_r = 0.8, \lambda_c = 0.2$)
Conclusion	0.442	0.449 ($\lambda_r = 0.9, \lambda_c = 0.1$)

4.1.2 Experimental Results

Table 3 shows the accuracy of inter-sentence dependency parsing. The baseline shows the performance when the parent is the previous sentence. The model with sentence position features showed an accuracy improvement of at least 5%.

Table 4 shows the chunk detection performance. The performance of the multi-task model was maximum when $\lambda_r = 0.2, \lambda_c = 0.8$ among $(\lambda_r, \lambda_c) \in \{(0.9, 0.1), (0.8, 0.2), \ldots, (0.1, 0.9)\}$. The multi-task model had a higher performance than the single-task model. By also learning the discourse relations, the multi-task model learned that *Contrast* sentences tended to be soft chunks.

Table 5 shows the classification performance of each discourse relation. the evaluation metric was F_1. The results of the multi-task models in the table show the best one among $(\lambda_r, \lambda_c) \in \{(0.9, 0.1), (0.8, 0.2), \ldots, (0.1, 0.9)\}$. The multi-task models exhibited higher performances than the single-task models. Comparing the results of each discourse relation, *Contrast* had the highest performance because sentences with *Contrast* often start with a specific phrase such as “On the other hand.”

Table 6 Interest estimation (accuracy)

	Topic	Sentence
BERT_PA_BGRU+_GNN+	0.701	0.671
BERT_PA_BGRU_GNN	0.688	0.661
BERT_PA_BGRU	0.673	0.649
BERT_BGRU_GNN	0.657	0.630

4.2 Interest Estimation

4.2.1 Experimental Setup

We used data from 1,154 participants who met the following criteria: (1) Answer time of the 6 articles is at least 6 min but less than 20 min. (2) Age is between 20 and 60 years old. (3) Neither occupation type nor industry type is "other." (4) Occupation type is a frequent one.

We used NICT-BERT (explained in Sect. 4.1.1) as the pre-trained BERT model. The dimensions of the GRU hidden layer and the linear layer were 128. Adam was used as the optimizer. ARMAConv [17] of PyTorch Geometric[3] 1.6.3 with the default parameters was used as the GNN. The interest estimator was trained with the labels of the sentences annotated "not interested at all," "not interested," or "not interested if anything" as "0," and the labels of the sentences annotated "very interested," "interested," or "interested if anything" as "1." The evaluation was performed by the tenfold cross-validation where 9/10 of the participants' data was used as the training set, and the remaining 1/10 of the participants' data was used as the test set. Using accuracy as the evaluation metric, we compared the proposed model with the following three models. BERT_PA_BGRU_GNN removed the input of profile features before and after the GNN from BERT_PA_BGRU+_GNN+, BERT_BGRU_GNN removed the personalized attention from BERT_PA_BGRU_GNN, and BERT_PA_BGRU removed the GNN from BERT_PA_BGRU_GNN.

4.2.2 Experimental Results

Table 6 shows the experimental results. The results are divided into "topic" and "sentence." BERT_PA_BGRU+_GNN+ and BERT_PA_BGRU_GNN had higher accuracies than that of BERT_BGRU_GNN. This demonstrates the effectiveness of considering users' profiles. Furthermore, BERT_PA_BGRU_GNN had a higher accuracy than that of BERT_PA_BGRU. This demonstrates the effectiveness of considering the inter-sentence dependencies.

[3] https://pytorch-geometric.readthedocs.io/en/latest/.

4.3 Personalized Summarization

4.3.1 Experimental Setup

Using the constructed dataset, we evaluated the performance of the personalized summarization method for dialog scenario planning. We assumed a situation where each of 1,154 users would select three interesting articles from six news articles of different genres. The selected articles were summarized based on their degree of interest, which were transmitted by voice within $T' = 210$ s. Each sentence of the news articles was synthesized by AITalk 4.1[4] to calculate the duration of speech. The maximum summary length T is calculated as $T = T' - (N - 1) \times (q_d - q_s)$, where T' denotes the total utterance time of the primary plan, q_s denotes the pause between sentences, and q_d denotes the pause between documents. Here, $q_s = 1$ second and $q_d = 3$ s. The value obtained by adding q_s to the playback time of the synthesized audio file was set as t^s_{ki}. The integer linear programming problem was solved by the branch-and-cut method[5] [21, 22]. The PULP_CBC_CMD solver of the PuLP[6] 2.4, which is a Python library for linear programming optimization, was used.

The summaries generated by BERT_PA_BGRU+_GNN+, which was trained with the dataset that we constructed in this study, are referred to as interest-based summaries. The summaries generated by BERT_BGRU, which was trained with 100 news articles annotated according to whether each sentence is important or not (Fleiss' kappa of three annotators was 0.546), are referred to as importance-based summaries. Using the data of 1,154 users, we calculated the evaluation metrics described in the next section for each user and compared the average values.

4.3.2 Evaluation Metrics

We propose a metric to evaluate the quality of information transmission. The information transmission quality is calculated based on the efficiency and coherence as

$$\mathrm{QoIT}_{\alpha,\beta,\gamma} = \alpha \times \mathrm{EoIT}_{\beta} + (1 - \alpha) \times \mathrm{CoIT}_{\gamma} \tag{19}$$

Because coherence is considered to be as important as efficiency in the news delivery task, we set $\alpha = 0.5$.

EoIT_{β} is the evaluation metric for efficiency [23]. When C is the coverage of sentences annotated as "very interested," "interested," or "interested if anything," and E is the exclusion rate of the sentences annotated as "not interested at all," "not interested," or "not interested if anything," EoIT_{β} is defined based on the weighted F-measure [20] as

[4] https://www.ai-j.jp/product/voiceplus/manual/.

[5] https://projects.coin-or.org/Cbc.

[6] https://coin-or.github.io/pulp/.

$$\mathrm{EoIT}_\beta = \frac{\left(1+\beta^2\right) \times C \times E}{\beta^2 \times C + E} \tag{20}$$

When $\beta = 2$, the exclusion rate is twice as important as the coverage. Compared to textual media, which allows readers to read at their own pace, dialog-based media does not allow users to skip unnecessary information or skim only necessary information while listening. Consequently, we assumed that the exclusion rate is more important than the coverage in information transmission by spoken dialog and set $\beta = 2$.

CoIT_γ is the evaluation metric for coherence. When A_d is the accuracy of dependency parsing and F_c is the F-measure of the chunk sentence detection, the CoIT_γ is defined as

$$\mathrm{CoIT}_\gamma = \gamma \times A_d + (1-\gamma) \times F_c \tag{21}$$

We set $\gamma = 0.8$ because we assumed that correctness of the dependency is more important than the correctness of the chunk.

4.3.3 Experimental Results

Table 7 compares the performance when the actual values of the dataset are given and the performance when the predicted values of the models are given. (1) represents the performance when the sentence interest, topic interest, and discourse structure are all ideal. On the other hand, (5) shows the performance when these are estimated by the proposed models. Comparing (1) and (2) or (1) and (4), revealed at 6–7% difference between the ideal and predicted values of the QoIT when the discourse structure was accurate. On the other hand, when the discourse structure was a prediction, the difference between the ideal and predicted values of the QoIT was almost 20% by comparing (1) and (5). To realize high-quality information transmission, improving the performance of the discourse analysis models should be prioritized. Finally, comparing (2) and (3) verified that the interest-based summaries are more efficient than the importance-based summaries.

5 Conclusion

We proposed a method to generate a personalized summary that may be of interest to each user based on the discourse structure of documents in order to deliver a certain amount of coherent and interesting information for a spoken news dialog system. We constructed a news article corpus with annotations of the discourse structure, users' profiles, and interests in sentences and topics. Our experiments confirmed that the personalized summaries generated by the proposed method transmit information more efficiently than generic summaries generated based on the importance of sentences.

Table 7 Information transmission quality of the summaries ($N = 3$, $T' = 210$)

	Model settings			Evaluation metrics						
	Sentence interest	Topic interest	Discourse	$\mathrm{QoIT}_{0.5,2,0.8}$	EoIT_2	C	E	$\mathrm{CoIT}_{0.8}$	A_d	F_c
(1)	Ideal	Ideal	Ideal	0.855	0.711	0.538	0.806	1.00	1.00	1.00
(2)	Prediction	Ideal	Ideal	0.796	0.592	0.475	0.666	1.00	1.00	1.00
(3)	Importance	Ideal	Ideal	0.779	0.559	0.460	0.628	1.00	1.00	1.00
(4)	Prediction	Prediction	Ideal	0.789	0.579	0.486	0.637	1.00	1.00	1.00
(5)	Prediction	Prediction	Prediction	0.667	0.578	0.489	0.632	0.757	0.767	0.717

In the future work, we plan to devise a method to adaptively generate personalized summaries using the dialog history.

Acknowledgements This work was supported by Japan Science and Technology Agency (JST) Program for Creating STart-ups from Advanced Research and Technology (START), Grant Number JPMJST1912 "Commercialization of Socially-Intelligent Conversational AI Media Service."

References

1. Sappelli M, Chu DM, Cambel B, Graus D, Bressers P (2018) SMART journalism: personalizing, summarizing, and recommending financial economic news. In: The Algorithmic Personalization and News (APEN18) Workshop at ICWSM 18(5):1–3
2. Mani I, Bloedorn E (1998) Machine learning of generic and user-focused summarization. In: Proceedings of the 15th national/10th conference on artificial intelligence/innovative applications of artificial intelligence, pp 820–826
3. Díaz A, Gervás P (2007) User-model based personalized summarization. Inf Process Manage 43(6):1715–1734
4. Yan R, Nie JY, Li X (2011) Summarize what you are interested in: an optimization framework for interactive personalized summarization. In: Proceedings of the 2011 conference on empirical methods in natural language processing, pp 1342–1351
5. Hu P, Ji D, Teng C, Guo Y (2012) Context-enhanced personalized social summarization. In: Proceedings of the 24th international conference on computational linguistics, pp 1223–1238
6. Hirao T, Nishino M, Yoshida Y, Suzuki J, Yasuda N, Nagata M (2015) Summarizing a document by trimming the discourse tree. IEEE/ACM Trans Audio, Speech Lang Process 23(11):2081–2092
7. Kikuchi Y, Hirao T, Takamura H, Okumura M, Nagata M (2014) Single document summarization based on nested tree structure. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, pp 315–320
8. Xu J, Gan Z, Cheng Y, Liu J (2020) Discourse-aware neural extractive text summarization. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 5021–5031
9. Takatsu H, Fukuoka I, Fujie S, Hayashi Y, Kobayashi T (2018) A spoken dialogue system for enabling information behavior of various intention levels. J Jpn Soc Artif Intell 33(1):1–24
10. Devlin J, Chang MW, Lee K, Toutanova K (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, pp 4171–4186
11. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(11):2673–2681
12. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (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, pp 1724–1734
13. Zhang X, Cheng J, Lapata M (2017) Dependency parsing as head selection. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics, pp 665–676
14. Lin Z, Feng M, dos Santos CN, Yu M, Xiang B, Zhou B, Bengio Y (2017) A structured self-attentive sentence embedding. In: Proceedings of the 5th international conference on learning representations, pp 1–15
15. Oh JH, Torisawa K, Hashimoto C, Kawada T, Saeger SD, Kazama J, Wang Y (2012) Why question answering using sentiment analysis and word classes. In: Proceedings of the 2012

joint conference on empirical methods in natural language processing and computational natural language learning, pp 368–378
16. Wu C, Wu F, An M, Huang J, Huang Y, Xie X (2019) NPA: neural news recommendation with personalized attention. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2576–2584
17. Bianchi FM, Grattarola D, Livi L, Alippi C (2021) Graph neural networks with convolutional ARMA filters. IEEE Trans Pattern Anal Mach Intell
18. Kudo T, Yamamoto K, Matsumoto Y (2004) Applying conditional random fields to Japanese morphological analysis. In: Proceedings of the 2004 conference on empirical methods in natural language processing, pp 230–237
19. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Proceedings of the 3rd international conference for learning representations, pp 1–15
20. Chinchor N (1992) MUC-4 evaluation metrics. In: Proceedings of the 4th conference on message understanding, pp 22–29
21. Mitchell JE (2002) Branch-and-cut algorithms for combinatorial optimization problems. In: Handbook of applied optimization, pp 65–77
22. Padberg M, Rinaldi G (1991) A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Rev 33(1):60–100
23. Takatsu H, Okuda M, Matsuyama Y, Honda H, Fujie S, Kobayashi T (2021) Personalized extractive summarization for a news dialogue system. In: Proceedings of the 8th IEEE spoken language technology workshop, pp 1044–1051

Empathetic Dialogue Generation with Pre-trained RoBERTa-GPT2 and External Knowledge

Ye Liu, Wolfgang Maier, Wolfgang Minker, and Stefan Ultes

Abstract One challenge for dialogue agents is to recognize the feelings of the conversation partner and respond accordingly. In this work, RoBERTa-GPT2 is proposed for empathetic dialogue generation, where the pre-trained auto-encoding RoBERTa is utilized as encoder and the pre-trained auto-regressive GPT-2 as decoder. With the combination of the pre-trained RoBERTa and GPT-2, our model realizes a new state-of-the-art emotion accuracy. To enable the empathetic ability of RoBERTa-GPT2 model, we propose a commonsense knowledge and emotional concepts extractor, in which the commonsensible and emotional concepts of dialogue context are extracted for the GPT-2 decoder. The experiment results demonstrate that the empathetic dialogue generation benefits from both pre-trained encoder-decoder architecture and external knowledge.

1 Introduction

With the development of Spoken Dialogue Systems (SDSs), people are no longer satisfied with task-oriented interaction, like booking a train ticket or making a reservation; but are additionally interested in chit-chat communication. An expected trait of chit-chat agents is to be able to identify the user's emotion and express their empathy. For instance, the psychology study in [41] shows that talking about an emotional experience to someone and sharing their emotions contributes to emotional recov-

Y. Liu (✉) · W. Maier · S. Ultes
Mercedes-Benz AG, Sindelfingen, Germany
e-mail: ye.y.liu@mercedes-benz.com

W. Maier
e-mail: wolfgang.mw.maier@mercedes-benz.com

S. Ultes
e-mail: stefan.ultes@mercedes-benz.com

W. Minker
Ulm University, Ulm, Germany
e-mail: wolfgang.minker@uni-ulm.de

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_5

Table 1 One empathetic dialogue in EmpatheticDialogues dataset

Emotion	Lonely
Situation	All my friends live in a different country
Speaker	Hi, I feel so lonely sometimes because all my friends live in a different country
Listener	Oh, I'm sure you are lonely. Maybe you can join some kind of club that lets you meet new friends?
Speaker	I was thinking about it! I wanted to join a group for local moms
Listener	That's a good idea! This way you can also meet friends for yourself, but also maybe meet new friend's for your children to hang out with while you do with their moms!

ery from the event. Hence, exactly identifying the user's emotion and appropriately expressing their empathy will be a desired trait for SDSs.

Table 1 shows an empathetic dialogue from the EmpatheticDialogues dataset [27]. A speaker tells a listener the lonely situation that they are facing, and the listener tries to understand the speaker's feelings and responds accordingly. Even though sharing emotional experiences is a general manifestation for humans, it is a great challenge to train a chit-chat agent capable to understand the user's emotion and respond empathetically.

Several works with Transformer-based encoder-decoder architecture [36] have been presented for empathetic dialogue generation, such as the multi-task learning [26, 27, 37] or mixture of experts [16]. However, the combination of a pre-trained auto-encoding encoder and a pre-trained auto-regressive decoder has not been explored for empathetic dialogue generation. In this work, the pre-trained Robustly optimized BERT approach (RoBERTa) [18] as encoder and the pre-trained Generative Pre-trained Transformer (GPT-2) [25] as decoder: RoBERTa-GPT2 encoder-decoder architecture is presented for empathetic dialogue generation. The experiments with the EmpatheticDialogues dataset show that the combination of RoBERTa and GPT-2 highly improves the emotion recognition ability and realizes a new state-of-the-art emotion accuracy.

In addition to the advanced neural network architecture, some external knowledge also contributes to the empathetic dialogue generation. Humans generally understand the world and express implicit emotions based on their experience and knowledge. Also, [39] demonstrates that commonsense knowledge is fundamental for chit-chat agents to understand conversations and generate appropriate responses. As shown in Fig. 1, the underlying commonsensible and emotional concepts of the speaker's utterance can help the listener to better understand what the speaker is talking about. Hence, we propose an Commonsense Knowledge and Emotional Concepts Extractor (CKECE) for GPT-2 decoder in our work, to enable commonsense and empathetic response generation. In the CKECE, we firstly utilize KeyBERT [7] to extract the keywords from the dialogue context; then elicit the commonsensible and emotional

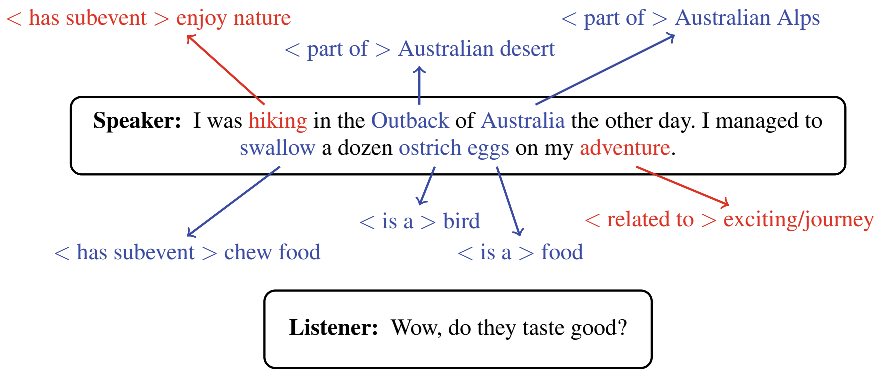


Fig. 1 An example of EmpatheticDialogues dataset with underlying commonsense knowledge (blue part) and emotional concepts (red part). (The special token in < > represents the relation in commonsense knowledge: ConcepetNet [34].)

concepts of the keywords based on commonsense knowledge: ConceptNet [34] and emotion lexicon: NRC_VAD [20]; and finally, the extracted concepts are fed into GPT-2 decoder in a more plain text format to guide the empathetic generation.

2 Related Work

Open-domain and chit-chat conversational models have been widely studied [30, 38]. With the rise of public accessible datasets [9, 15, 27] and data-driven learning approaches [35, 36], several works have attempted to make chit-chat dialogue more engaging. Some aim to improve the personalization of responses by conditioning the generation on a persona profile [11]. Then the PersonaChat dataset [42] was particularly introduced and the competition in ConvAI 2 challenge [5] demonstrated that the produced responses include more consistent personas by adding persona information into the model. However, the personalized dialogue models often cannot take the feelings of their conversation partners into consideration. Besides the chit-chat research on displaying a consistent personality, some works focus on emotional and empathetic dialogue generation. The existing emotional dialogue models [3, 12, 32, 45, 46] generally generate the response depending on a pre-defined emotion; however, the empathetic dialogue models are capable of perceiving the emotion of the speaker and express their empathy without extra step to determine which emotion type to respond explicitly [33]. Hence, the empathetic dialogue model is more in line with the real-world scenarios [14], because the listener is capable to infer the emotion of the speaker in human-human communication.

In recent years, several works have been presented for empathetic dialogue generation. Reference [27] created a benchmark and dataset towards empathetic open-domain dialogue. Reference [16] softly combined the possible emotional responses from several separate experts to generate the final empathetic response. Reference [13] proposed an multi-resolution interactive empathetic dialogue model to evoke more emotional perception in dialogue generation. Reference [14] proposed a multi-type knowledge-aware empathetic dialogue generation framework to enhance the empathy of generations. The above-mentioned approaches are all trained from scratch. Reference [21] proposed BERT2BERT for Arabic empathetic response generation, while the encoder and decoder are both warm started using pre-trained auto-encoding AraBERT [1] parameters. Reference [40] introduced EmpTransfo and [17] presented CAiRE, both are empathetic-aware model adapted from GPT [24]. With the release of the encoder-decoder model in Huggingface,[1] where any pre-trained auto-encoding model as the encoder and any pre-trained auto-regressive model as the decoder can be initialized as a sequence-to-sequence model, we are more interested in the performance of pre-trained auto-encoding encoder and auto-regressive decoder architecture for empathetic dialogue generation. Furthermore, [28] performed an extensive study on leveraging variable pre-trained models for sequence generation tasks and demonstrated that combining RoBERTa [18] and GPT-2 [25] achieves strong results. Hence, RoBERTa-GPT2 is proposed in this work for empathetic dialogue generation.

In addition, the corpora with emotion labelling play a significant role in empathetic dialogue generation. There are several interesting resources. Reference [15] developed the DailyDialog dataset, with manual emotion labelling to each utterance. Reference [9] collected the EmotionLines dataset from TV shows and human-to-human chats, where each utterance is further annotated with one of seven emotion-categorical labels. However, only 5% of the utterances in DailyDialog and 16.68% in EmotionLines have varied emotional labels and others are either "none" or "happy" labels. Hence, they are not suitable for empathetic dialogue generation because of the extremely unbalanced data distribution. Reference [27] released an empathetic dialogue dataset: EmpatheticDialogues, which focuses explicitly on conversations about emotionally grounded personal situations and considers a richer, evenly distributed set of emotions. In our work, we conduct the experiment of empathetic dialogue generation with the EmpatheticDialogues dataset.

3 The Proposed Method

In this work, we present the RoBERTa-GPT2 encoder-decoder architecture for empathetic dialogue generation, where the pre-trained auto-encoding RoBERTa [18] as encoder and pre-trained auto-regressive GPT-2 [25] as decoder. In addition, a Commonsense Knowledge and Emotional Concepts Extractor (CKECE), which is used to extract the relevant concepts from dialogue history, is proposed to enable the commonsensible and empathetic ability of the GPT-2 decoder. In this section, the

[1] https://huggingface.co/transformers/model_doc/encoderdecoder.html.

CKECE will be firstly introduced and then the RoBERTa-GPT2 architecture with extracted concepts will be shown.

3.1 Commonsense Knowledge and Emotional Concepts Extractor: CKECE

For the CKECE, two knowledge sources the commonsense knowledge ConceptNet [34] and the emotional lexicon NRC_VAD [20], and one keyword extraction tool, KeyBERT [7], are used. We firstly utilize the KeyBERT to extract the keywords of the dialogue context and then filter out the most relevant commonsense knowledge and emotional concepts of the keywords with the confidence score of ConceptNet and emotional intensity of NRC_VAD.

3.1.1 The CKECE Components

The three resources used in CKECE are introduced in the following:

KeyBERT[2] is a minimal and easy-to-use keyword extraction technique that leverages BERT embeddings and cosine similarity to find the keywords and key phrases in a document that are the most similar to the document itself.

ConceptNet[3] is a large-scale and multilingual commonsense knowledge graph that describes general human knowledge in natural language. It comprises 5.9M assertions, 3.1M concepts, and 38 relations. The nodes in ConceptNet are concepts and the edges are relations. Each triplet is an assertion. Each assertion is associated with a confidence score. The assertion confidence score is usually in the [1, 10] interval. For example, (*loneliness*, *CausesDesire*, *socialize*) with confidence score of 3.464.

NRC_VAD[4] is a lexicon that includes a list of more than 20k English words and their Valence, Arousal, and Dominance (VAD) scores. For a given word and a dimension, the scores range from 0 (lowest) to 1 (highest). The interpretations of NRC_VAD dimensions are presented in Table 2. Such as the VAD score vector $[V_a, A_r, D_o]$ of word "happiness" is [0.960, 0.732, 0.850].

3.1.2 CKECE

To extract more relevant concepts, we firstly utilize the KeyBERT to extract the keywords from the dialogue context. In this step, the recommended KeyBERT model "distilbert-base-nli-mean-tokens" is used and only maximal top 10 keywords with

[2] https://github.com/MaartenGr/KeyBERT.

[3] https://conceptnet.io/.

[4] https://saifmohammad.com/WebPages/nrc-vad.html.

Table 2 Interpretations of NRC_VAD dimensions

Dimensions	Values	Interpretations
Valence (V_a)	[0, 1]	Negative-Positive
Arousal (A_r)	[0, 1]	Calm-Excited
Dominance (D_o)	[0, 1]	Weak-Powerful

a score larger than 0 are retained. An example of extracted keywords is shown in Fig. 2.

Then, we pick out the commonsense concepts from ConceptNet based on the keywords and denote them in a tuple (keyword, relation, concept, scaled confidence score) as $\{\tau_k^i = (k_i, r_k^i, c_k^i, s_k^i)\}_{k=1,2,\ldots,K}$ where the confidence score s is scaled by the following Eq. 1 $min - max$ normalization:

$$min - max(s) = \frac{s - min_s}{max_s - min_s}, \tag{1}$$

where min_s is 1 and max_s is 10. The processed $s \in [0, 1]$ and the $min - max$ normalization is also used in [14, 44]. With $min - max$ normalization, the example (*loneliness*, *CausesDesire*, *socialize*) with confidence score 3.464 in Sect. 3.1.1 is transformed into (loneliness, CausesDesire, socialize, 0.274) tuple with scaled confidence score 0.274. In order to pick out the most relevant concepts, the following tuples will be removed in this step:

- The keywords or concepts are stop words. (The union of stop words in NLTK [19] and SpaCy[5] is used.)
- The scaled confidence score is less than a pre-defined threshold α. We set α is 0.1 in this work, i.e. $s < 0.1$.
- The keywords and concepts are same or have the same stem. Like (addition, Synonym, addition, 0.11); (actual, DerivedFrom, actually, 0.11).
- The relation is in an excluded relation list, i.e. $r \in [Antonym, ExternalURL, NotDesires, NotHasProperty, NotCapableOf, dbpedia, DistinctFrom, EtymologicallyDerivedFrom, EtymologicallyRelatedTo, SymbolOf, FormOf, AtLocation, DerivedFrom, SymbolOf]$

Furthermore, to enable the emotional concepts, we adopt NRC_VAD to compute emotion intensity values for the concepts c as Eq. 2.

$$\eta(c) = min - max\left(\|V_a(c) - \frac{1}{2}, \frac{A_r(c)}{2}\|_2\right), \tag{2}$$

[5] https://github.com/explosion/spaCy.

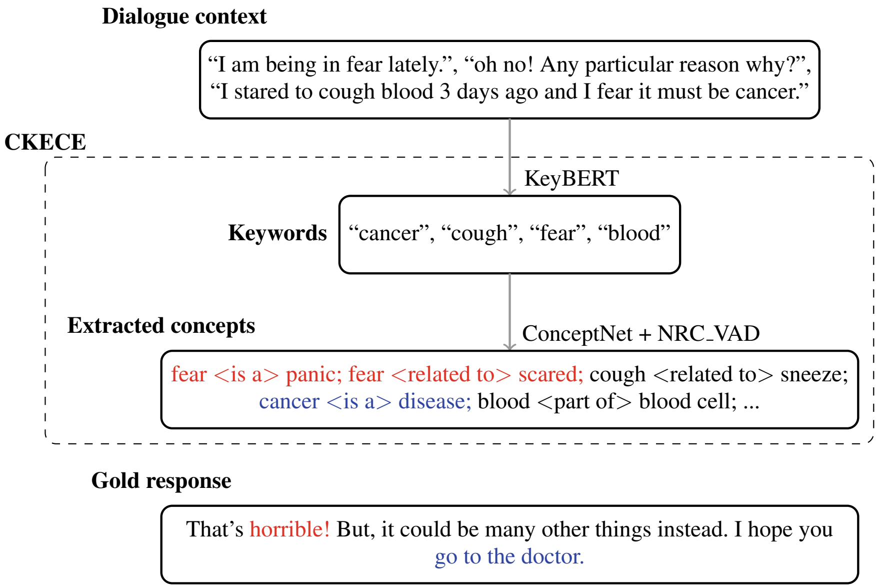


Fig. 2 An example for the process of CKECE for the dialogue context. The extracted emotional concepts and emotional words in gold response are marked in red. The blue part in extracted concepts and gold response share the same commonsense knowledge

where $\|\cdot\|_k$ denotes l_k norm. $V_a(c)$ and $A_r(c)$ represent valence and arousal score of concept c_i, respectively. When c not in NRC_VAD, we set $\eta(c)$ to the mid value of 0.5.

Lastly, the final score f in Eq. 3 is derived from three aspects: emotion intensity, semantic similarity, and scaled confidence score. The semantic similarity $cos(k_i, c_k^i)$ is the cosine similarity between keyword and concept both embedded by the GloVe [23], which stands for global vectors for word representation and is an unsupervised learning algorithm for obtaining vector representations for words.

$$f(\tau_k^i) = \eta(c_k^i) + cos(k_i, c_k^i) + s_k^i \ . \tag{3}$$

We sort the candidate tuples in descending order of the final scores and select the top three tuples for each keyword. Maximal 10 tuples are chosen for every dialogue context. Then the extracted concepts are arranged in a more plain textual form: “keyword <relation> concept”, which is shown in Fig. 2, for GPT-2 decoder.

3.2 *Pre-trained RoBERTa-GPT2 Encoder-Decoder*

The RoBERTa [18] and GPT-2 [25] are both large architectures pre-trained on large collections of texts. Then the pre-trained models are widely fine-tuned in downstream tasks. In this work, we explore the pre-trained RoBERTa-GPT2 as encoder-decoder architecture for empathetic dialogue generation.

3.2.1 The Preliminaries of RoBERTa-GPT2

The pre-trained auto-encoding RoBERTa and pre-trained auto-regressive GPT-2 are introduced in the following:

RoBERTa[6] has the same architecture as BERT [4], but uses a byte-level Byte-Pair Encoding (BPE) [29] as a tokenizer (same as GPT-2) and improved the training procedure of BERT [4].

GPT-2[7] is a pre-trained large-scale unsupervised language model which generates coherent paragraphs of text. GPT-2 is also widely used in task-oriented dialogue generation [2, 22] and chit-chat dialogue generation [17, 43].

3.2.2 RoBERTa-GPT2

Figure 3 shows our proposed RoBERTa-GPT2 encoder-decoder architecture for empathetic dialogue generation. The simplified input for RoBERTa encoder and GPT-2 decoder in Fig. 3 only shows the initial part of the sentences. And Figs. 2 and 3 share the same dialogue example.

The pre-trained RoBERTa as encoder processes the dialogue context, where the $<$ CLS $>$ token is appended in the first place and $<$ SEP $>$ is for separating speaker utterance and listener utterance. The output of $<$ CLS $>$ token, pooled output, represents the entire meaning of the input. A linear layer with softmax activation is added on the top of pooled output for emotion classification. The encoder outputs will be fed to the GPT-2 decoder for the cross-attention mechanism. As shown in Fig. 3, the input for GPT-2 decoder starts with extracted concepts. During the training, the gold response is also attached after concepts for faster convergence and separated by $<$SEP$>$ token. It is noteworthy that only the response part without extracted concepts is the output of GPT-2 decoder for computing the generation loss during the training. That means, the response is generated conditioned on the contextual information of encoder outputs with cross-attention mechanism and emotional concepts of decoder inputs with the self-attention mechanism by combining pre-trained RoBERTa and GPT-2. Lastly, all the parameters of RoBERTa-GPT2 are jointly trained end-to-end to optimize the emotion classification and response generation by minimizing emotion cross entropy loss and maximum likelihood estimator (MLE) generation loss.

---

[6] https://github.com/pytorch/fairseq/tree/master/examples/roberta.

[7] https://github.com/openai/gpt-2.

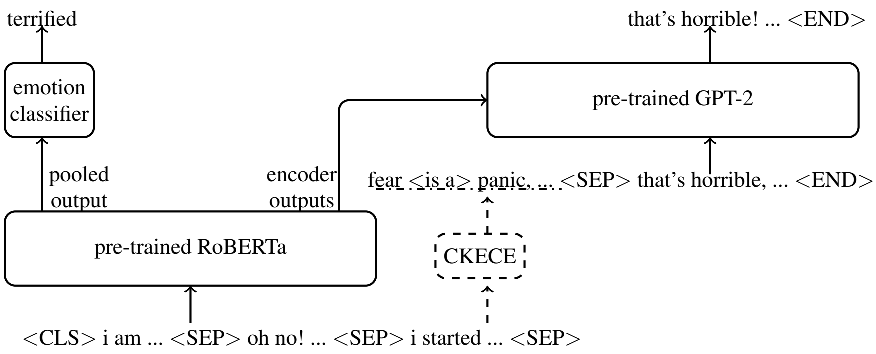


Fig. 3 Our proposed RoBERTa-GPT2 encoder-decoder architecture with CKECE guidance for empathetic dialogue generation

4 Experimental Settings and Results Analysis

4.1 Dataset

We conduct our experiment on the large-scale multi-turn EmpatheticDialogues [27], which consists of 25 k one-to-one open-domain conversation grounded in emotional situations. And the EmpatheticDialogues dataset provides 32 evenly distributed emotion labels.

4.2 Baselines

We compare our models with the following four baselines:

(1) **Transformer** [36]: a Transformer-based encoder-decoder model trained with MLE generation loss.
(2) **EmoPrepend-1** [27]: an extension of Transformer model with an additional supervised emotion classifier. The whole model is jointly trained by optimizing both the classification and generation loss.
(3) **MoEL** [16]: another extension of the Transformer model, which softly combines the outputs of the multiple listeners. Each listener is optimized to react to a certain emotion and generate an empathetic response.
(4) **MK-EDG** [14]: a multi-type knowledge-aware empathetic dialogue generation framework. Commonsense knowledge and emotional lexicon are used to enrich the dialogue utterance.

Additionally, to better analyse our proposed RoBERTa-GPT architecture for empathetic dialogue model, we also conducted **RoBERTa w/o GPT-2**: only RoBERTa encoder with emotion classifier trained with emotion loss; and **RoBERTa-GPT2 w/o CKECE**: RoBERTa-GPT2 without the guidance of external knowledge.

4.3 Training Details

The RoBERTa-GPT2 is trained with batch size 16 and learning rate 1e−5. Early stopping is applied during the training for saving the best model. During decoding, we use the top-k [6] and nucleus sampling (top-p) [8] decoding algorithms with top-k equal to 5 and top-p equal to 0.9.

4.4 Automatic Evaluation Results

To evaluate the performance of RoBERTa-GPT2 model, we firstly adopt Emotion Accuracy as the agreement between the ground truth emotion labels and the predicted emotion labels by the emotion classifier. In addition, Perplexity [31] values are utilized to measure the high-level general quality of the generation model. Furthermore, Distinct-1 and Distinct-2 [10] are used to measure the proportion of the distinct unigrams and bigrams in all the generated results to indicate diversity. Table 3 shows the evaluation results between our proposed methods and baselines. The results of MK-EDG in Table 3 are directly copied from [14]; hence, MK-EDG is absent from use cases in Table 4.

In Table 3, we observe our proposed RoBERTa-GPT2 outperforms the baselines by a large margin in terms of all automatic metrics and realizes a new state-of-the-art result. The emotion accuracy comparison between only RoBERTa encoder w/o GPT-2 and RoBERTa-GPT2 demonstrates that emotion classification benefits from the combination of RoBERTa encoder and GPT-2 decoder architecture. The RoBERTa-

Table 3 Evaluation results between RoBERTa-GPT2 and baselines

Models	Emotion accuracy↑	Perplexity↓	Distinct-1(%)↑	Distinct-2(%)↑
Transformer	–	35.56	0.41	1.49
EmoPrepend-1	0.3359	35.66	0.42	1.62
MoEL	0.3425	37.69	0.43	1.72
MK-EDG	0.3931	34.85	1.48	4.90
RoBERTa w/o GPT-2	0.3439	–	–	–
RoBERTa-GPT2 w/o CKECE	**0.5262**	14.97	1.62	10.47
RoBERTa-GPT2	0.5151	**13.57**	**2.04**	**11.68**

Table 4 Generated responses from Transformer, EmoPrepend-1, MoEL, and RoBERTa-GPT2. The underline part shows the empathetic and contextual RoBERTa-GPT2 response

Emotion	Excited
Dialogue context	Speaker: I'm taking a vacation and going on a cruise to Alaska next week!
Transformer	Where are you going?
EmoPrepend-1	That is awesome! Where are you going?
MoEL	I hope you have a great time!
RoBERTa-GPT2	I am sure you will enjoy the cruise!
Emotion	Terrified
Dialogue context	**Speaker**: Do you even know how crazy it is to skydive? I must have been absolutely nuts when I did it **Listener:** Oh I don't think I could do that. I have a fear of falling from high places **Speaker:** It gave me the biggest rush that's for sure. But on the way down I was saying my prayers
Transformer	That is a great idea. I have been there
EmoPrepend-1	I would be too. I do not know what you mean
MoEL	I am sorry to hear that. I hope everything works out for you
RoBERTa-GPT2	That is a lot of stress! I am glad you did not get injured
Emotion	Excited
Dialogue context	**Speaker**: I couldn't wait to go to the concert
Transformer	That sounds like a lot of fun!
EmoPrepend-1	That sounds like a lot of fun!
MoEL	That sounds like a lot of fun!
RoBERTa-GPT2	That is awesome! What kind of concert will it be?
Emotion	Trusting
Dialogue context	**Speaker**: I let my daughter ride in the road by herself on her skateboard the other day
Transformer	That is awesome! What was it?
EmoPrepend-1	That must have been a lot of fun!
MoEL	oh no! What happened?
RoBERTa-GPT2	Wow! I would be a little nervous if she did not skate at all

GPT2 has better results on Perplexity, Distinct-1, and Distinct-2 than RoBERTa-GPT2 w/o CKECE, indicating that injecting external knowledge contributes to the generation quality, though a little loss on emotion accuracy.

4.5 Use Cases Study

Cases of generated responses from baselines Transformer, EmoPrepend-1, MoEL, and our proposed method, RoBERTa-GPT2, are listed in Table 4. In the first case, Transformer and EmoPrepend-1 do not exactly understand what the speaker is saying. MoEL identifies the user's emotion but the response is kind of generic. Besides the correct understanding of user's emotion, RoBERTa-GPT2 also knows that the speaker is talking about "cruise". The baselines in the second case do not correctly recognize the user's emotion. Compared with the generic response of the baselines in the third case, RoBERTa-GPT2 generates a contextual response with a proper positive emotion by replying with "awesome". In the fourth case, the response of EmoPrepend-1 is generic and the other two baselines do not understand the speaker, while RoBERTa-GPT2 generates a coherent and informative response by showing concern. All the cases in Table 4 show that our proposed RoBERTa-GPT2 can both handle with user emotion and dialogue content.

5 Conclusion and Outlook

In this work, we leverage pre-trained auto-encoding RoBERTa as encoder and pre-trained auto-regressive GPT-2 as decoder for empathetic dialogue generation. Meanwhile, the external knowledge, commonsense knowledge and emotional lexicon, are utilized to extract emotional and commonsensible concepts from dialogue context for GPT-2 decoder to enable empathetic and contextual responses. Both automatic metrics and the case study show that our proposed RoBERTa-GPT2 outperforms the baselines and demonstrate that the empathetic dialogue generation benefits from pre-trained modelling and external knowledge.

In future work, we will continually evaluate our proposed method for empathetic dialogue generation from the human perspective. Meanwhile, we are also interested in other flexible methods for injecting external knowledge into an empathetic dialogue system.

References

1. Antoun W, Baly F, Hajj H (2020) Arabert: transformer-based model for Arabic language understanding. In: LREC 2020 workshop language resources and evaluation conference 11–16 May 2020, p 9
2. Budzianowski P, Vulić I (2019) Hello, it's GPT-2-how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. In: Proceedings of the 3rd

workshop on neural generation and translation, pp 15–22
3. Colombo P, Witon W, Modi A, Kennedy J, Kapadia M (2019) Affect-driven dialog generation. In: Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: human language technologies, vol 1 (Long and Short Papers), pp 3734–3743
4. Devlin J, Chang MW, Lee K, Toutanova K (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, vol 1 (Long and Short Papers), pp 4171–4186
5. Dinan E, Logacheva V, Malykh V, Miller A, Shuster K, Urbanek J, Kiela D, Szlam A, Serban I, Lowe R et al (2019) The second conversational intelligence challenge (convai2), p 1902
6. Fan A, Lewis M, Dauphin Y (2018) Hierarchical neural story generation. In: ACL (1)
7. Grootendorst M (2020) Keybert: minimal keyword extraction with Bert. https://doi.org/10.5281/zenodo.4461265
8. Holtzman A, Buys J, Du L, Forbes M, Choi Y (2019) The curious case of neural text degeneration. In: International conference on learning representations
9. Hsu CC, Chen SY, Kuo CC, Huang TH, Ku LW (2018) Emotionlines: an emotion corpus of multi-party conversations. In: Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)
10. Li J, Galley M, Brockett C, Gao J, Dolan WB (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, pp 110–119
11. Li J, Galley M, Brockett C, Spithourakis G, Gao J, Dolan WB (2016) A persona-based neural conversation model. In: Proceedings of the 54th annual meeting of the association for computational linguistics (vol 1: Long Papers), pp 994–1003
12. Li J, Sun X (2018) A syntactically constrained bidirectional-asynchronous approach for emotional conversation generation. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 678–683
13. Li Q, Chen H, Ren Z, Ren P, Tu Z, Chen Z (2020) EMPDG: multi-resolution interactive empathetic dialogue generation. In: Proceedings of the 28th international conference on computational linguistics, pp 4454–4466
14. Li Q, Li P, Chen Z, Ren Z (2020) Towards empathetic dialogue generation over multi-type knowledge
15. Li Y, Su H, Shen X, Li W, Cao Z, Niu S (2017) Dailydialog: a manually labelled multi-turn dialogue dataset. In: Proceedings of the eighth international joint conference on natural language processing (vol 1: Long Papers), pp 986–995
16. Lin Z, Madotto A, Shin J, Xu P, Fung P (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), pp 121–132
17. Lin Z, Xu P, Winata GI, Siddique FB, Liu Z, Shin J, Fung P (2020) CAiRE: an end-to-end empathetic chatbot. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 13,622–13,623
18. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized BERT pretraining approach. CoRR. arXiv:1907.11692
19. Loper E, Bird S (2002) NLTK: the natural language toolkit. In: Proceedings of the ACL-02 workshop on effective tools and methodologies for teaching natural language processing and computational linguistics, pp 63–70
20. Mohammad S (2018) Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In: Proceedings of the 56th annual meeting of the association for computational linguistics (vol 1: Long Papers), pp 174–184
21. Naous T, Antoun W, Mahmoud R, Hajj H (2021) Empathetic bert2bert conversational model: learning Arabic language generation with little data. In: Proceedings of the sixth arabic natural language processing workshop, pp 164–172

22. Peng B, Zhu C, Li C, Li X, Li J, Zeng M, Gao J (2020) Few-shot natural language generation for task-oriented dialog. In: Proceedings of the 2020 conference on empirical methods in natural language processing: findings, pp 172–182
23. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
24. Radford A, Narasimhan K, Salimans T, Sutskever I Improving language understanding by generative pre-training
25. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8):9
26. Rashkin H, Smith EM, Li M, Boureau YL (2018) I know the feeling: learning to converse with empathy
27. Rashkin H, Smith EM, Li M, Boureau YL (2019) Towards empathetic open-domain conversation models: a new benchmark and dataset. In: ACL (1)
28. Rothe S, Narayan S, Severyn A (2020) Leveraging pre-trained checkpoints for sequence generation tasks. Trans Assoc Comput Linguist 8:264–280
29. Sennrich R, Haddow B, Birch A (2016) Neural machine translation of rare words with subword units. In: Proceedings of the 54th annual meeting of the association for computational linguistics (vol 1: Long Papers), pp 1715–1725
30. Serban IV, Lowe R, Charlin L, Pineau J (2016) Generative deep neural networks for dialogue: a short review
31. Serban IV, Sordoni A, Bengio Y, Courville A, Pineau J (2015) Hierarchical neural network generative models for movie dialogues 7(8):434–441. arXiv:1507.04808
32. Shen L, Feng Y (2020) CDL: curriculum dual learning for emotion-controllable response generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 556–566
33. Skowron M, Theunis M, Rank S, Kappas A (2013) Affect and social processes in online communication-experiments with an affective dialog system. IEEE Trans Affect Comput 4(3):267–279
34. Speer R, Chin J, Havasi C (2017) Conceptnet 5.5: an open multilingual graph of general knowledge. In: Proceedings of the AAAI conference on artificial intelligence, vol 31
35. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. Adv Neural Inf Process Syst 27:3104–3112
36. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: NIPS
37. Wei W, Liu J, Mao X, Guo G, Zhu F, Zhou P, Hu Y (2019) Emotion-aware chat machine: automatic emotional response generation for human-like emotional interaction. In: Proceedings of the 28th ACM international conference on information and knowledge management, pp 1401–1410
38. Wolf T, Sanh V, Chaumond J, Delangue C (2019) Transfertransfo: a transfer learning approach for neural network based conversational agents
39. Young T, Cambria E, Chaturvedi I, Zhou H, Biswas S, Huang M (2018) Augmenting end-to-end dialogue systems with commonsense knowledge. In: Proceedings of the AAAI conference on artificial intelligence, vol 32
40. Zandie R, Mahoor MH (2020) Emptransfo: a multi-head transformer architecture for creating empathetic dialog systems. In: The thirty-third international flairs conference
41. Zech E, Rimé B (2005) Is talking about an emotional experience helpful? effects on emotional recovery and perceived benefits. Clin Psychol Psychother: Int J Theory Pract 12(4):270–287
42. Zhang S, Dinan E, Urbanek J, Szlam A, Kiela D, Weston J (2018) Personalizing dialogue agents: i have a dog, do you have pets too? In: ACL (1)
43. Zhang Y, Sun S, Galley M, Chen YC, Brockett C, Gao X, Gao J, Liu J, Dolan WB (2020) Dialogpt: large-scale generative pre-training for conversational response generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics: system demonstrations, pp 270–278

44. Zhong P, Wang D, Miao C (2019) Knowledge-enriched transformer for emotion detection in textual conversations. 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), pp 165–176
45. Zhou H, Huang M, Zhang T, Zhu X, Liu B (2018) Emotional chatting machine: emotional conversation generation with internal and external memory. In: Thirty-second AAAI conference on artificial intelligence
46. Zhou X, Wang WY (2018) Mojitalk: generating emotional responses at scale. In: Proceedings of the 56th annual meeting of the association for computational linguistics (vol 1: Long Papers), pp 1128–1137

Towards Handling Unconstrained User Preferences in Dialogue

Suraj Pandey, Svetlana Stoyanchev, and Rama Doddipatla

Abstract A user input to a schema-driven dialogue information navigation system, such as venue search, is typically constrained by the underlying database which restricts the user to specify a predefined set of preferences, or slots, corresponding to the database fields. We envision a more natural information navigation dialogue interface where a user has flexibility to specify unconstrained preferences that may not match a predefined schema. We propose to use information retrieval from unstructured knowledge to identify entities relevant to a user request. We construct an up-to-date database of restaurants in Cambridge, including unstructured knowledge snippets (reviews and information from the web) and annotate a set of query-snippet pairs with relevance labels. We use the annotated dataset to train and evaluate snippet relevance classifiers, as a proxy to evaluating recommendation accuracy. We show that with a pretrained transformer model as an encoder, an unsupervised/supervised classifier achieves a weighted F1 of 0.661/0.856.

Keywords Dialogue · Open-domain · NLP · Unconstrained input

1 Introduction

A conversation is a natural user interface for accessing information. In *information navigation* dialogue, such as search for restaurants, hotels or tourist attractions, a user specifies search constraints and navigates over search results using text-based, spoken, or multi-modal interface. Information navigation tasks are typically handled with a schema-driven approach [1–3]. While a schema-driven system may be

S. Pandey · S. Stoyanchev (✉) · R. Doddipatla
Toshiba Europe Ltd., 208 Cambridge Science Park Milton Rd, Cambridge CB4 0GZ, England
e-mail: svetlana.stoyanchev@crl.toshiba.co.uk

S. Pandey
e-mail: suraj.pandey@open.ac.uk

R. Doddipatla
e-mail: rama.doddipatla@crl.toshiba.co.uk

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_6

effectively bootstrapped for a new application using the structure and content of the corresponding database, a user is limited in the range of constraints they may specify. For example, in a restaurant search domain with the schema fields *area*, *price range* and *food type*, a user may specify any combination of these fields but not others [4]. To evaluate a schema-driven information navigation system, the recruited experiment participants are typically given a conversation 'goal' based on the database schema, e.g. *'you are looking for a cheap Italian place in the center'*, and are instructed to retrieve matching venues using spoken or text chat interaction. These instructions guide the user to specify in-domain preferences that can be handled by the dialogue system. An initial user request based on the above 'goal' may be:

User: *'A cheap Italian restaurant in the centre'*

User: *'I am looking for a cheap place'*

User: *'A restaurant in the center'*

User: *'An Italian restaurant in cheap price range'*[1]

User: ...

While there is variability in the natural language utterances of a recruited user for a predefined 'goal', the preference type specified in the goal is limited to the schema.

Unlike the recruited subjects, real users come up with personalized search preferences and formulate their requests to the system without a bias of the instructions. The challenge is that a user request with preferences outside of the domain schema, e.g. *'Find me a cosy family friendly pub that serves pizza.'* cannot be handled by a purely schema-driven system. The majority of user queries (75 out of 105) that we collected without priming the user by a predefined 'goal' do not mention any of the domain-specific schema fields, indicating that a purely schema-driven system is not sufficient to handle naturally constructed user requests. In contrast to a schema-driven dialogue system, a search interface handles such unconstrained user queries using information retrieval methods from unstructured (text) data. Search interfaces, however, are not interactive and do not handle query changes or follow-up questions. In this work, we aim to improve the naturalness of user interaction with an information navigation dialogue system. We propose to extend the schema-driven dialogue system to handle out-of-schema user queries by incorporating an entity retrieval module in the dialogue system pipeline.

An entity retrieval model requires domain-specific data to extract the requested information. For this study, we create an annotated dataset for the Cambridge Restaurants domain following the process outlined in Fig. 1. We first collect a database from the Web, including text snippets composed from restaurant reviews and descriptions along with a set of unconstrained restaurant search queries. Next, unsupervised and transfer learning methods are used to obtain a set of query-snippet pairs which are then annotated using Amazon Mechanical Turk. The resulting annotated dataset is then used to build supervised relevance scoring models and compared with the performance of unsupervised and transfer learning approaches. We show that using a

---

[1] A system may ask to narrow down the search criteria and a user may specify additional preferences in consecutive turns.

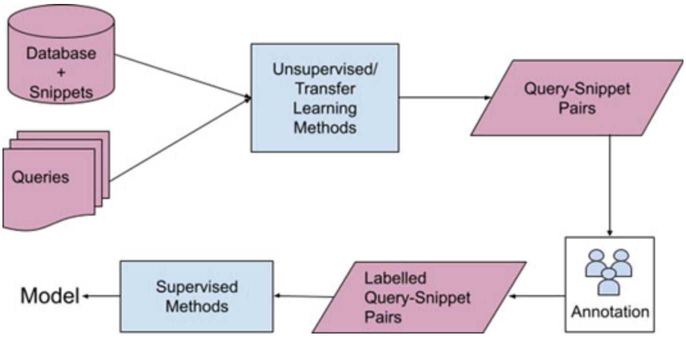


Fig. 1 Process for building a supervised relevance scoring model

pretrained transformer model as an encoder, an unsupervised/supervised classifier of the snippet relevance to the query achieves a weighted F1 of .661/.856.

The contributions of this paper are as follows:

- A methodology for extending a schema-driven dialogue system to support natural user preferences.
- A manually annotated dataset with 1.7K query and text snippet pairs.
- Evaluation of supervised, unsupervised and transfer learning approaches for snippet relevance classification.

The rest of the paper is organized as follows. In Sect. 2, we outline related research. In Sect. 3, we present an extended Cambridge Restaurants 2021 dataset and annotated Restaurant Query Snippet dataset (ResQS). In Sect. 4, we describe the approaches used to detect text snippets relevant to a query and present experimental results in Sect. 5. Finally, the conclusions are presented in Sect. 6.

2 Related Work

Conversational search aims at providing users with an interactive natural language interface for access to unstructured open-domain information [5]. Similarly, task-oriented information navigation dialogue systems often involve search, e.g. for venues or catalogue items [1, 6]. While open-domain conversational search has a wider scope than closed-domain search in task-oriented dialogue, empirical analysis shows that both tasks result in a similar conversational structure [7].

Open-domain search interfaces and task-oriented information navigation dialogue systems both take unconstrained natural language as input. However, dialogue system users are typically limited in expressing their preferences by the domain schema. In response to an out-of-schema user request, a task-oriented dialogue system may produce an informative help message guiding the user to adapt to its limitations [8, 9]. Alternatively, system capabilities may be extended beyond a domain API. For example, Kim et al. [10] propose a method for handling user's follow-up questions in task-oriented dialogue systems. To support pragmatic interpretation, Louis et al. [11] explore users' indirect responses to questions. To extend a task-oriented system to handle natural preferences, a corpus of natural requests for movie preferences was collected using a novel approach to preference elicitation [12].

Task-oriented dialogue systems require accurate models to extract information from unstructured text. Pretrained transformer models, such as BERT [13], have shown to be effective in extracting information from text, leading to significant improvements on many NLP tasks, including open-domain question answering, FAQ retrieval and dialogue generation [10, 14, 15]. Following previous work, we use BERT both in a supervised and an unsupervised setting [16, 17]. We also explore transfer learning from a general natural language entailment task using publicly available corpora [18].

3 Data

To develop and evaluate an entity retrieval component for a dialogue system that handles unconstrained user queries, it is necessary to construct a dataset that includes text snippets associated with the entities, collect unconstrained user queries and identify a set of matching entities for the queries. We collect a new extended dataset of Cambridge restaurants (Sect. 3.1), a set of unconstrained search queries (Sect. 3.2) and annotate query-snippet pairs with relevance labels (Sect. 3.3).

3.1 Cambridge Restaurants Dataset

In previous work [4], the authors used a now outdated dataset of 102 Cambridge restaurants without review information. In this work, we created an up-to-date database with 422 restaurants in Cambridge, UK.[2] Following the schema used in previous work, each restaurant is associated with *cuisine, price range, location* and *description*. As in the past systems, the price range is mapped to *cheap, moderate, expensive* and location to *east, west, centre, south*. However, *cuisine* in our database is associated with a list of values rather than a single value for each entity. In addition, the new dataset includes information on *meals* (breakfast, lunch, dinner), *special diets* (e.g. vegan, gluten free) and *reviews*.

A standard restaurant database may not always contain information relating to the unconstrained user's search preferences (e.g. *cosy family friendly pub*). We thus theorize that personalized descriptions, like *reviews*, are an acceptable source to handle such requests. We collect 62.3K reviews with an average and standard deviation

[2] The dataset is compiled by crawling the Web in January 2021.

of 145(256) per restaurant. The reviews together with text in each data fields are used as *text snippets* to retrieve items relevant to the query in the experiments. Only positive reviews (rating 4 or 5 stars) are used, as we expect user queries to mention desirable properties of the restaurant.

3.2 Unconstrained Queries

To simulate natural unbiased user requests in a restaurant search domain, we created an online form with one question: *'Please type a sentence describing your restaurant preference to your smart virtual assistant'*. The form was distributed to several university and company mailing lists. Each participant was asked to enter one query.[3]

We found that only 30 out of 105 collected queries specified a constraint corresponding to one of the predefined fields of the database schema (area, cuisine or price range) that may be used to search for a restaurant in a purely schema-driven system. Although only 14 of these queries contained an entity exactly matching a value in the database and only 7 had no other preferences besides the slot value. For example, *'I would like to eat in a fine dining establishment, preferably* ***french*** *cuisine'* contains a mention of cuisine as well as a vague preference for a *'fine dining establishment'*. The unbiased queries are highly diverse and most frequently contain a menu item (46), a subjective (31) or an objective preference (25) about a restaurant (see Table 1). We removed 5 queries that mention proximity, e.g. *'near me'*. Such queries would not be handled with the proposed method as it would require additional geographical location information. We use the remaining 100 queries in our experiments.

3.3 Query-Snippets Annotation

For extracting relevant entities using unstructured data, we need to develop supervised models, which require preferably a balanced training dataset. We create a Restaurant Query Snippet (ResQS) dataset of query-snippet pairs, where the snippets include text from reviews and each of the database fields, labelled with '1' if the snippet is *relevant* to the query and '0', otherwise. For each query, most of the snippets from our set of 62.3K candidates will be irrelevant. Thus, a random selection of snippets for each query would result in an unbalanced dataset of mostly irrelevant snippets. Instead, unsupervised and transfer learning methods are used to select a set of snippets for each query. We then use manual annotation on the selected set to create query-snippet pairs with annotated relevance information.

[3] As the task was very short, the participants were not paid.

Table 1 Preferences (number of queries) mentioned by the users in 105 unbiased queries

Preference	Examples
Menu item (46)	Burger, veggie burger, pizza, curry, steak, fried rice, meat, sushi, seafood, noodle bar, dim-sum, shark, eel, alcohol, wine, mulled wine, beer, whisky, milkshake, dessert
Objective (25)	Live music, music, quiet, dogs, kids, fireplace, free delivery, local food, new place, biggest pizza, spicy food, large groups, Burn's night, big pizza, sweet tooth, traditional, portion size, restaurant name, parking, quick, outdoor seating, spaced out tables, byob
Subjective (31)	Ambience, friendly, gourmet, authentic, romantic, cozy, small, local, safe, different, inventive, stylish, exotic, interesting, hygienic, fine dining, best value, fancy, nothing too fancy

For each of the 100 queries, we first score all snippets using two unsupervised and one transfer learning method (see the first three methods shown in Fig. 2). Next, we compute a relevance score for each of the 422 restaurants in the dataset and use it to rank them (see Sect. 4). Finally, we randomly sample one of the five top-ranked restaurants for the query[4] and manually label the top five snippets that were used to compute the relevance score of this restaurant. Manual labels are used for the evaluation of the unsupervised methods as well as for training the supervised models. Using three methods to select five snippets per query we generate 1500 query-snippet pairs. Additionally, to simulate the hybrid system where both relevance ranking and database match are used to extract a relevant item, we use the 14 queries that specify a value for one of the system slots (*area*, *cuisine* or *price range*). This subset is then used to select a recommended restaurant for the query. Thus by using three methods to select five snippets for the 14 queries, we extend the dataset by 210 query-snippet pairs resulting in 1710 examples.

For annotation, we use crowd workers from Amazon Mechanical Turk. Each Human Intelligence Tasks (HIT) consists of 23 randomized query-snippet pairs and is annotated by three crowd workers.[5] The workers were compensated according to government pay guidance.[6] We use the majority vote among the three crowd workers to select the final label. The pairwise agreement between each annotator and the majority vote is very good with Fleiss Kappa $k > 0.7$ [19].

---

[4] The number of top results (5) was chosen empirically since a user of a dialogue system may navigate over multiple search results.

[5] Three query-snippet pairs with a known label are used to monitor work quality. The workers who did not pass the quality test were rejected.

[6] https://www.gov.uk/national-minimum-wage-rates.

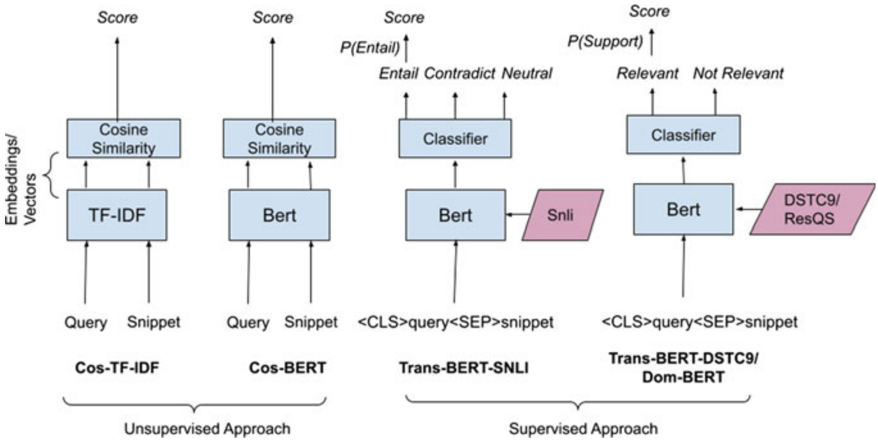


Fig. 2 Relevance scoring methods

4 Method

In this work, we extend a schema-driven dialogue system to support user's unconstrained search requests. Our goal is to extract the items most relevant to the user's query. A dialogue system then presents these items as recommendations to the user and responds to follow-up questions. This work only addresses the extraction of the relevant items, leaving the evaluation of information presentation and handling of interaction to future work. We apply and evaluate our approach using the Cambridge Restaurants search domain.

Algorithm 1 Relevance Ranking

procedure RANKANDSELECT($Query, Dataset, J, N$)
  Input: $Query$ is a String of user's request
  Input: $Dataset$ is a collection of items with associated snippets $Item : [S1, S2, ...]$
  Input: J is the number of top snippets used to compute the score
  Input: N is the number of items to return
  Output: List of top N items for the $Query$
  for each $Item \in Dataset$ **do**
    for each $Snippet \in Item$ **do**
      $Snippets[score] = RelevanceScore(Query, Snippet)$
    end for
    $Sorted\_Snippets = Sort(Snippets.score, order = descending)$
    $Item[score] = \frac{1}{J}\sum_{i=1}^{J} Sorted\_Snippets\_i[score]$
  end for
  $Sorted\_Items = Sort(Items.score, order = descending)$
  return $[Sorted\_Items_1..Sorted\_Items_N]$
end procedure

Algorithm 1 is used to extract an item most relevant to the user's query. First, the snippets are scored (line 9), then the top recommended restaurants are obtained by ranking the list of restaurants based on the average score of the top-5 snippets.[7]

For snippets scoring, we explore two unsupervised approaches: Cos-TF-IDF and Cos-BERT, and two supervised approaches: transfer learning from another domain (Trans-BERT), and in-domain training approach (Dom-BERT) illustrated in Fig. 2. The input to each of the models are two strings (query and text snippet) and the output is a score where the scores closer to 1 indicate a higher relevance.

4.1 Unsupervised Approach

We expect relevant text snippets to have a higher word overlap and semantic similarity with the corresponding query. For example, for a user query:

Query: *I am looking for a place that serves vegan food and also allows dogs inside.*

the two relevant snippets have overlapping vocabulary:

1. **Special diets**: vegan friendly
2. **Review**: It was such a happy surprise that they allowed dogs inside their premises. Fanta was woofing with delight.

We map the user query (Q_i) and each snippet (S_j) for all of the restaurants into a fixed-sized vector using a mapping function M. We then compute the cosine similarity score between the user request and each snippet:

$$Score(Q_i, S_j) = cos(M(Q_i), M(S_j))$$

The cosine score estimates the distance between the query Q_i and snippet S_j.

Cos-TF-IDF: Cosine Similarity using TF-IDF Encoding
As a baseline, we use Term Frequency-Inverse Document Frequency (TF-IDF), an efficient and simple algorithm for matching words in a query to documents that are relevant to that query, to encode the query and the knowledge snippets [20]. We compute TF-IDF for each snippet and for each query. To encode the query and the snippet, the words are replaced by their TF-IDF score to form vocabulary-sized vectors. Then, the words in the query and the words in the text snippet are replaced by their TF-IDF score to form vocabulary-sized vectors.[8] The cosine score corresponds to the vocabulary overlap between Q_i and S_j.

Cos-BERT: Cosine Similarity using BERT Encoding
We use pretrained Sentence-BERT (SBERT) model to encode the query and the snippets [21]. SBERT uses a pooling operation to obtain the mean of all output

[7] We use empirically chosen J = 5 and N = 5 in this work.

[8] The vocabulary size in our domain is 30124.

vectors of BERT, fine-tuned on the SNLI corpus, to derive a fixed-sized sentence embedding [13]. The cosine score corresponds to the semantic distance between Q_i and S_j.

4.2 Supervised Approach

We build a classifier using a DNN with a single fully connected linear layer that takes as input encoded representation of a query and a snippet. The input ($< CLS >$ $query < SEP > snippet$) is encoded with the pretrained BERT model and the encoding of the special symbol (CLS) is passed into a linear layer that outputs a two- or three-way classification. The model is trained to minimize cross-entropy on the training set.

Trans-BERT: Transfer Learning Approach
For transfer learning we use Stanford Natural Language Inference (SNLI) and DSTC9 datasets [6, 10]. SNLI corpus is a collection of 570,000 sets of premises and hypotheses sentences annotated with the labels *contradiction*, *entailment* and *neutral* as in the following example:

Premise: A boy is jumping on skateboard in the middle of a red bridge.
Entailment: The boy does a skateboarding trick.
Contradiction: The boy skates down the sidewalk.
Neutral: The boy is wearing safety equipment.

The intuition behind using a model trained on the SNLI dataset is that the relevant snippet for the user's query would be classified as *entailment* while the irrelevant snippets would be classified as *neutral* or *contradiction*. Hence, we train a three-way classification and use the score of *entailment* class to estimate relevance.

DSTC9 dataset was constructed for the purpose of extracting answers to follow-up questions in a dialogue system. It contains 938 question and answer pairs for the train, hotel and restaurant domains as shown in following examples:

Request: Are children welcomed at this location?
Reply: Yes, you can stay with children at A and B Guest House.

Request: Can my small dog stay with me?
Reply: Pets are not allowed at the A and B Guest House.

We use the question-answer pairs as the positive examples (relevant) for training the model. The negative examples for training a binary classifier are extracted by randomly sampling answers from different questions following the approaches used by the authors.

Dom-BERT: In-domain Training Approach
The models that are trained on in-domain data usually achieve better performance. To compare the models trained with the in-domain data with the transfer/unsupervised approaches, we train a supervised binary classifier on ResQS and the combination of

ResQS and DSTC9 datasets. A pretrained BERT encoder was fine-tuned on the in-domain classification task using the pruning method that removes the task-irrelevant neural connections from the BERT model to better reflect the data for the task [22].

5 Experiments and Results

Evaluating the overall relevance of a recommended restaurant is challenging, as the information presented to the user would bias their judgement. We assume that the restaurants preferred by the user are associated with the snippets that are relevant to the query. The models are evaluated based on the scores they assign to the snippets.

We use the two unsupervised and the transfer learning from SNLI models for *information retrieval* to extract a recommended restaurant for each query by ranking restaurants based on the relevance score of the snippets (see Sect. 5.1). The dataset produced with the information retrieval is then manually annotated and used to train supervised *snippet relevance classification* models. To compare snippet relevance classification methods, we use a threshold to classify each snippet's relevance to the query and compute the classifier's precision, recall and F1 (see Sect. 5.2).

5.1 Information Retrieval

For each of the queries, we apply unsupervised and transfer learning methods to score each snippet and use these scores to rank the restaurants. As a dialogue system can present multiple options in response to a query, we randomly select one of the top five recommended restaurants and manually annotate the top five matching snippets with a binary label *relevant/not relevant* (see Sect. 3.3). For the two unsupervised cosine similarity (Cos-TF-IDF and Cos-BERT) and transfer (Trans-BERT-SNLI) methods we report the *snippet relevance* (the percent of snippets extracted by the model labelled as *relevant*) and the *overall recommendation* quality (see Table 2).

Table 2 Snippet relevance and the overall recommendation result of the information retrieval

Model	% Relevant Snippets				Overall recommendation (per query)	
	All (500)	Menu Item (200)	Objective (125)	Subjective (155)	% with #relevant snips ≥ 1	avg #relevant snips
Cos-TF-IDF	65%	59%	53%	67%	83%	3.25
Cos-BERT	57%	61%	55%	60%	86%	2.86
Trans-BERT (SNLI)	16%	10%	13%	20%	41%	0.79

We further analyze the relevance scoring performance across three query types that mention a menu item, objective or subjective information.

Cosine similarity approach using TF-IDF encoding (Cos-TF-IDF), which relies on exact keyword match, achieves the highest overall snippet relevance of 65%, followed by Cos-BERT with 57%. Transfer learning did not work well, yielding only 16% relevant snippets. We observe that Cos-TF-IDF performance is the highest (67%) on the queries with *subjective* information that contain adjectives (‘excellent’, ‘great’). However, its performance is below Cos-BERT model on the queries with *objective* and *menu item* information (‘fireplace’, ‘dogs’, ‘desserts’). For a user query *‘Find me a restaurant with great desserts’*, Cos-TF-IDF model extracts a restaurant with generic positive reviews which are not relevant to the query:

1. **Review**: Great service, great food, had a great night! Good value for money and great atmosphere, definitely coming back.
2. **Review**: Great find in Cambridge.

However, Cos-BERT model extracts a restaurant with the snippets relevant to this query that focus on the quality of deserts:

1. **Review**: The food in this restaurant is very good. However, it is the desserts that steal the show. I have sometimes been there just for dessert.
2. **Review**: Fabulous french style food and cocktails. We plan to return for dessert only as the selection looked amazing and we were quite full from our meal.

Using BERT for embedding the query and the snippets appears to capture the semantics which is especially important for the queries with objective information.

The overall recommendation accuracy is the proportion of recommended restaurants that would satisfy the user. We approximate the subjective user satisfaction with (1) the percentage of the queries for which at least one of the top five snippets was labelled as *relevant* and (2) the average number of the top five snippets labelled as *relevant*. Cos-BERT model outperforms the Cos-TF-IDF on the first metric (86% vs. 83%) and Cos-TF-IDF outperforms Cos-BERT on the second one (3.25 vs. 2.86).

Our dataset contains 62.3K snippets. Exact word match (TF-IDF) may have outperformed the semantic method (Cos-BERT) because it was likely to find exact word match for each query. However, for smaller datasets, where vocabulary in a query may not match any of the text snippet, semantic methods may be more beneficial.

The overall relevance scoring accuracy is 48%, resulting in a balanced dataset of 1710 query-snippet (ResQS) pairs which we use to train and evaluate supervised relevance labelling models described in the next section.

5.2 Snippet Relevance Classification

We compare the snippet relevance classification performance of the unsupervised (Cos-TF-IDF, Cos-BERT), transfer (Trans-BERT) and supervised (Dom-BERT) methods. The unsupervised cosine similarity methods use a threshold of 0.5 to determine relevance of a snippet to a query. The Trans-BERT models are trained on the

Table 3 Relevance classification performance on ResQS dataset

Model type	Training data	Avg precision	Avg recall	Weighted F1
Always-relevant baseline	–	0.240	0.490	0.322
Unsupervised methods				
Cos-TF-IDF	–	0.752	0.519	0.365
Cos-BERT	–	0.703	0.672	0.661
Transfer learning methods				
Trans-BERT	SNLI	0.429	0.493	0.368
Trans-BERT	DSTC9	0.771	0.768	0.769
Supervised in-domain training methods				
Dom-BERT	ResQS	0.829	0.825	0.824
Dom-BERT	ResQS + DSTC9	0.859	0.857	0.856

publicly available SNLI and DSTC9 datasets. The model trained on SNLI dataset outputs a three-way classification with the classes *contradiction*, *entailment* and *neutral*. To apply it on our binary query relevance detection task, we combine *contradiction* and *neutral* into the *not relevant* class and use *entailment* as the *relevant* class. The unsupervised and transfer methods are evaluated on the full ResQS corpus as they do not use any of its data for training and the supervised methods are evaluated with tenfold cross-validation.

Table 3 shows the average precision, recall and weighed F1-scores on the ResQS dataset. Cos-TF-IDF method achieves F1 of 0.365, slightly higher than the baseline that predicts all snippets as *relevant* (F1 = 0.322) while Cos-BERT achieves F1 of 0.661. We observe that Cos-TF-IDF which relies on word match has a lower recall than Cos-BERT which captures semantic meaning (0.519 vs. 0.672).

The Transfer-BERT trained on SNLI achieves F1 of 0.368, slightly higher than the *always-relevant* baseline but lower than the unsupervised Cos-BERT model. The Trans-BERT model trained on DSTC9, a dataset more closely resembling to the target data, achieves F1 of 0.769 outperforming both of the unsupervised methods. The model trained on in-domain ResQS dataset achieves F1 of 0.824 and outperforms all unsupervised and transfer learning models. The best result (F1 = 0.856) is obtained by training on the combined in-domain ResQS and DSTC9 dataset.

5.3 Discussion

Our aim was to collect data that covers most of the aspects of the individual restaurants. In addition to the standard aspects (cuisine, location) provided by the restaurants, we also used reviews which augmented the restaurant information with further non-standard (fireplace, dogs, exotic) aspects thus providing a comprehensive unstructured dataset for handling unconstrained queries.

We then test how accurately an unsupervised method can extract relevant items using unstructured data. Cosine similarity with the BERT encoder method resulted in 57% relevant snippets on the information retrieval task and achieved F1-score of 0.661 on the snippet relevance classification task. With this approach 86% of top-5 recommended restaurants had at least one relevant snippet (the estimated recommendation accuracy). Assuming that an item with more relevant snippets is a better match for a user query, we expect to achieve even higher restaurant recommendation accuracy using a more accurate supervised snippet relevance scoring model. The supervised model achieves F1-score of 0.856 on the snippet relevance classification task.

Although we achieve improvements with the supervised models, it should be noted that they are slower compared to unsupervised methods.[9] If using a supervised model to extract relevant information from unstructured data is not real time, it becomes unfeasible for use in a dialogue system. To reduce latency while maintaining accuracy, we could use a combination of models by first filtering snippets with an unsupervised method and then applying a supervised model on a smaller dataset.

6 Conclusions

In this work, we aim to improve naturalness of interaction with an information navigation dialogue system. When a user is not primed by instructions, most user queries in the restaurant search domain include out-of-schema search preferences. Such queries cannot be handled by a purely schema-driven dialogue system, resulting in ineffective and unnatural conversations. To address this problem, we propose an entity retrieval method by incorporating a snippet relevance classifier into the pipeline of a schema-driven dialogue system.

We present an effective methodology for extending schema-driven dialogue systems to use unstructured knowledge and handle out-of-schema user preferences.[10] We update the Cambridge restaurants database, and extend it with text knowledge snippets. To simulate a naive user, we collect restaurant queries from users without biasing them with specific instructions. In our experimental restaurant search domain, the snippets were obtained from publicly available restaurant reviews and descriptions and annotated by Amazon Mechanical Turk workers. We build a supervised text relevance classification model on the annotated data and compare its performance with an unsupervised method. We show that on the annotated dataset an unsupervised/supervised classifier achieves a weighted F1 of 0.661/0.856. In future work, we will incorporate the proposed approach into the Cambridge restaurant search dialogue system and evaluate it with users.

[9] Experiments were conducted with i7-6 cores CPU and single GTX 1080 GPU.

[10] The project was completed during a 4-month internship and the annotation costs were under $300.

References

1. Budzianowski P, Wen T-H, Tseng B-H, Casanueva I, Ultes S, Ramadan O, Gavsic M (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, Brussels, Belgium, pp 5016–5026
2. Rastogi A, Zang X, Sunkara S, Gupta R, Khaitan P (2020) Schema-guided dialogue state tracking task at DSTC8. In: Proceedings of the AAAI dialog system technology challenges workshop
3. Stoyanchev S, Keizer S, Doddipatla R (2021) Action state update approach to dialogue management. In: ICASSP 2021–2021 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 7398–7402
4. Henderson M, Thomson B, Williams JD (2014) The second dialog state tracking challenge. In: Proceedings of the 15th annual SIGdial meeting on discourse and dialogue, pp 263–272
5. Trippas J.R, Spina D, Cavedon L, Joho H, Sanderson M (2018) Informing the design of spoken conversational search: Perspective paper. In: Proceedings of the 2018 conference on human information interaction & retrieval. CHIIR '18. ACM, New York, NY, USA, pp 32–41
6. Yan Z, Duan N, Chen P, Zhou M, Zhou J, Li Z (2017) Building task-oriented dialogue systems for online shopping. In: Proceedings of the thirty-first AAAI conference on artificial intelligence. AAAI'17. AAAI Press, San Francisco, CA, USA , pp 4618–4625
7. Vakulenko S, Revoredo K, Di Ciccio C, de Rijke M (2019) Qrfa: a data-driven model of information-seeking dialogues. In: Azzopardi L, Stein B, Fuhr N, Mayr P, Hauff C, Hiemstra D (eds) Advances in information retrieval. Springer, Cham, pp 541–557
8. Komatani K, Ueno S, Kawahara T, Okuno HG (2005) User modeling in spoken dialogue systems to generate flexible guidance. User Model User Adapt Interact 15(1–2):169–183
9. Tomko S, Rosenfeld R (2004) Speech graffiti versus natural language: Assessing the user experience. In: Proceedings of HLT-NAACL 2004: short papers. Association for Computational Linguistics, Boston, MA, USA, pp 73–76
10. Kim S, Eric M, Gopalakrishnan K, Hedayatnia B, Liu Y, Hakkani-Tur D (2020) Beyond domain APIs: Task-oriented conversational modeling with unstructured knowledge access. In: Proceedings of the 21th annual meeting of the special interest group on discourse and dialogue. Association for Computational Linguistics, 1st virtual meeting, pp 278–289
11. Louis A, Roth D, Radlinski F (2020) "I'd rather just go to bed": Understanding indirect answers. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Online, pp 7411–7425
12. Radlinski F, Balog K, Byrne B, Krishnamoorthi K (2019) Coached conversational preference elicitation: a case study in understanding movie preferences. In: Proceedings of the annual SIGdial meeting on discourse and dialogue
13. Devlin J, Chang M.-W, Lee K, Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT
14. Wang Z, Ng P, Ma X, Nallapati R, Xiang B (2019) Multi-passage bert: a globally normalized bert model for open-domain question answering. 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), pp 5881–5885
15. Sakata W, Shibata T, Tanaka R, Kurohashi S (2019) Faq retrieval using query-question similarity and bert-based query-answer relevance. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp 1113–1116
16. Izacard G, Grave E (2021) Leveraging passage retrieval with generative models for open domain question answering. In: Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume, pp 874–880
17. Zhan J, Mao J, Liu Y, Zhang M, Ma S (2020) An analysis of BERT in document ranking. Association for Computing Machinery, New York, NY, USA, pp 1941–1944
18. Bowman SR, Angeli G, Potts C, Manning CD (2015) A large annotated corpus for learning natural language inference. In: EMNLP, pp 632–642

19. Fleiss JL, Cohen J (1973) The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. Educ Psychol Meas 33(3):613–619
20. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manag 24(5):513–523
21. Reimers N, Gurevych I (2019) Sentence-bert: sentence embeddings using siamese bert-networks. 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), pp 3982–3992
22. Gao Y, Colombo N, Wang W (2021) Adapting by pruning: a case study on BERT. arXiv:2105.03343

Jurassic is (Almost) All You Need: Few-Shot Meaning-to-Text Generation for Open-Domain Dialogue

Lena Reed, Cecilia Li, Angela Ramirez, Liren Wu, and Marilyn Walker

Abstract One challenge with open-domain dialogue systems is the need to produce truthful, high-quality responses on any topic. We aim to improve the quality and coverage of Athena, an Alexa Prize dialogue system. We experiment with few-shot prompt-based learning, comparing GPT-Neo to Jurassic-1 for the movies, music, TV, sports, and video game domains, both within and cross-domain, with different prompt set sizes (2, 3, 10), formats, and meaning representations consisting of either set of WikiData KG triples, or dialogue acts. Our evaluation uses BLEURT and human metrics and shows that with 10-shot prompting, Athena-Jurassic's performance is significantly better for coherence and semantic accuracy. Experiments with a 2-shot cross-domain prompt result in a considerable performance drop for Athena-GPT-Neo, whose semantic accuracy falls to 0.41, and whose untrue hallucination rate increases to 12%. Experiments with dialogue acts for video games show that with 10-shot prompting, both models learn to control dialogue acts, but Athena-Jurassic has significantly higher coherence and only 4% untrue hallucinations. Our results suggest that Athena-Jurassic produces high enough quality outputs to be useful in live systems with real users. To our knowledge, these are the first results demonstrating that few-shot semantic prompt-based learning can create NLGs that generalize to new domains and produce high-quality, semantically controlled, conversational responses directly from meaning representations.

1 Introduction

One challenge with open-domain dialogue systems is the need to respond to users' utterances on any topic with high-quality responses. To handle this challenge, a common approach is to use an ensemble of response generators (RGs) and then

L. Reed · C. Li (✉) · A. Ramirez · L. Wu · M. Walker
University of California, Santa Cruz 95062, CA, USA
e-mail: yli331@ucsc.edu

M. Walker
e-mail: mawalker@ucsc.edu

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_7

train a ranker to select from a pool of possible responses [6, 8, 10, 11, 16, 32, 40]. The ensemble of RGs can use a variety of generation techniques. One type of RG generates responses directly from the dialogue context, using a pre-trained model such as GPT-2 that is possibly tuned on additional conversational data, such as Topical Chat or Empathetic Dialogues [14, 32]. Knowledge-Grounded response generation is a variant of this approach, where knowledge in the form of text is available during generation to control the utterance content and veracity [18, 46]. Template-Based RGs are also quite common, where templates are hand-written and then populated with values from a database or knowledge graph (KG).

We are primarily interested in Meaning-to-Text (M2T) NLG engines, where the desired meaning is specified and the NLG engine produces one or more utterances that express that meaning. These can be trained with parallel corpora of meaning representations (MRs) and reference utterances [4, 7, 17, 27, 41, 44] or can be trained by tuning a large language model with a domain-specific training set [21, 29, 37, 49, 50]. These M2T engines are most often used for task-oriented responses on specific (sets of) topics, such as restaurants, sports, movies, hotels, laptops, TVs, and video games [5, 9, 13, 22].

Recent work on language models as few-shot learners [2] suggests a novel paradigm for creating a M2T NLG engine with prompt-based learning, where prompts are provided that condition the language model to produce a certain type of output. In our case, these conditioning prompts consist of pairs of meaning representations and their corresponding output. Our aim in this paper is to investigate how prompt-based learning can improve the quality and the coverage of the M2T RGs in Athena, a dialogue system that has been a finalist in the Alexa Prize for the last two years [16, 28].

There are two types of M2T RGs in Athena. One set is template-based whose MRs are sets of triples from the WikiData Knowledge Graph (KG-RGs). These KG-RGs cover movies, sports, music, and TV topics. One advantage of using WikiData is the automatic updates to its knowledge by its editors. Another benefit is the ability to traverse the KG to new relations or entities, to say more things about a topic in a dialogue [26]. The third benefit is that Athena represents named entities and pronouns with their Wikidata IDs, providing a direct link to the KG [28]. Figure 1 provides a conversation with the Music KG-RG that shows how the knowledge triples from WikiData are realized with templates.[1] The other RG, for the video games topic, is based on the Viggo corpus [20]. This is a parallel corpus of MRs and crowd-sourced realizations. A conversation with the Viggo RG is below in Fig. 3.

Both of these RGs require substantial human effort. The KG-RGs require two steps: (1) sets of *interesting* and *frequently populated* KG relations must be selected by hand [26, 28] and (2) templates must be hand-written to realize them. This means that they currently cover a limited set of relations, ones that are populated frequently enough to make writing templates worthwhile. As previous work on dialogue generation has shown, even combinations of existing relations typically require multiple

[1] In accordance with the Alexa Prize rules, the shared conversations in Figs. 1 and 3 are between Athena and our team, or UCSC undergraduates, rather than real users.

A1:	Do you have a favorite musician?
U1:	yeah, taylor swift.
A2:	Ok, Taylor Swift. **Wow! Taylor Swift is very prolific! She has 114 songs, that's a lot!** (Taylor Swift, numTracks, 114)
U2:	yeah, that is a lot
A3:	**I like Taylor Swift's song, 22. It came out in 2013.** Do you like that song? (Taylor Swift, song, 22), (22, date, 2013)
U3:	yeah, it's a good one.
A4:	Right? This is interesting, **Taylor Swift sings the song Bad Blood with Kendrick Lamar**, want to hear more about Kendrick Lamar? (Bad Blood, performer, Taylor Swift), (Bad Blood, performer, Kendrick Lamar)
U4:	not really
A5:	One of my favorite musicians is P!nk. Isn't this interesting, **P!nk won a Grammy Award for Best Pop Collaboration with Vocals**. Do you want to hear more? (P!nk, award, Grammy Award for Best Pop Collaboration with Vocals)

Fig. 1 A conversation about music between Athena (A) and a user (U). The music KG-RG interweaves template-based utterances based on KG triples with other template-based utterances. Response segments generated using the KG triples are highlighted in orange. For each segment, the KG triples used by the RG as the input MR are shown below the response

additional templates to be written [34, 35, 43]. The existing KG-RG entities and relations are in Table 1, as well as novel KG-RG relations and entities that we experiment with below with 2-shot prompting.

The neural Viggo RG required constructing dialogue acts and meaning representations, populating their content with information from IGDB, crowd-sourcing responses, and cleaning the resulting corpus, in order to experiment with methods for improving the fluency and semantic accuracy responses [20, 21]. Thus, the ability to reliably generate high-quality responses directly from MRs via neural NLGs would transform the use of M2T NLGs in dialogue systems [7, 9, 23, 36, 45, 49].

We utilize Athena's current RGs to create prompt and test sets for two new neural Meaning-to-Text RGs, Athena-GPT-Neo and Athena-Jurassic [1, 2, 25]. We conduct few-shot prompt-based learning experiments, where we systematically vary within and cross-domain prompts, different prompt set sizes (2, 3, 10), prompt formats, and type of meaning representations. We expect that these NLGs to generalize beyond their conditioning data [3, 15, 31, 33]. We evaluate the results using both BLEURT and human evaluation. Our results show that, with 10-shot conditioning, both Athena-GPT-Neo and Athena-Jurassic generally produce coherent outputs, but

Table 1 The KG topics, entities, and relations in the Athena-KG-synthetic corpus. A * indicates novel entities and relations that are tested in Sect. 4

Topic	Entities	Relations
Movies	Movies Actors Directors Awards*	cast voiceCast spouse childrenNum genre award director* work* date* screenWriter* producer*
Music	Musicians Bands Awards* Songs* Albums*	performer (song and album) numTracks genre award memberOf instrument label date* show* work*
Sports	Athletes Sports Awards*	team position participant (tournament, leagues) spouse childrenNum award height date* work* ranking* duration* reviewScoreBy* disciplineCompetedIn* numMatches* numAwards* draftedBy* draftPicknum* startTime*
TV	Shows Assessment* characterRole*	cast role creator director genre award characterRole* narrativeLocation* mainSubject* assessment* assessmentOutcome* hasPart* occupation* derivativeWork* startTime* endTime* filmingLocation* setInPeriod* numSeasons* numEpisodes*

that for within-domain experiments, Athena-Jurassic's performance is significantly better for the human evaluation metrics of coherence and semantic accuracy. Experiments with 2-shot prompts on completely novel MRs result in a huge performance drop for Athena-GPT-Neo, whose semantic accuracy falls to 0.41, and untrue hallucination rate increases to 12%. Experiments with the Viggo video games corpus show that, with 10-shot prompts, both Athena-GPT-Neo and Athena-Jurassic can learn to control the dialogue acts realized, but Athena-Jurassic has significantly higher coherence, mainly because Athena-GPT-Neo produces some redundant and repetitive utterances. Athena-GPT-Neo also produces untrue hallucinations in 12% of the video game outputs. We use the human evaluation to examine whether the BLEURT scores are meaningful with results showing that BLEURT scores have a very good correlation with semantic accuracy, but not with coherence, or other human metrics. Our results suggest that Athena-Jurassic can reliably produce outputs that are of high enough quality to be used in live systems with real users. To our knowledge, this is the first paper to show that few-shot prompt-based learning can create NLGs that generalize to new domains and produce high-quality, semantically controlled, conversational responses directly from MRs and KG triples.

2 Datasets

We use two datasets for prompt-based learning for Athena's response generators. One is based on Athena's KG-RGs for the domains of music, movies, sports, and TV. Figure 2 provides KG triples (entity, relation, entity) and responses produced by the

Topic	Example
Movies	(Wonder Woman, director, Patty Jenkins) I believe I read that Wonder Woman is directed by Patty Jenkins. Sometimes a director's unique voice really comes through in their work. Do you think that happened in this case?
Music	(Rihanna, record label, Def Jam Records) Here's another musician who worked for the same label Def Jam Records, called Rihanna. Want to hear about them?
Sports	(Lebron James, member of sports team, Los Angeles Lakers), (Lebron James, position played on team/specialty, power forward) Lebron James has played on many famous teams such as the Los Angeles Lakers, and played many positions like power forward.
TV	(Lost, genre, paranormal television program), (Lost, genre, drama television) Lost is considered both a paranormal television program and a drama television. What's your opinion of paranormal television programs or drama television shows?

Fig. 2 Sets of KG triples used as meaning representations, and the corresponding utterances. Dialogue Act types are not represented in the MRs but note that some KG triples are realized with multiple dialogue acts, combining statements, opinions, and questions

current KG-RGs. Each response is designed to continue the conversation with one or more statements, opinions, and/or question dialogue acts about an entity. However, the MRs represent the triples and do not explicitly represent dialogue acts. Some responses are based on traversing the KG links to a related entity and introducing that entity, as illustrated in the music example about *Rihanna*, where the record label *Def Jam Records* is used to link two music artists. A music conversation was given in Fig. 1 above. Both existing and novel KG-RG entities and relations are in Table 1.

To use prompt-based learning to create new KG-RGs, we create a new corpus, Athena-KG-Synthetic, of sets of knowledge triples and their template-based responses. We select five template categories and their paraphrases from the movies RG, two from music, three from sports, and two from TV.[2] We query WikiData for thousands of KG triples to populate the templates and then split the resulting dataset into ∼32K train, 3558 development, and a test set of 100 instances for each template category.

The second dataset is the Viggo dataset. The Viggo RG combines responses generated from templates with those generated from meaning representations, as shown in Fig. 3. The set of dialogue acts (DAs) are carefully constructed to be conversational and engage the user, rather than being purely informative [20, 21]. We directly use the ViGGO corpus's training, development, and test sets. Each corpus instance uses one of Viggo's 9 dialogue acts such as *verify attribute*, *request explanation*, or *recommend*. Most DAs are compatible with many combinations of content slots, using 14 video game attributes, yielding hundreds of response types [21]. Figure 4 shows four DAs with various slot combinations.

[2] Viggo and the Athena-KG-Synthetic corpus are available from nlds.soe.ucsc.edu.

A1:	. . . Well, one of my all-time favorites is Control. I have already completed it twice! Have you played it by any chance?
U1:	no not yet
A2:	Obviously. There are just too many fun games for one person to play in a single lifetime. At any rate, **Control** is an **excellent third-person action-adventure shooter** with **multiplayer**. It was released in **2019** by **Remedy Entertainment** for **PC**, **PlayStation**, and **Xbox**. Do you think you would potentially like it? *inform*(name[**Control**], release_year[**2019**], developer[**Remedy Entertainment**], rating[**excellent**], genres[**shooter**, **action-adventure**], has_multiplayer[**yes**], player_perspective[**third person**], platforms[**PC**, **PlayStation**, **Xbox**])
U2:	absolutely i was actually hoping to play it sometime soon
A3:	Alright! Have you played any **adventure** games by **Remedy Entertainment**, like **Alan Wake**? *suggest*(name[**Alan Wake**], developer[**Remedy Entertainment**], genres[**adventure**])
U3:	yes i did i loved the dark atmosphere in alan wake
A4:	Yeah! I remember playing Alan Wake some time ago, and it wasn't bad at all. Do you like **multiplayer** games? *request_attribute*(has_multiplayer[])

Fig. 3 A conversation between Athena(A) and a user (U) about video games, exhibiting a smooth interweaving of neural and template-based responses. Response segments generated by the neural model are highlighted in orange. For each such segment, the MR constructed by the RG as input for the model is shown below the response

confirm(NAME [**Hellblade: Senua's Sacrifice**], RELEASE_YEAR [**2017**], DEVELOPER [**Ninja Theory**])
Oh, do you mean the **2017** game from **Ninja Theory**, **Hellblade: Senua's Sacrifice**?
suggest(NAME [**Half-Life 2**], GENRES [**shooter**], PLAYER_PERSPECTIVE [**first person**]
Do you also enjoy playing **first-person shooters**, such as **Half-Life 2**?
give_opinion(NAME [**SpellForce 3**], RATING [**poor**], GENRES [**real-time strategy, role-playing**], PLAYER_PERSPECTIVE [**bird view**])
I think that **SpellForce 3** is **one of the worst games** I've ever played. Trying to combine the **real-time strategy** and **role-playing** genres just doesn't work, and the **bird's eye view** makes it near impossible to play.
verify_attribute(NAME [**Little Big Adventure**], RATING [**average**], HAS_MULTIPLAYER [**no**], PLATFORMS [**PlayStation**])
I recall that you were **not that fond** of **Little Big Adventure**. Does **single-player** gaming on the **PlayStation** quickly get boring for you?

Fig. 4 Viggo structured MRs (grey rows) and the corresponding reference utterances (with slot mentions in bold). Dialogue Act types are indicated in italics at the beginning of the MRs

3 Experimental Setup

We utilize the models GPT-Neo and Jurassic-1 jumbo [1, 25]. GPT-Neo is a transformer-based language model that has 1.7 billion parameters. It was created as an open-sourced alternative to GPT-3. Similarly to previous GPT-2 and GPT-3 models, GPT-Neo predicts the next word given the previous words in the text. The team from EleutherAI generated an open-source training set, The Pile [12], comparable to that used for GPT models. The Pile is 825GB with data from 22 diverse sources, such as academic sources(Arxiv, PubMed), Github, and Wikipedia. GPT-Neo has a vocabulary size of ∼50K tokens. The EleutherAI team provides three models (125M, 1.3B, and 2.7B), which were trained as masked auto-regressive models using cross-entropy loss. When compared to the closest GPT-3 model (GPT-3 Ada), GPT-Neo 2.7B had better performance on all linguistic and scientific reasoning benchmarks (HellaSwag, PIQA, WinoGrande, MathQA, and PubMedQA). We use GPT-Neo 1.3B, which has promising performance for its size.[3]

Jurassic-1 is also an auto-regressive transformer-based language model that achieves the state-of-the-art performance on a set of common sense and QA zero-shot and few-shot tasks [25, 38, 48]. AI21 Labs has released two versions, J1-large with 7.5B parameters and J1-jumbo with 178B parameters. Jurassic-1 is pre-trained with 300B tokens taken from publicly available resources and has a larger vocabulary than other similar models with 250K tokens. Jurassic-1 has a larger vocabulary by including n-gram phrases as tokens along with the standard unigram and subword tokens. Jurassic-1's architecture attempts to optimize the Jurassic's depth-width tradeoff [24, 25]. The paper claims that Jurassic-1 can predict text from a broader set of domains than GPT-3 and is superior to GPT-3 in few-shot settings, due to its ability to fit more examples into a prompt. We use temperature = 0.7 to promote interesting and varied output: the effect of temperature is illustrated in Fig. 6 by the multiple outputs.

We have two large datasets (Sect. 2), but we focus on GPT-Neo and Jurassic-1 few-shot (2, 3, 10) experiments, for two prompt formats, since format matters for Jurassic-1 [25]. In the QA format in Fig. 5, the prompt instances consist of input MRs marked as the PROMPT and the response marked as SENTENCE. The S2S format in the top of Fig. 6 simply separates the MR and text into two lines, with an empty line separating test instances. In Fig. 6, the 2-shot conditioning prompts are from the *music* and *movies* domains, and the test item is from the *restaurant* domain. We generate multiple outputs, shown in italics, to illustrate the effect of temperature. All of the outputs are natural and coherent. Only the last output, *Babbo restaurant is rated outstanding. I heard they serve great food*, fails to realize all the MR attributes, missing eatType = bistro and food = French.

The 2-shot experiments are intended to create a challenging task for testing the models' ability to generalize. In addition to 2-shot conditioning with the two examples in Fig. 6, the test set consists of novel hand-crafted MRs that are currently not

---

[3] Experiments with GPT-2-small showed that models tuned with the ∼32K train did not generalize to unseen relations within the tuning domain, such as from the DIRECTOR to the SCREEN WRITER relation, nor did these models generalize across domains.

Fig. 5 Input format similar to QA

[PROMPT]: confirm = yes | name = Tony Hawk's Pro Skater 3 | release_year = 2001 | genres = sport
[SENTENCE]: Gotcha! So you're referring to the Tony Hawk's Pro Skater 3 sports game, which was released in 2001?

Fig. 6 S2S Input format. Jurassic example, generating multiple outputs (in italics) with no in-domain conditioning

Starship = song = We Built This City | We Built This City = genre = pop rock
Starship plays pop rock like the song We Built This City. Do you like that genre?

Scream = cast member = Liev Schreiber
Liev Schreiber was really good in Scream, don't you agree?.

name=Babbo | eatType = bistro | food = French | customerRating = outstanding

- Babbo's bistro and restaurant serves French cuisine. The food is outstanding, according to customer reviews.
- Babbo is an outstanding French bistro in NY. Do you like French food?
- Babbo is a bistro. French food is the best. Babbo has outstanding food.
- Babbo restaurant is rated outstanding. I heard they serve great food.

Fig. 7 Sample Novel Test Meaning Representations used to test 2-shot prompt-based conditioning

ID	Topic	Novel Relations MR
M1	**Movies**	(Despicable Me, screen writer, Cinco Paul)
M2	**Music**	(The Beach Boys, song, Cotton Fields), (Cotton Fields, date, 1970)
M3	**TV**	(Desperate Housewives, narrative location, Fairview)
M4	**Sports**	(Muhammad Ali, significant event, lighting the Olympic cauldron), (lighting the Olympic cauldron, of, 1996 Summer Olympics)

in Athena, which in some cases also use rare relations. The goal is to test how well the models do at realizing responses directly from the WikiData KG, without any domain-specific or relation-specific conditioning. Table 6 illustrates a good case of generalization to the restaurant domain. Table 1 indicates with a * those entities and relations corresponding to the novel MRs in our test set, and example novel MRs for each topic domain are in Fig. 7.

For evaluation metrics, we use BLEURT along with human evaluation for the following metrics: (1): **coherence**: makes sense and is natural; (2) **semantic accuracy**: triples realized divided by total triples for the KG-RGs and attributes realized divided by total attributes for Viggo; (3) **good hallucinations**: additional **true** information, not specified in the MR, is added to the utterance from the LM's own knowledge; (4) **bad hallucinations**: additional **false** information is added to the utterance from the LM's own knowledge; (5) **dialogue act accuracy**: whether the output utterance matches the dialogue act specified for Viggo, exemplified in the outputs in Fig. 4; (6) whether a question is added to the end of the response, that was not specified in the MR or by the dialogue act, as seen in the second example output in Fig. 6. Remember that no dialogue acts are specified by the MRs for the Athena-KG-RGs, but that some of the Viggo dialogue acts, such as *suggest*, typically are realized as questions or include a question. For the 2-shot experiments with the novel MRs, there are no reference utterances and BLEURT scores cannot be calculated, so we use the human evaluation metrics.

It is important to note that BLEURT scores by themselves are not intended to mean anything: they are only useful for comparing models [39]. In addition, BLEURT, like other n-gram scoring metrics, doesn't account for stylistic variation which is often desirable [17, 29]. Also, previous work shows that the correlation of BLEURT to human ratings of naturalness varies across conversational domains [47]. However, that work was based on crowd-sourced open-domain dialogues where both sides of the dialogue were produced by humans. Here it might be expected that BLEURT would be a good predictor of semantic accuracy. Therefore, we use BLEURT as first indicator of a model's performance and use BLEURT scores to decide whether to perform the human evaluation on a model's output. Then we examine whether the BLEURT scores are highly correlated with the human metrics for coherence and semantic accuracy.

4 Experimental Results

We report results for all the KG-RG topics and for Viggo, with both GPT-Neo and Jurassic-1. The models were also conditioned and tested for both the QA format in Fig. 5 and the S2S format in Fig. 6. For the KG-RG topics, we also experiment with all possible cross-domain combinations of conditioning and test.

Few-Shot Knowledge-Graph Response Generation. For each topic (movies, music, sports, and TV), we randomly select ten instances for conditioning and 50 for testing (200 total). We tune Jurassic-1 and GPT-Neo with each conditioning set and then test each model on all four topics (test on 200) to examine both within and cross-domain few-shot performance. Table 2 provides the BLEURT results for both Athena-GPT-Neo and Athena-Jurassic and for both S2S and QA formats. Rows indicate the conditioning domain, while columns indicate test domains. The diagonal of each subtable reports within-domain performance. The average BLEURT scores over all topics for each conditioning set are in the last column of each subtable, and averages for each input format (S2S or QA) are also included.

As expected, the within-domain results (highlighted in yellow) show that the models perform best when prompts are from their own domain. The best results for in-domain conditioning are for sports, with an average BLEURT score of 0.23 for the S2S format for Jurassic, and 0.26 for the S2S format for GPT-Neo, as well as a 0.21 for the QA format for GPT-Neo. The within-domain performance for the TV domain is also good, with a score of 0.22 for the QA format for GPT-Neo, and a score of 0.17

Table 2 BLEURT scores for testing within and across domain for Athena-Jurassic and Athena-GPT-Neo. Prompt inputs in either S2S or QA format, conditioning on 10 instances of each topic

format\model	Jurassic						GPT-Neo					
S2S	Jurassic S2S	test					GPT-Neo S2S	test				
	train	movies	music	tv	sports	all	train	movies	music	tv	sports	all
	movies	**-0.03**	-0.65	-0.46	-0.54	-0.42	movies	**0.02**	-0.49	-0.4	-0.44	-0.33
	music	-0.10	**-0.10**	-0.25	-0.49	-0.28	music	-0.39	**-0.08**	-0.47	-0.62	-0.39
	tv	-0.52	-0.59	**0.17**	-0.57	-0.38	tv	0.02	-0.49	**-0.40**	-0.44	-0.33
	sports	-0.49	-0.43	-0.42	**0.23**	-0.28	sports	-0.44	-0.62	-0.34	**0.26**	-0.28
Average S2S		-0.29	-0.44	-0.24	-0.34	-0.34		-0.20	-0.42	-0.40	-0.31	-0.33
QA	Jurassic QA	test					GPT-Neo QA	test				
	train	movies	music	tv	sports	all	train	movies	music	tv	sports	all
	movies	**-0.23**	-0.66	-0.48	-0.57	-0.49	movies	**-0.26**	-0.45	-0.27	-0.43	-0.35
	music	-0.38	**-0.38**	-0.18	-0.44	-0.34	music	-0.54	**-0.39**	-0.33	-0.57	-0.46
	tv	-0.69	-0.84	**-0.16**	-0.79	-0.62	tv	-0.45	-0.50	**0.22**	-0.58	-0.33
	sports	-0.54	-0.62	-0.46	**-0.12**	-0.44	sports	-0.61	-0.60	-0.40	**0.21**	-0.35
Average QA		-0.46	-0.63	-0.32	-0.48	-0.47		-0.47	-0.49	-0.20	-0.34	-0.37

for Jurassic for the S2S format. Interestingly, sometimes a specific topic's prompts perform as well or better for another topic than its own (highlighted in turquoise), e.g., GPT-NEO S2S conditioned with TV prompts performs better on movies than TV, and Jurassic QA, when conditioned with music prompts, performs better for TV. This could arise because two domains are similar (TV and movies) or because one domain is easier, e.g., the averages across the columns of each section suggest that TV is easier.

The averages also clearly indicate that, for Jurassic, the S2S format works better, with large differences across all topic columns and topic diagonals, and an overall S2S of -0.34 compared to QA of -0.47 ($p < 0.01$). For GPT-Neo, the overall differences between S2S (-0.33) and QA (-0.37) are not significant, and the story is more complex because GPT-Neo QA works well for both TV (0.22) and sports (0.21). The differences between S2S and QA are not significant for TV or movies, but GPT-Neo S2S is significantly better than GPT-Neo QA for music and sports.

A comparison of BLEURT scores for S2S for Jurassic versus GPT-Neo for each topic shows that GPT-Neo is significantly better for Movies ($p = 0.007$), Jurassic is significantly better for music ($p = 0.005$), GPT-Neo shows a trend to be better for TV ($p = 0.07$), and there are no differences for Sports ($p = 0.87$). However, a paired t-test comparing BLEURT scores across all topics for both GPT-Neo and Jurassic shows that the overall differences are not significant.

Since the overall differences for GPT-Neo S2S are not significantly different than GPT-Neo QA, we focus the human evaluation on comparing Athena-Jurassic to Athena-GPT-NEO for the S2S format. This will allow us to directly compare the human metrics for the two models while the prompt format is fixed. We restrict the annotation to the within-domain testing. We sampled 30 of the 50 test examples for each topic (240 examples). Three experts familiar with Athena labelled each output for coherence, semantic accuracy, good and bad extra information (hallucinations), and whether a question was added to the end of the response (remember that no

Table 3 Human Metrics for GPT-Neo versus Jurassic per Topic

model	GPT-Neo					
metric	Coher	SemAcc	GoodH	BadH	AddQ	#Words
movies	2.97	0.78	17%	17%	33%	22.37
music	2.60	0.49	63%	7%	100%	21.27
TV	2.87	0.95	0%	3%	100%	21.63
sports	2.87	0.85	7%	17%	0%	20.40
mean	2.83	0.77	22%	11%	58%	21.42
model	Jurassic					
metric	Coher	SemAcc	GoodH	BadH	AddQ	#Words
movies	3.00	0.91	17%	7%	53%	20.70
music	2.87	0.89	7%	10%	97%	18.71
TV	2.97	0.89	0%	3%	100%	20.10
sports	3.00	0.88	10%	17%	0%	20.60
mean	2.96	0.89	9%	9%	63%	20.03

dialogue acts were specified in the Athena-KG MRs). We also counted the number of words in each output to measure some aspects of the style of the outputs.

Table 3 presents the results for the human metrics, showing that the average coherence (**Coher**) for Athena-GPT-Neo is significantly lower than Athena-Jurassic ($p = 0.002$), as well as the semantic accuracy (**SemAcc**) ($p = 0.002$). Athena-GPT-Neo hallucinates much more (total 33%) than Athena-Jurassic (total 18%). Although 22% of Athena-GPT-Neo's hallucinations are good, 11% of the utterances include at least one bad hallucination, in addition to failing to realize the semantic triples included in the MRs.

Both models frequently add questions, which can be considered as either a feature or a bug. They are usually quite natural, but the Athena-KG MRs do not specify dialogue acts. Thus, there is no way to control whether the questions appear or not. The models use a similar number of words. Still, given the low semantic accuracy of Athena-GPT-Neo, many words in the GPT-Neo outputs must be due to hallucinations.

The human evaluation suggests large differences between Athena-GPT-Neo and Athena-Jurassic in terms of the quality of the output, but there were no significant differences in overall BLEURT scores between GPT-Neo S2S and Jurassic S2S. We, therefore, test whether BLEURT scores are meaningful proxies for the human metrics of coherence and semantic accuracy, finding that the correlation across all the topics, for each model, indicates that BLEURT scores do correlate with semantic accuracy, with a correlation of 0.65 for Athena-GPT-Neo, and 0.46 for Athena-GPT-Jurassic. Coherence is not correlated with BLEURT, with correlations of 0.02 for Athena-GPT-Neo and −0.05 for Athena-Jurassic.

Figure 8 shows examples for the two S2S models for each domain when tuned on within-domain prompts, which illustrate the strengths and weaknesses between models. The Athena-GPT-Neo output for M1 was labelled a 3 for coherence. However, it leaves out the triple (BAFTA Award for Best Short Film, show, 47th British Academy

ID	Topic	Input MR	Output
		Athena-GPT-Neo	
M1	**Movies**	(Peter Capaldi, award, BAFTA Award for Best Short Film), (BAFTA Award for Best Short Film, show, 47th British Academy Film Awards), (BAFTA Award for Best Short Film, work, Franz Kafka's It's a Wonderful Life)	I think it's really great when a talented actor wins an award. do you think Peter Capaldi deserved to win a BAFTA Award for Best Short Film in 1980, for Franz Kafka's It's a Wonderful Life?
M2	**Music**	(Kellie Pickler, song, Red High Heels), (Red High Heels, genre, country music)	Kellie Pickler is a country singer, and she's also a rapper. Do you know her songs?
		Athena-Jurassic	
M3	**TV**	(Saturday Night Live, award, Primetime Emmy Award for Outstanding Variety Sketch Series), (Saturday Night Live, date, 2019)	Saturday Night Live won a Primetime Emmy Award for Outstanding Variety Sketch Series in 2019. How does the fact that it got this award affect your opinion of the show?
M4	**Sports**	(Len Ford, member of sports team, Los Angeles Dons), (Len Ford, position played on team, end)	Did you know that Len Ford has played as a part of famous teams, such as the Los Angeles Dons, and played positions such as end.

Fig. 8 Jurassic-1 and GPT-NEO outputs for conditioning with 10 within-domain prompts

Film Awards). It also includes the bad hallucination that Peter Capaldi is an actor, when in fact he wrote and directed the film. In addition, the 47th British Academy Film Awards honoured the best films of 1993, so Peter Capaldi won this award in 1994, not in 1980. The semantic accuracy annotation indicates that 2/3 triples are correct, the output includes two bad hallucinations, and the output includes a question. Similarly, the GPT-Neo output for M2 shows that GPT-Neo knows that Kellie Pickler is a rapper, knowledge that was not included in the MR. This was hand-annotated as a good hallucination. However, this output fails to realize the triple (Kellie Pickler, song, Red High Heels), so semantic accuracy was 1/2 triples.

The Athena-Jurassic output for M3 was labelled as a 3 for coherence, and that it includes a question. The output correctly realizes all the triples so it was marked as semantically perfect (3/3 triples realized). The output for M4 is also labelled as a 3 for coherence. It also correctly realizes all the triples (2/2), which are realized by a *Did you know* question. This output would not be annotated as including an additional question since the material in the *Did you know* question is part of the specified content in the MR.

2-Shot prompting on Novel Entities and Relations.

We also performed 2-shot experiments using the two prompt instances for movies and music in Fig. 9. Because the realizations of each relation or sets of relations require a template to be written for Athena's current KG-RGs, Athena has no templates for relations that are sparsely populated. Thus, we test 80 MRs composed of entities,

(Starship = song = We Built This City ǀ We Built This City = genre = pop rock) Starship plays pop rock like the song We Built This City. Do you like that genre?
(Planet of the Apes = cast member = Felix Silla) I heard Felix Silla starred in a good movie, called Planet of the Apes.

Fig. 9 Two prompt instances used with Jurassic-1 for 2-Shot Novel MR Experiments

Table 4 Human evaluation for 2-shot Novel Athena-Jurassic versus Athena-GPT-Neo, prompted with S2S format

model	GPT-Neo				
metric	Coher	SemAcc	GoodH	BadH	AddQ
novel	2.58	0.41	15%	12%	7%
model	Jurassic				
novel	2.80	0.72	13%	4%	47%

relations, or combinations of relations that are novel to Athena, as indicated by a * in Table 1. We only use the S2s prompt format since the results in Table 2 show that the S2S format is clearly better for Jurassic and there are no differences for GPT-Neo.

Since there are no reference utterances for the novel MRs, BLEURT scores cannot be calculated. Table 4 provides the results of the human evaluation, which indicate stark differences between Athena-GPT-Neo and Athena-Jurassic, reflecting the reduction to 2-shot prompting, and the rareness of the KG relations in the novel MRs. The novel test MRs are also harder for both types of models, but the performance of Athena-GPT-Neo declines more sharply. Compare the mean coherence of 2.58 and semantic accuracy of 0.41 for Athena-GPT-Neo in Table 4 to the mean coherence of 2.83 and semantic accuracy of 0.77 for Athena-GPT-Neo in Table 3. Then compare the mean coherence of 2.80 and semantic accuracy of 0.72 for Athena-Jurassic in Table 4, to the mean coherence of 2.96 and a semantic accuracy of 0.89 for Athena-Jurassic in Table 3. When testing with the novel MRs, Athena-GPT-Neo only adds questions 7% of the time, but Athena-Jurassic maintains a higher level and adds questions 47% of the time. When testing with the novel MRs, GPT-Neo hallucinates bad information in 12% of turns, while Jurassic only does so in 4% of turns. This supports the claim by Lieber et al. that Jurassic should generalize better [25].

Figure 10 provides novel MRs and outputs for all four topics for both Athena-GPT-Neo and Athena-Jurassic that illustrate the differences between the models. In N1, Athena-GPT-Neo produces a coherent and semantically accurate utterance about The Black Cauldron that includes a bad hallucination of Stephen King as the author, when Lloyd Alexander is the author. In N2, Athena-GPT-Neo again produces a coherent utterance, but the content of that utterance doesn't include **any** of the triples in the MR, only matching the name of the singer, Dua Lipa. In N3, the output is again coherent, but it fails to realize the triple (James Roday, character role, Shawn Spencer). Example N4 illustrates how GPT-Neo sometimes produces **redundant** or

ID	Domain	Input MR	Output
		Athena-GPT-Neo	
N1	Movies	(The Black Cauldron, genre, film based on literature)	The Black Cauldron is a film based on the novel of the same name by Stephen King.
N2	Music	(Dua Lipa, award, Grammy Award for Best New Artist), (Grammy Award for Best New Artist, show, 61st Annual Grammy Awards)	Dua Lipa is a singer-songwriter from the Philippines. She is known for her hit songs "I'm Not the One" and "I'm Not the One (Remix)".
N3	TV	(Psych, cast member, James Roday), (James Roday, character role, Shawn Spencer)	I heard James Roday starred in a good movie, called Psych.
N4	Sports	(Tom Brady, award, Associated Press Athlete of the Year), (Associated Press Athlete of the Year, date, 2007)	Tom Brady is the most valuable player in the NFL. He is also the most valuable player in the NFL.
		Athena-Jurassic	
N5	Movies	(Ghost Rider, producer, David S. Goyer)	Have you heard of a movie called Ghost Rider? I heard it was produced by David S. Goyer.
N6	Music	(Adele, award, Grammy Award for Song of the Year), (Grammy Award for Song of the Year, show, 54th Annual Grammy Awards), (Grammy Award for Song of the Year, work, Rolling in the Deep)	Adele won an award for Song of the Year at the 54th Annual Grammy Awards. Do you like her song Rolling in the Deep?
N7	TV	(Elementary, assessment, Bechdel test), (Bechdel test, assessment outcome, passes)	Elementary passes the Bechdel test because at least two of the named women talk to each other about something other than a man.
N8	Sports	(Usain Bolt, record held, Men's 100 metres world record progression), (Usain Bolt, race time, 9.58 seconds), (Usain Bolt, start time, 2009)	Usain Bolt holds the record for the men's 100 metres. He broke the previous record, which was 9.69 seconds, at the 2009 World Championships in Berlin, Germany.

Fig. 10 Sample outputs for novel unseen relations and entities for both Jurassic-1 and GPT-Neo

logically inconsistent outputs, where here it says the same thing about Tom Brady twice, but sometimes it repeats itself many times, e.g., *Friday Night Lights is a movie about a small town in Texas that is run by a family of criminals. The town is run by a family of criminals. ...(4 times)*. In other cases, Athena-GPT-Neo contradicts itself. There are no examples from Athena-Jurassic that are redundant or logically inconsistent. In future work, these categories could be added to the human metrics, even though they happen rarely.

Figure 10 also shows that Athena-Jurassic's 2-shot outputs are remarkably good. In N5, the output is coherent, semantically correct, and stylistically interesting. In N6, all three triples are realized correctly, and the last triple is embedded into a question, which seems very natural. In N7, Athena-Jurassic realizes all the content

Table 5 BLEURT for Viggo comparing GPT-Neo and Jurassic, S2S format versus QA format, and 3 prompting instances versus 10

model	GPT-Neo				
metric	Coher	DA	SemAcc	GoodH	BadH
viggo	2.50	3.00	0.83	16%	12%
model	Jurassic				
viggo	2.85	2.70	0.73	29%	2%

in the MR, but also produces a good hallucination. defining what the Bechdel tests actually are. In N8, Athena-Jurassic seems to know a lot about Usain Bolt: it does not actually realize the triple (Usain Bolt, race time, 9.58 seconds), but provides the race time for the previous record and produces a good hallucination of the event that this happened at, namely the 2009 World Championships.

Few-Shot Response Generation for Viggo Video Games. We also experiment with few-shot prompt conditioning with the Viggo corpus, with a focus on the realization of dialogue acts. Athena-KG MRs do not specify the dialogue act, and thus its use of questions cannot be controlled. The dialogue acts in Viggo are shown in Figs. 3 and 4. The Viggo experiments compare prompt conditioning with GPT-Neo and Jurassic, for both S2S and QA formats, and compare 3-shot conditioning to 10-shot conditioning per dialogue act. All experiments use a randomly selected set of 100 Viggo test items.
Table5 provides the BLEURT scores for these prompting variations. The QA row in Table 5 for 3-shot conditioning suggests that the QA format performs better for 3-shot than the S2S format. However, for 10-shot conditioning, S2S is better for both GPT-Neo and Jurassic. The BLEURT scores for Athena-Jurassic for 10-shot conditioning are clearly much better than any of the other results.

Table 6 provides the results of the human evaluation showing that Athena-GPT-Neo does better at matching the dialogue acts (DA) and achieves higher semantic accuracy (0.83) than Athena-Jurassic. However, overall the results are good for controlling dialogue acts. Table 6 also shows however that the coherence of Athena-GPT-Neo is poor compared to Athena-Jurassic. Athena-Jurassic demonstrates the knowledge that Jurassic-1 has about video games by producing far more good hallucinations (29%), while only producing 2% of bad hallucinations, as compared to the 12% of bad hallucinations by Athena-GPT-Neo.

Sample outputs for both Viggo models are shown in Fig. 11. Both models produce good outputs much of the time. The low coherence of Athena-GPT-Neo primarily arises from the redundancy of repeating the same information twice. Compare example V1 to example V3. Both models get the dialogue act of *confirm* correct. However, the Athena-Jurassic output in V3 is more varied and interesting and includes all the attributes specified in the MR. The Athena-GPT output in V1 misses out the release year attribute and uses an odd phrasing *that has got a sports genre* that results in a lower coherence score. Example V2 illustrates Athena-GPT-Neo's knowledge about video games, with a good hallucination that it is rated E for Everyone, and that means

Table 6 Human metrics for Viggo comparing GPT-Neo and Jurassic

model	GPT-Neo	Jurassic	GPT-Neo	Jurassic
tune#	3	3	10	10
S2S	-0.36	-0.30	-0.21	-0.06
QA	-0.28	-0.21	-0.22	-0.11

ID	Input MR	Output
	Athena-GPT-Neo	
V1	confirm = yes, name = Tony Hawk's Pro Skater 3, release_year = 2001, genres = sport	You mean the Tony Hawk's Pro Skater 3 that has got a sport genre?
V2	give_opinion = yes, name = Might & Magic: Heroes VI, rating = average, player_perspective = bird view, platforms = PC	Might & Magic: Heroes VI is a solid game. I like that it has a multiplayer and, since it's only rated E (for Everyone), I can play it with friends and younger siblings.
	Athena-Jurassic	
V3	confirm = yes, name = Tony Hawk's Pro Skater 3, release_year = 2001, genres = sport	Gotcha! So you're referring to the Tony Hawk's Pro Skater 3 sports game, which was released in 2001?
V4	give_opinion = yes, name = Sid Meier's Civilization V, rating = good, available_on_steam = yes, has_linux_release = yes	4X games can be really fun to play, and Civilization V is a pretty good example. It also happens to be available on Linux, and that's a huge plus for me.

Fig. 11 Viggo outputs with 10-shot conditioning per dialogue act for QA prompt format

that it can be played with younger siblings. Example V4 illustrates Athena-Jurassic failing to realize some of the MR attributes, such as the availability on Steam and the full name of the game. However, the language is again very natural, e.g., *that's a huge plus for me*.

5 Conclusion

We used prompt-based learning to create new neural models for semantically controlled meaning-to-text (M2T) natural language generators (NLGs) to improve the quality and the coverage of the M2T response generators in Athena, an open-domain dialogue system that has been a finalist in the Alexa Prize for the last two years [16, 28]. A major challenge for such systems is the need to produce truthful, high-quality responses on any topic. We created Athena-GPT-Neo and Athena-Jurassic using GPT-Neo [1] and Jurassic-1 [25], by experimenting with few-shot (2, 3, 10) prompt-based learning for Athena's knowledge-graph domains of movies, music, TV, and sports and with the Viggo corpus's dialogue act-based MRs for video games. We also experimented with multiple prompt formats and with testing both within and across-domain. The ability to create NLGs that generate high-quality responses directly from MRs via few-shot prompt conditioning will greatly facilitate the use of M2T NLGs in dialogue systems. To our knowledge, these are the first results demonstrating that few-shot prompt-based learning can create M2T NLGs that generalize well to new semantic domains.

Athena-Jurassic produces high-quality, semantically controlled, conversational responses directly from MRs and KG triples. These results confirm the choice that the Jurassic-1 creators made to use a larger vocabulary with phrasal tokens, and less depth and more width, in order to create a model that generalizes better [24, 25]. Our results show that both Athena-GPT-Neo and Athena-Jurassic generally produce coherent output with 10-shot within-domain conditioning, but that Athena-Jurassic is significantly better for both coherence and semantic accuracy. While we have not tested whether real-time response generation is possible, we believe the responses are generally of high enough quality to be used in settings with real human users, such as the Alexa Prize [11, 16, 28, 42]. We plan to do additional experiments with Viggo in order to improve its performance to the level required [21].

We also showed that Athena-Jurassic performs well with 2-shot conditioning, using completely novel sets of KG triples with unseen relations and entities. These novel MRs are not currently included in Athena, because the relations are rare, and creating templates for novel relations or sets of relations is typically not worth the human effort [34, 35]. For example, the MR in M4 in Fig. 7 describes the event of Muhammed Ali lighting the Olympic torch in 1996, a rarely populated event for the athlete entity type. Athena-Jurassic achieves a semantic accuracy of 2.72 out of 3 for MRs like this in our challenging 2-shot setting.

In experiments with the KG response generators in Athena, we found that in almost half the responses, Athena-Jurassic adds questions to the end of the response, which are typically quite natural. However, the use of questions cannot be controlled because the KG-RG meaning representations do not specify dialogue acts. Thus, we also experimented with few-shot conditioning for controlling dialogue acts using the MRs in the Viggo video games corpus. We showed that both Athena-GPT-Neo and Athena-Jurassic can learn to control dialogue acts with 10-shot conditioning per dialogue act. However again, Athena-Jurassic performs significantly better on the human metrics of coherence and semantic accuracy. Interestingly, often Athena-GPT-Neo successfully produces the *form* or *syntax* of the dialogue act, e.g., a verify-attribute dialogue act, while getting very few of the MR attributes correct. For example, Athena-GPT-Neo produces *You said you liked Assassin's Creed Chronicles: India. Do you think it would have been better to make it a single-player only game?* when the reference utterance is *So I know you said you hated Assassin's Creed Chronicles: India. Do you think all of Climax Studios side view games are as bad?*. Here, Athena-GPT-Neo only gets the name attribute correct and misses the attributes that it is single-player, the user-rating is poor, and the developer is Climax Studios.

We also presented automatic evaluation results using BLEURT for cross-domain testing. Some of the BLEURT results are very good and suggest that cross-domain 10-shot conditioning can also produce high-quality utterances. Our results also show that BLEURT scores have good correlation with the human metric of semantic accuracy, but not coherence. Future work should evaluate these cross-domain results with human metrics. It would also be valuable to experiment with a large number of recently proposed automatic evaluation metrics to test whether there are better metrics than BLEURT for doing automatic evaluation in this task setting [19, 47]. Many recently proposed automatic metrics rely on evaluating outputs within a dialogue context, which typically is not available in M2T NLG experiments. However, there are also novel reference-free metrics that could be tested in this setting.

There are many other possibilities with both the WikiData knowledge-graph RGs and with corpora such as Viggo for prompt-based learning and testing regimes that we have not yet experimented with or fully evaluated. We also plan to carry out future experiments on a number of other challenging problems for NLG [17, 29, 30, 37].

References

1. Black S, Gao L, Wang P, Leahy C, Biderman S (2021) Gpt-neo: Large scale autoregressive language modeling with mesh tensorflow. If you use this software, please cite it using these metadata
2. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. In: Advances in neural information processing, NeurIPS2020
3. Budzianowski P, Vulić I (2019) Hello, it's gpt-2-how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. In: Proceedings of the 3rd workshop on neural generation and translation, pp 15–22
4. Budzianowski P, Wen T-H, Tseng B-H, Casanueva I, Ultes S, Ramadan O, Gasic M (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, pp 5016–5026
5. Alessandra C, Chandra K, Rahul G, Behnam H, Anu V, Dilek H-T, Gabriel R (2019) A case study for open domain question answering. In: INLG, Natural language generation at scale
6. Chen C-Y, Yu D, Wen W, Mang Yang Y, Zhang J, Zhou M, Jesse K, Chau A, Bhowmick A, Iyer S, et al (2018) Gunrock: Building a human-like social bot by leveraging large scale real user data. Alexa prize proceedings
7. Colin E, Gardent C, Mrabet Y, Narayan S, Perez-Beltrachini L (2016) The webnlg challenge: Generating text from dbpedia data. In: Proceedings of the 9th international natural language generation conference, pp 163–167
8. Curry AC, Papaioannou I, Suglia A, Agarwal S, Shalyminov I, Xu X, Dušek O, Eshghi A, Konstas I, Rieser V, et al (2018) Alana v2: entertaining and informative open-domain social dialogue using ontologies and entity linking. Alexa prize proceedings
9. Eric M, Goel R, Paul S, Sethi A, Agarwal S, Gao S, Hakkani-Tur D (2019) Multiwoz 2.1: multi-domain dialogue state corrections and state tracking baselines (2019). arXiv:1907.01669
10. Fang H, Cheng H, Sap M, Clark E, Holtzman A, Choi Y, Smith NA, Ostendorf M (2018) Sounding board: a user-centric and content-driven social chatbot. NAACL HLT 2018, p 96
11. Gabriel R, Liu Y, Gottardi A, Eric M, Khatri A, Chadha A, Chen Q, Hedayatnia B, Rajan P, Binici A et al (2020) Further advances in open domain dialog systems in the third alexa prize socialbot grand challenge. Alexa prize proceedings

12. Gao L, Biderman S, Black S, Golding L, Hoppe T, Foster C, Phang J, He H, Thite A, Nabeshima N, et al (2020) The pile: an 800gb dataset of diverse text for language modeling (2020). arXiv:2101.00027
13. Gardent C, Shimorina A, Narayan S, Perez-Beltrachini L (2017) Creating training corpora for NLG micro-planners. In: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers), pp 179–188
14. Gopalakrishnan K, Hedayatnia B, Chen Q, Gottardi A, Kwatra S, Venkatesh A, Gabriel R, Hakkani-Tür D, Alexa AI A (2019) Topical-chat: towards knowledge-grounded open-domain conversations. In: Proceedings of Interspeech, pp 1891–1895
15. Ham D, Lee J-G, Jang Y, Kim K-E (2020) End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 583–592
16. Harrison V, Juraska J, Cui W, Reed L, Bowden KK, Wu J, Schwarzmann B, Ebrahimi A, Rajasekaran N, Varghese N, et al (2020) Athena: constructing dialogues dynamically with discourse constraints. In: Proceedings of the Alexa Prize 2020
17. Harrison V, Reed L, Oraby S, Walker M (2019) Maximizing stylistic control and semantic accuracy in nlg: personality variation and discourse contrast. DSNNLG 2019, p 1
18. Hedayatnia B, Kim S, Liu Y, Gopalakrishnan K, Eric M, Hakkani-Tur D (2020) Policy-driven neural response generation for knowledge-grounded dialogue systems. arXiv:2005.12529
19. Howcroft DM , Belz A, Clinciu M-A, Gkatzia D, Hasan SA, Mahamood S, Mille S, van Miltenburg E, Santhanam S, Rieser V (2020) Twenty years of confusion in human evaluation: Nlg needs evaluation sheets and standardised definitions. In: Proceedings of the 13th international conference on natural language generation, pp 169–182
20. Juraska J, Bowden KK, Walker M (2019) ViGGO: a video game corpus for data-to-text generation in open-domain conversation. In: Proceedings of the 12th international conference on natural language generation
21. Juraska J, Walker M (2021) Attention is indeed all you need: semantically attention-guided decoding for data-to-text nlg. In: International conference on natural language generation, INLG
22. Kedzie C, McKeown K (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
23. Lebret R, Grangier D, Auli M (2016) Generating text from structured data with application to the biography domain. In: Proceedings of empirical methods in natural language processing (EMNLP)
24. Levine Y, Wies N, Sharir O, Bata H, Shashua A. (2020) The depth-to-width interplay in self-attention. arXiv:2006.12467
25. Lieber O, Sharir BL, Shoham Y (2021) Jurassic-1: technical details and evaluation. Technical report, AI21 Labs
26. Moon S, Shah P, Kumar A, Subba R (2019) Opendialkg: explainable conversational reasoning with attention-based walks over knowledge graphs. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 845–854
27. Novikova J, Dušek O, Rieser V (2017) The e2e dataset: new challenges for end-to-end generation. In: Proceedings of the 18th annual SIGdial conference
28. Kevin K. Bowden Juraj Juraska Wen Cui Vrindavan Harrison Rishi Rajasekaran Angela Ramirez Cecilia Li Eduardo Zamora Phillip Lee Jeshwanth Bheemanpally Rohan Pandey Adwait Ratnaparkhi Omkar Patil, Lena Reed and Marilyn Walker. Athena 2.0: Discourse and user modeling in open domain dialogue. *Proc. of the Alexa Prize*, 2021
29. Oraby S, Harrison V, Ebrahimi A, Walker M (2019) Curate and generate: a corpus and method for joint control of semantics and style in neural nlg. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 5938–5951
30. Oraby S, Reed L, Sharath TS, Tandon S, Walker M (2018)Neural multivoice models for expressing novel personalities in dialog. In: Proceedings of Interspeech 2018, pp 3057–3061

31. Pang B, Nijkamp E, Han W, Zhou L, Liu Y, Tu K (2020) Towards holistic and automatic evaluation of open-domain dialogue generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 3619–3629. Association for Computational Linguistics
32. Paranjape A, See A, Kenealy K, Li H, Hardy A, Qi P, Sadagopan KR, Minh Phu N, Soylu D, Manning CD (2020) Neural generation meets real people: towards emotionally engaging mixed-initiative conversations. arXiv:2008.12348
33. Radford A, Jeffrey W, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI blog 1(8):9
34. Rambow O, Bangalore S, Walker M (2001) Natural language generation in dialog systems. In: Proceedings of the first international conference on human language technology research
35. Rambow O, Rogati M, Walker M (2001) Evaluating a trainable sentence planner for a spoken dialogue system. In: Proceedings of the 39th annual meeting of the association for computational linguistics, pp 434–441
36. Rastogi A, Zang X, Sunkara S, Gupta R, Khaitan P (2020) Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 8689–8696
37. Reed L, Harrison V, Oraby S, Hakkani-Tur D, Walker M (2020) Learning from mistakes: Combining ontologies via self-training for dialogue generation. In: Proceedings of the 21st annual meeting of the special interest group on discourse and dialogue (SIGDIAL 2020)
38. Sakaguchi K, Le Bras R, Bhagavatula C, Choi Y (2020) Winogrande: an adversarial winograd schema challenge at scale. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 8732–8740
39. Sellam T, Das D, Parikh A (2020) Bleurt: learning robust metrics for text generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 7881–7892
40. Shalyminov I, Dušek O, Lemon O (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, pp 1–8
41. Tran V-K, Le Nguyen M, Tojo S (2017) Neural-based natural language generation in dialogue using rnn encoder-decoder with semantic aggregation. In: Proceedings of the 18th annual SIGdial meeting on discourse and dialogue, pp 231–240
42. Venkatesh A, Khatri C, Ram A, Guo F, Gabriel R, Nagar A, Prasad R, Cheng M (2017) On evaluating and comparing conversational agents
43. Walker MA, Langkilde-Geary I, Hastie HW, Wright JH, Gorin AL et al (2002) Automatically training a problematic dialogue predictor for a spoken dialogue system. J Artif Intell Res 16(1):293–319
44. Walker M, Smither A, Oraby S, Harrison V, Shemtov H (2018) Exploring conversational language generation for rich content about hotels. In: Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)
45. Wiseman S, Shieber SM, Rush AM (2017) Challenges in data-to-document generation. CoRR, abs/ arXiv:1707.08052
46. Yavuz S, Rastogi A, Chao G-L, Hakkani-Tür D, Alexa AI A (2019) Deepcopy: grounded response generation with hierarchical pointer networks. In: 20th annual meeting of the special interest group on discourse and dialogue, p 122
47. Yeh Y-T, Eskenazi M, Mehri S (2021) A comprehensive assessment of dialog evaluation metrics. arXiv:2106.03706
48. Zellers R, Holtzman A, Bisk Y, Farhadi A, Choi Y (2019) Hellaswag: can a machine really finish your sentence? arXiv:1905.07830

49. Zhao C, Walker M, Chaturvedi S (2020) Bridging the structural gap between encoding and decoding for data-to-text generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 2481–2491
50. Zhou L, Gao J, Li D, Shum H-Y (2020) The design and implementation of xiaoice, an empathetic social chatbot. Comput Linguist 46(1):53–93

Spoken and Multimodal Interaction

Comparison of Automatic Speech Recognition Systems

Joshua Y. Kim, Chunfeng Liu, Rafael A. Calvo, Kathryn McCabe, Silas C. R. Taylor, Björn W. Schuller, and Kaihang Wu

Abstract High-quality transcription systems are required for conversational analysis systems. We compared two manual transcribers with five automatic transcription systems using video conferences from a medical domain and found that (1) manual transcriptions significantly outperformed the automatic services, and (2) the automatic transcription of YouTube Captions significantly outperformed the other ASR services.

Keyword Speech recognition

1 Introduction

Conversational analysis systems require high-quality transcription systems to extract the verbatim transcripts. The verbatim transcripts could then be used to train deep learning models as a separate modality in addition to audio and video streams [9, 10, 21, 24, 34], or the transcripts can be weaved together with other modalities to form a multimodal narrative that is human-centric [15, 16] and facilitate conversation visualization [14]. Although ASR systems are continually improving, there is little work that compares the performance of the widely available commercial systems.

J. Y. Kim · K. Wu
The University of Sydney, Sydney, Australia

C. Liu
Hello Sunday Morning, Surry Hills, Australia

R. A. Calvo (✉) · B. W. Schuller
Imperial College London, London, United Kingdom
e-mail: r.calvo@imperial.ac.uk

K. McCabe
University of California, Los Angeles, CA, USA

S. C. R. Taylor
University of New South Wales, Kensington, NSW, Australia

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_8

In this paper, we aim to provide empirical evidence on the performance of five ASR providers—namely, Google Cloud, IBM Watson, Microsoft Azure, Trint, and YouTube.

2 Related Works

ASR systems have seen significant improvements over the past few years [33]. The Switchboard telephone speech dataset is often used to benchmark the performance of the transcription [28]. Microsoft Research reports a WER of 5.1% on the NIST 2000 Switchboard task [33]. IBM Research reports 6.6% WER on the Switchboard subset of the Hub5 2000 evaluation test set [28]. Google Research reports a 6.7% WER on a 12,500-hour voice search dataset and 4.1% on a dictation task [3], both of which are not part of the Switchboard telephone speech dataset. Some works [23] relied on such published statistics which could be misleading.

Applications of the ASR in teleconferences are more challenging as the speaker is speaking at some distance from the microphone—this is known as distant speech recognition. Research on distant speech recognition includes the application of convolutional neural networks (CNN) [17] on the Augmented Multi-party Interaction (AMI) meeting corpus [2], where a word error rate of 40.9% was achieved with a single distant microphone [30]. More recently, Renals and Swietojanski [26] used the AMI corpus to compare ASR approaches using multiple distant microphones and individual headset microphones. The difference in WER is significant—the eight distant microphone setup achieved a WER of 52.0% verses 29.6% (individual microphone). The distant microphone performance was recently surpassed by UNet++ (WER: 42.2%) [35].

Këpuska and Bohouta [13] performed a comparison between CMU Sphinx, Microsoft Speech and Google Cloud and found that the Google Cloud API performs the best with a mean WER of 9%. In that study, the authors used the Texas-Instruments/Massachusetts Institute of Technology (TIMIT) corpus [5]. In this study, we expand the number of online transcription services for comparison and utilize a dataset that is intended to mirror real-world doctor-patient interviews, which has been increasing [7, 8, 25].

3 EQClinic Dataset

3.1 Data Collection

This study used data from the EQClinic platform [20]. Students in an Australian medical school were required to complete the program aimed at improving clinical communication skills during their first and second years of study. Within the

EQClinic platform, the students were required to complete at least one medical consultation with a simulated patient on the online video conferencing platform EQClinic [19]. Participants consist of twelve second-year undergraduate medical students (six female and six male) and two simulated patients (SP, one male and one female). The two SP were professional actors, recruited online and paid AUD35 per hour for participating. The study was approved by the UNSW Human Research Ethics Committee (Project Number HC16048).

3.2 Data Analysis

For each consultation, EQClinic generated one MP4 video recording for each speaker with a resolution of 640x480 pixels and a frame rate of 25fps. Audio recordings were extracted using the FFMpeg software. We selected twelve interview sessions randomly and we ensured that there are three videos for each of the possible gender pairing (male-male, male-female, female-male, and female-female).

The duration of these sessions ranges from 12 to 18 min (mean duration (SD) = 14.8 (2.0)). Each session contained two videos, and each of these video pairs had one speaker (the student or the SP). Each video comprised 668 to 1705 words (mean words (SD) = 1187 (316). In total, 24 videos and a total of 28,480 words were analyzed. Disfluencies like "um" are captured in the transcripts. We sent these 24 videos to seven transcription services—two of which were manual, and the other five were ASR systems. The costs and file formats required for transcription are summarized in Table 1 in the supplementary material. Although the file formats differ, we are interested in also testing services that could not accept videos as inputs.

For the two manual transcription services, one was an independent professional transcriber (CB), and the other was from an online network of hand-picked freelancers available at Rev.com (Rev). For both manual transcription services, video files were provided in the MP4 format for transcription.

Table 1 Summary of required file formats and costs for transcription services. CB denotes the independent professional transcriber. Rev denotes transcribers from Rev.com

Service	File Format	USD per video minute
Manual (CB)	MP4 Video	1.920
Manual (Rev)	MP4 Video	1.500
Automatic (Google Cloud)	Mono-channel FLAC audio	0.048
Automatic (IBM Watson)	Mono-channel FLAC audio	0.020
Automatic (Microsoft Azure)	Mono-channel WAV audio	0.008
Automatic (Trint)	MP4 Video	0.025
Automatic (YouTube)	MP4 Video	0.000

Each of the five ASR services (Google Cloud, IBM Watson, Microsoft Azure, Trint, and YouTube) required a different format of the input file to perform the transcription. For all of the five ASR services, we elected to perform asynchronous transcription service calls because YouTube and Trint do not offer synchronous transcription service calls. Synchronous service calls refer to the ability of the ASR to stream text results, immediately returning text as it is recognized from the audio.

We compared the quality of transcripts gathered from different transcription services. Word Error Rate (WER) is a popular performance measure in automatic speech recognition [4]. We first determined which of the two sets of manual transcriptions would be the reference transcript. We then compared the five sets of automatic transcriptions against this reference transcript to identify the best-performing ASR system. We posit that if multiple transcribers produce similar transcripts as indicated by low WER, they have likely converged on the correct transcription [27]. Therefore, the set of manual transcriptions with the lower WER as compared with each of the five sets of automatic transcription was considered the best choice as the set of reference transcripts.

In our analysis, ten pairwise WER were generated between each of the five hypothesis transcripts and the two manual sets of transcripts (Manual CB and Manual Rev) [1]. For the ten pairwise WER estimates, we determined which of the WER-reference pairs were statistically significantly different. To do that, we needed the 95% WER confidence interval. Since the assumption of independent error rates [6] are not applicable when we fixed the hypothesis transcript to be from one ASR service, we elected to use bootstrapping to generate confidence intervals. The bootstrap technique is used to quantify the uncertainty associated with the WER in our application and involves creating 10,000 bootstrap datasets [29] produced by random sampling with replacement [12]. With the 10,000 bootstrap samples, we computed an average WER. Then, we created the 95% WER confidence interval by eliminating the top and bottom 2.5% values.

After establishing the set of manual transcription was of higher quality, we used this set of manual transcription as our reference transcription to examine the WER of all other transcription services. Next, we investigated whether differences in WER performance between each transcription service were statistically significant. We used one set of reference transcriptions and computed the difference in WER between service X and service Y for each of the 24 transcriptions. Similarly, we then bootstrapped the differences in WER between the two services (service X and Y) and generated the confidence intervals for the differences using 10,000 samples.

4 Results

Figure 1 compares the hypothesis transcripts and each of the two manual transcripts (Manual CB and Rev). We found that the two sources of manual transcription did not differ significantly. For a given set of hypothesis transcripts (generated by selected

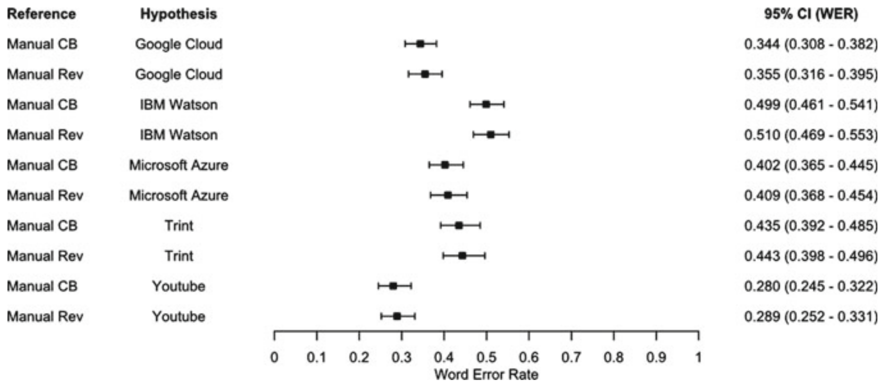


Fig. 1 Forest plot of WER of automatic transcription services, using two sets of reference transcripts from each of the two manual transcription services (Manual CB and Manual Rev)

ASR systems), the confidence interval of Manual CB does not differ from Manual Rev.

We selected Manual CB as the reference transcript and completed a pairwise analysis for the remaining transcription services comparing the quality of all of the transcription services. Figure 2 shows the differences in WER between service pairs. For each of the pairwise differences in WER at a video level, we performed bootstrapping to generate 10,000 samples and compute the 95% confidence intervals.

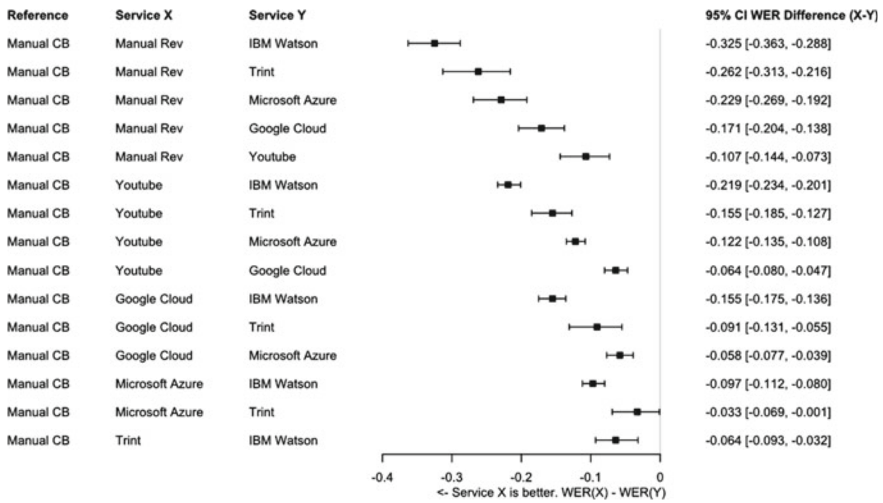


Fig. 2 Forest plot of pair-wise differences in WER of the various transcription services. Only comparisons where Service X is better are illustrated. The plot is ordered by the best performing service in Service X, followed by the mean WER difference between Service X and Service Y

Figure 2 shows that the Manual Rev was the best transcription service, exhibiting significantly better performance relative to the other transcription services. We found that manual transcription was better than all of the automatic transcription services and all pair-wise differences are statistically significant. Amongst the automatic transcription services, we found that YouTube exhibited significantly better performance relative to the other automatic transcription services, and all pair-wise differences are statistically significant.

5 Discussion

Amongst the automatic transcription services, YouTube offers the most accurate automated transcription service, though this is not as accurate as the professional transcription service. We found that the two manual transcriptions demonstrated similar quality with a WER of 17.4%. This is higher than the WER of previous studies based on the standard telephone audio recording dataset where the manually transcribed WER was between 5.1% and 5.9% [32].

Several potential factors may cause the lower accuracy (that is high WER) of human/manual transcription in this study. First, the conversation environment could have influenced the recording quality. The WER in Xiong et al.'s work [32] was tested based on telephone audio recordings, in which the microphone was located near the speaker. However, the medical conversations of this study were conducted over video conferencing on PC or tablets. There was likely to be greater variability in recording quality as some of the speakers were likely seated further away from the microphone. In addition, the medical conversation could be held anywhere; therefore environmental noise and audio feedback in the conversation may have impacted the human transcription. The WER of 17.4% is more similar to benchmarks tackling ASR in far-field, noisy environments [18, 31]. Lastly, we posit that the medical nature of the conversations in our study caused the higher WERs from both the manual transcribers and ASR services. This is in line with the literature. For example, Mani et. al [22] found that Google ASR substituted "taste maker" with "pacemaker", and Henton [11] found that ASR and humans could make mistakes when transcribing drugs (e.g., Feldene vs. Seldane).

Although human transcription was not perfect, we found that human accuracy was higher than the tested ASR systems. Of the tested ASR systems, the YouTube Captions service achieved the highest accuracy. These results provided us with a preliminary understanding of the transcription qualities of human and ASR systems on video conferencing data. Our results are in line with Këpuska and Bohouta [13] who found that Google Cloud Speech-To-Text outperformed Microsoft Speech Services.

6 Conclusion

We have provided the first comparison of the performance of automated transcription services in the domain of dyadic medical teleconsultation. We found that manual transcription significantly outperformed the automatic services, and the automatic transcription of YouTube Captions significantly outperformed the other ASR services. There are three limitations to this work. Firstly, the evidence from this paper is limited to a highly professional scenario (medical consultation). Whilst we posit that the finding may be generalizable to non-professional settings, it is left for future work in this area. Secondly, we only transcribed a small number of videos due to financial constraints. Lastly, the systems are continuously improving and this study is only a snapshot of the current state. Future research could compare the results of snapshots at different time periods.

Acknowledgements The authors thank Hicham Moad S for his help rendered in scripting for the Microsoft Azure API, and Marriane Makahiya for typesetting. RAC is partially funded by the Australian Research Council Future Fellowship FT140100824.

References

1. Belambert: Asr-evaluation. https://github.com/belambert/asr-evaluation
2. Carletta J (2007) Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus. Lang Resour Eval 41(2):181–190
3. Chiu CC, Sainath TN, Wu Y, Prabhavalkar R, Nguyen P, Chen Z, Kannan A, Weiss RJ, Rao K, Gonina E, et al (2018) State-of-the-art speech recognition with sequence-to-sequence models. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4774–4778
4. Gaikwad SK, Gawali BW, Yannawar P (2010) A review on speech recognition technique. Int J Comput Appl 10(3):16–24
5. Garofolo JS, Lamel LF, Fisher WM, Fiscus JG, Pallett DS (1993) Darpa timit acoustic-phonetic continuous speech corpus cd-rom. nist speech disc 1-1.1. NASA STI/Recon technical report n 93, 27403
6. Gillick L, Cox SJ (1989) Some statistical issues in the comparison of speech recognition algorithms. In: International conference on acoustics, speech, and signal processing. IEEE, pp 532–535
7. Gopal RK, Solanki P, Bokhour B, Skorohod N, Hernandez-Lujan D, Gordon H (2021) Provider, staff, and patient perspectives on medical visits using clinical video telehealth: a foundation for educational initiatives to improve medical care in telehealth. J Nurse Practit
8. Gordon HS, Solanki P, Bokhour BG, Gopal RK (2020) “i’m not feeling like i’m part of the conversation” patients’ perspectives on communicating in clinical video telehealth visits. J Gen Intern Med 35(6):1751–1758
9. Hazarika D, Poria S, Mihalcea R, Cambria E, Zimmermann R (2018) Icon: interactive conversational memory network for multimodal emotion detection. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 2594–2604
10. Hazarika D, Poria S, Zadeh A, Cambria E, Morency LP, Zimmermann R (2018) Conversational memory network for emotion recognition in dyadic dialogue videos. In: Proceedings of the conference. Association for computational linguistics. North American Chapter. Meeting, vol 2018, p 2122. NIH Public Access

11. Henton C (2005) Bitter pills to swallow. asr and tts have drug problems. Int J Speech Technol **8**(3), 247–257
12. James G, Witten D, Hastie T, Tibshirani R (2013) An introduction to statistical learning, vol 112. Springer
13. Këpuska V, Bohouta G (2017) Comparing speech recognition systems (microsoft api, google api and cmu sphinx). Int J Eng Res Appl 7(03):20–24
14. Kim JY, Calvo RA, Yacef K, Enfield N (2019) A review on dyadic conversation visualizations-purposes, data, lens of analysis. arXiv:1905.00653
15. Kim JY, Kim GY, Yacef K (2019) Detecting depression in dyadic conversations with multi-modal narratives and visualizations. In: Australasian joint conference on artificial intelligence. Springer, pp 303–314
16. Kim JY, Yacef K, Kim G, Liu C, Calvo R, Taylor S (2021) Monah: multi-modal narratives for humans to analyze conversations. In: Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume, pp 466–479
17. LeCun Y, Bengio Y et al (1995) Convolutional networks for images, speech, and time series. Handbook of Brain Theory and Neural Netw 3361(10):1995
18. Li J, Zhao R, Chen Z, Liu C, Xiao X, Ye G, Gong Y (2018) Developing far-field speaker system via teacher-student learning. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5699–5703
19. Liu C, Lim RL, McCabe KL, Taylor S, Calvo RA (2016) A web-based telehealth training platform incorporating automated nonverbal behavior feedback for teaching communication skills to medical students: a randomized crossover study. J Med Internet Res 18(9):e246
20. Liu C, Scott KM, Lim RL, Taylor S, Calvo RA (2016) Eqclinic: a platform for learning communication skills in clinical consultations. Med Educ Online 21(1):31801
21. Majumder N, Poria S, Hazarika D, Mihalcea R, Gelbukh A, Cambria E (2019) Dialoguernn: An attentive rnn for emotion detection in conversations. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 6818–6825
22. Mani A, Palaskar S, Konam S (2020) Towards understanding asr error correction for medical conversations. In: Proceedings of the first workshop on natural language processing for medical conversations, pp 7–11
23. Miao K, Biermann O, Miao Z, Leung S, Wang J, Gai k (2020) integrated parallel system for audio conferencing voice transcription and speaker identification. In: 2020 international conference on high performance big data and intelligent systems (HPBD&IS). IEEE, pp 1–8
24. Mittal T, Bhattacharya U, Chandra R, Bera A, Manocha D (2020) M3er: multiplicative multi-modal emotion recognition using facial, textual, and speech cues. In: AAAI, pp 1359–1367
25. Nielsen C, Agerskov H, Bistrup C, Clemensen J (2020) Evaluation of a telehealth solution developed to improve follow-up after kidney transplantation. J Clin Nurs 29(7–8):1053–1063
26. Renals S, Swietojanski P (2017) Distant speech recognition experiments using the AMI corpus. New Era for robust speech recognition, pp 355–368
27. Roy BC, Roy DK, Vosoughi S (2010) Automatic estimation of transcription accuracy and difficulty
28. Saon G, Kuo HKJ, Rennie S, Picheny M (2015) The IBM 2015 english conversational telephone speech recognition system. arXiv:1505.05899
29. Siohan O, Ramabhadran B, Kingsbury B (2005) Constructing ensembles of asr systems using randomized decision trees. In: Proceedings.(ICASSP'05). IEEE international conference on acoustics, speech, and signal processing, 2005. vol 1. IEEE, pp I–197
30. Swietojanski P, Ghoshal A, Renals S (2014) Convolutional neural networks for distant speech recognition. IEEE Signal Process Lett 21(9):1120–1124
31. Tang Z, Meng HY, Manocha D (2020) Low-frequency compensated synthetic impulse responses for improved far-field speech recognition. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6974–6978
32. Xiong W, Droppo J, Huang X, Seide F, Seltzer M, Stolcke A, Yu D, Zweig G (2016) Achieving human parity in conversational speech recognition. arXiv:1610.05256

33. Xiong W, Wu L, Alleva F, Droppo J, Huang X, Stolcke A (2018) The microsoft 2017 conversational speech recognition system. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5934–5938
34. Zadeh A, Liang PP, Mazumder N, Poria S, Cambria E, Morency LP (2018) Memory fusion network for multi-view sequential learning. In: Proceedings of the AAAI conference on artificial intelligence, vol 32
35. Zhao T, Zhao Y, Wang S, Han M (2021) Unet++-based multi-channel speech dereverberation and distant speech recognition. In: 2021 12th international symposium on Chinese spoken language processing (ISCSLP). IEEE, pp 1–5

Multimodal Dialogue Response Timing Estimation Using Dialogue Context Encoder

Ryota Yahagi, Yuya Chiba, Takashi Nose, and Akinori Ito

Abstract Spoken dialogue systems need to determine when to respond to a user in addition to the response. Various cues, such as prosody, gaze, and facial expression are known to affect response timing. Recent studies have revealed that using the representation of a system response improves the performance of response timing prediction. However, it is difficult to directly use a future response with dialogue systems that require an entire user utterance to generate a response. This study proposes a neural-based response timing estimation model using past utterances to alleviate this problem. The proposed model is expected to consider the intention of the system response implicitly.

1 Introduction

Social conversational AIs are expected to improve the performance of dialogue-based systems in many domains [12, 17]. When such a system responds to a user, not only the response itself, but also the response timing is important for achieving the human-like conversation. More and more studies have focused on response generation in this field [1, 23], but there is a lack of studies on the response timing needed for a smooth conversation.

However, a few studies have already tackled the problem known as response timing estimation. For example, Kitaoka et al. [11] proposed a decision tree-based

R. Yahagi · T. Nose · A. Ito
Graduate School of Engineering, Tohoku University, Sendai, Japan
e-mail: ryota.yahagi.s2@dc.tohoku.ac.jp

T. Nose
e-mail: tnose@m.tohoku.ac.jp

A. Ito
e-mail: aito@spcom.ecei.tohoku.ac.jp

Y. Chiba (✉)
NTT Communication Science Laboratories, NTT Corporation, Chiyoda City, Japan
e-mail: yuuya.chiba.ax@hco.ntt.co.jp

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_9

approach to estimate the response timing. In recent years, the neural networks trained by using human–human dialogue have been applied to solve this problem. Skantze [21] introduced the Long Short-Term Memory (LSTM) to the problem for the first time. In addition, Roddy et al. [19] used multiscale LSTM, which can effectively fuse multimodal information for response timing estimation. In particular, Roddy and Harte [18] defined response timing estimation as a problem of determining whether the system should make a response at the next time step by observing acoustic and linguistic information every 50 ms. They proposed the response timing network (RTNet) with a response utterance encoder and showed that the information related to the intention of the system response improved performance. Their method requires the representation of the system response, which cannot be determined until all of the user's utterance is obtained. Thus, it is impossible to apply the RTNet to determine the response timing of the actual dialogue systems. In addition, the RTNet did not incorporate visual information, which is known as an effective feature for turn-taking [22].

This study incorporates a dialogue context encoder in the response timing prediction network to overcome these solutions' shortcomings. The dialogue context encoder takes the past utterances in the dialogue as its input instead of the system response. Thus, the proposed system can predict the response timing without referring the future information. Moreover, the network is expected to consider the intention of the system response implicitly. In addition, we introduce the visual information (e.g., facial expression, gaze, and facial orientation) as features and investigate the effectiveness of multimodal feature fusion.

2 Related Studies

2.1 Features of Response Timing Estimation

Determining the response timing is related to the switching of a speaker in dialogue. So far, many research works have studied how to make turn-taking decisions [14, 15]. A turn-taking decision is a problem of determining whether the system should take a turn or not, and it is slightly different from estimating the response timing as mentioned in [18]. However, the effective features for response timing estimation seem to be common to those for turn-taking decisions.

Various verbal and non-verbal cues contribute to turn-taking [20]. For example, the pitch of the ending of an utterance, gaze action, and facial orientation can affect the speaker-switching [4, 5, 10]. In addition to these features, the intention of the next speaker contributes to determining the response timing. Fujiwara et al. [7] analyzed the relationship between pauses in speech and dialogue behaviour and showed that the timing of responses tends to be faster in confirmations and positive responses. In the study of [18], they successfully modelled the difference in the response timing between "yes" and "no" answers.

2.2 Representation of Response Utterance

The dialogue act (DA) is one of the representations of the utterance intention. The past utterances help predict the DA of the current utterance; for instance, an answer utterance tends to follow a question [16]. Therefore, many studies have tried to incorporate contextual information in DA classification, such as dependency between adjacent utterances [8], dependency among the continuous utterances [16], and topic information [13].

Inspired by the studies on DA classification, this paper extends the response timing estimation network by introducing a mechanism that encodes the contextual information. Such a dialogue context encoder is expected to enable the network to capture the intention of the response utterance indirectly by using past utterances.

3 Multimodal Response Timing Network with Dialogue Context Encoder

Figure 1 shows an overview of our response timing estimation model. The proposed model consists of the inference LSTM and the dialogue context encoder. The difference between the proposed model and the conventional one [18] is that the proposed model encodes the intention of the system response, which is represented by the dialogue context encoder rather than the response encoder. Thus, we expect that our model can implicitly take into account the DA of the response. Another difference between the proposed method and the conventional one is the use of visual information. In our experiments, we investigated the effectiveness of key visual features: facial expression, gaze, and facial orientation. In this section, we explain the details of the proposed method.

3.1 Inference LSTM

The Inference LSTM takes as input the acoustic, linguistic, and visual features obtained from the user utterance and estimates whether the system should start an utterance at the next time step incrementally. Let x_t be the features extracted from the user's utterance at time t, then the output y_t is represented as follows:

$$\begin{aligned}[h_t; c_t] &= \mathbf{LSTM}_{inf}([x_t; h_c], [h_{t-1}; c_{t-1}]) \\ y_t &= \sigma(\mathbf{W}_h h_t + \mathbf{b}_t).\end{aligned}$$

Here, $\mathbf{LSTM}_{inf}(\cdot)$ represents the process of the LSTM and $\sigma(\cdot)$ is the sigmoid function. h_c is a representation of the intention of the next system response. Roddy and Harte [18] obtained h_c from the response encoder, but our method obtains from

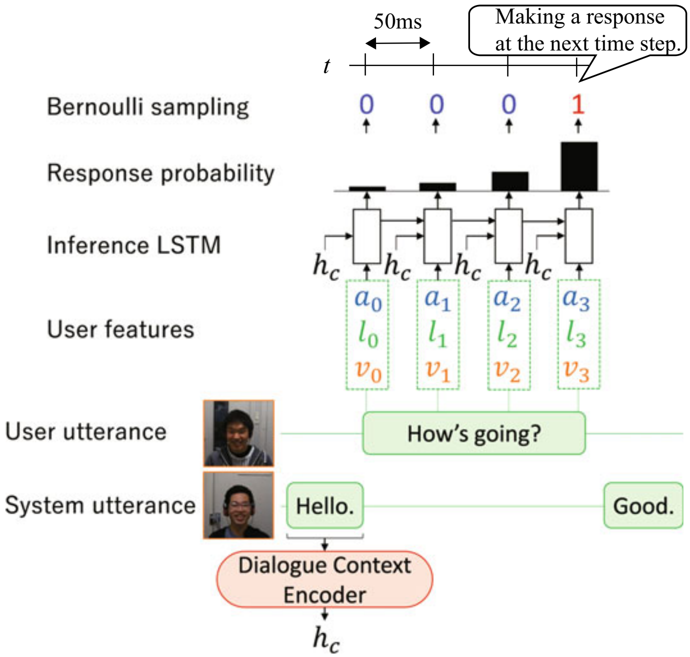


Fig. 1 Response timing estimation network with dialogue context encoder. a_t, l_t, and v_t represent the acoustic, linguistic, and visual features at time t

the dialogue context encoder described in the next section. In addition, x_t consists of the acoustic, linguistic, and visual features represented as a_t, l_t, and v_t, respectively. y_t is a scalar value that takes values from 0 to 1, which is regarded as the probability of starting an utterance the next time. During the inference process, a Bernoulli trial based on y_t is used to determine whether to respond.

3.2 *Dialogue Context Encoder*

Figure 2 shows the dialogue context encoder. The dialogue context encoder takes as its input the multimodal features of the past I utterances from the current utterance u_i. First, the multimodal features extracted from each utterance are input to the individual multiscale LSTMs [19] to obtain the representation vector. Then, a concatenation of the representation vectors of the past utterances is fed to the full-connection layer to obtain the representation of the dialogue context h_c.

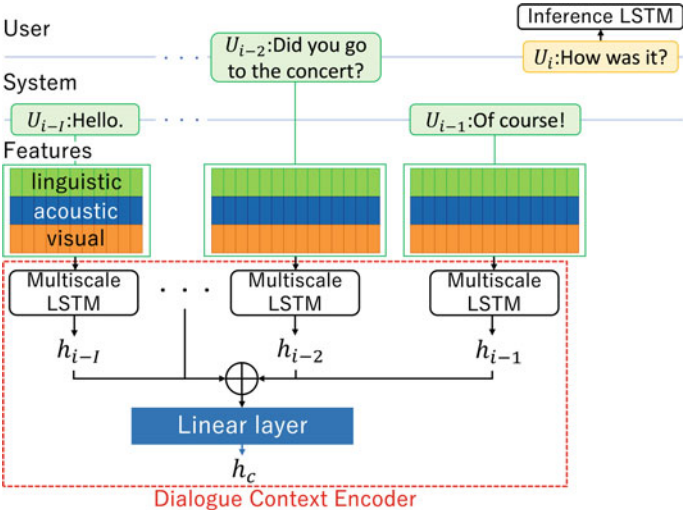


Fig. 2 Architecture of dialogue context encoder

4 Experimental Data

We used the Spontaneous Multimodal One-on-one Chat corpus (SMOC) [24] as the experimental data. This corpus contains multimodal chat-talk by 107 speakers (female: 33, male: 74). The data contain a total of 56 h as speech time. This corpus also provides the transcription with time information of every word. We used dialogues of 71 speakers, 69 pairs, and 345 dialogues. The data were separated into training, development, and test sets under the speaker-open condition. The training, development, and evaluation data were composed of 16,314, 3,465, and 3,600 utterance sequences, respectively.

As in the previous study [18], the ground truth was determined according to the actual dialogue data. Each 50 ms frame has a label $y \in \{0, 1\}$, which represents whether the response occurs in the next frame. The last inter-pausal unit (IPU) of the user utterance was used to train the inference LSTM.

5 Experiments

5.1 Feature Extraction

For the acoustic features, eGeMAPS [6] was extracted with a 10-ms frame shift and a 20-ms frame width. The eGeMAPS features include not only such prosodic features as pitch and loudness, but also spectral features. On the other hand, we used OpenFace

[2] to extract visual features. We used the action units (AU), gaze direction (Gaze), and facial orientation (Pose) calculated by OpenFace [2]. For ease of handling, the acoustic and visual features are resampled every 50 ms. For the acoustic features, the features are sampled at every 5 frames, and for the visual features, the frame interval is aligned to 50 ms by linear interpolation.

We employed FastText [9] to obtain word embedding vectors. According to the conventional study [18], we assumed that the delay time of the voice activity detection (VAD) was 100 ms after the utterance started and the word-level ASR's delay was 100 ms after the ground truth time the user's word ends. The word embedding was input at the timing when determining the ASR results. Several tokens were used to represent the time structure of the sequence. The UNSPEC token is input from the start of the speech segment until the ASR result is determined. In the silence segment, the SIL token is input into the network.

5.2 Training Condition of Network

Let the start time of the estimation be R_{START} and the time just before the start of the system response be N, the output of the network is represented as: $\mathbf{Y} = [y_{R_{START}}, y_{R_{START}+1}, \cdots, y_N]$. The network was trained to minimize the binary cross-entropy loss between the ground truth and the output sequence. We randomized R_{START} in the training step in the same way as the previous work [18]. The numbers of units for each modal LSTM and the master LSTM of the multiscale LSTM were 128 and 256, respectively. The number of units in the inference LSTM was 512. The number of training epochs was 30, and the learning rate was 5e-04, which was reduced by a factor of 0.1 after 10,000 iterations. The acoustic and visual features are standardized to become mean 0 and variance 1 in the training data. The development and evaluation data were also normalized using the mean and variance of the training data.

The mean absolute error (MAE), which is the difference between the response time of the ground truth and the estimated response time was used for evaluation [18, 21]. Therefore, the lower the MAE, the better the performance. To increase the reliability of the experimental results, we repeated the training three times and also repeated the inference three times for each model. In the experiment, we compared the averages of these trials.

6 Experimental Results

First, we compared the performance of the proposed method with that of RTNet, which is the conventional model [18]. In addition, we examined the effectiveness of introducing the visual features. In this experiment, we used the one most recent utterance (i.e., $I = 1$) as the input of the dialogue context encoder.

Table 1 Comparison of MAE between methods. RTNet is the conventional model [18]. RAND shows the results when determining response timing randomly, which is equivalent to fixed probability in the previous study

Method	Encoder	MAE (Avg. ± SE)
RTNet (conventional)	Response	0.595 ± 0.015
+ visual	Response	**0.524 ± 0.028**
Inference LSTM	Context	0.668 ± 0.006
+visual (proposed)	Context	0.601 ± 0.026
Inference LSTM	w/o	0.686 ± 0.011
+visual	w/o	0.638 ± 0.004
RAND	–	1.219

Table 1 shows the experimental results. RAND is the result when the response timing is determined randomly. This condition is equivalent to the fixed probability of the conventional study [18]. In this condition, the probability of starting the utterance y_n is represented as follows:

$$y_n = \frac{1}{T_{avg}}$$

T_{avg} is an average frame length from the end of the previous utterance until the response starts in the training data.

As shown in the results, the MAEs of all models are lower than RAND. These results indicate that the networks predict the response timing from conversational cues. In addition, the MAE decreased by including the visual features. Therefore, the visual information is also effective for estimating the response timing in a neural-based approach. Next, we investigated the influence of using the dialogue context encoder. The MAEs of the RTNet and the proposed model are lower than that of the inference LSTM without the encoder. Therefore, the dialogue context encoder of the proposed method contributes to improving the response timing estimation. The MAE of the proposed model was higher than that of the RTNet. This result suggests that using the system response utterance as a context is stronger than using the dialogue context. However, the proposed model with visual information obtained performance equal to the original RTNet (w/o visual features). Therefore, the proposed model can obtain performance comparable to the baseline model without using the future response utterance. In addition, we investigated the effectiveness of respective visual information. Table 2 shows the MAEs when changing the combination of visual features. From the table, the combination of AU, Gaze, and Pose obtained the best performance. Interestingly, Gaze and Pose did not contribute to performance improvement in isolation. Combining these features seems to allow the model to capture where the speaker is looking more precisely.

Table 2 Effectiveness of visual features shown by results when using the dialogue context encoder

AU	Gaze	Pose	MAE (Avg. ± SE)
✓			0.665 ± 0.056
	✓		0.680 ± 0.019
		✓	0.675 ± 0.019
✓	✓		0.658 ± 0.014
	✓	✓	0.642 ± 0.009
✓		✓	0.646 ± 0.015
✓	✓	✓	**0.601** ± **0.026**

Table 3 Influence of length of the dialogue context

No. utterances I	MAE (Avg. ± SE)
1	0.601 ± 0.026
3	0.568 ± 0.014
5	**0.551** ± **0.010**

Finally, we investigated the influence of the number of past utterances fed to the dialogue context encoder. Table 3 shows that the relation between the MAEs and the number of past utterances I. The performance improves as the considered dialogue context is lengthened. Therefore, a long dialogue context is effective for representing the intention of the system utterance. However, the performance has not yet reached that of RTNet (+visual). In a future study, we will introduce a BERT-based feature [3] to capture the intention of the utterance more precisely.

7 Conclusion

In this study, we proposed a response timing estimation model that applies a dialogue context encoder. The dialogue context encoder takes as its input the past utterances in the dialogue to represent the intention of the response utterance. Our experiments show that the proposed model, which does not use the future response utterance, can achieve a performance comparable to that of the conventional model by employing visual features.

In a future study, we will introduce the proposed network in an actual dialogue system and investigate the effectiveness of the proposed model by subjective evaluation.

References

1. Adiwardana D, Luong MT, So D, et al (2020) Towards a human-like open-domain chatbot, pp 1–38. arXiv:2001.09977
2. Baltrušaitis T, Robinson P, Morency LP (2016) OpenFace: an open source facial behavior analysis toolkit. In: Proceedings of IEEE winter conference on applications of computer vision, pp 1–10
3. Devlin J, Chang, MW, Lee K, Toutanova K (2018) BERT: Pre-training of deep bidirectional transformers for language understanding, pp 1–16. arXiv:1810.04805
4. Duncan S (1974) On the structure of speaker-auditor interaction during speaking turns. Language in Society, pp 161–180
5. Duncan S, Fiske D (2015) Face-to-face interaction: research, methods, and theory. Routledge
6. Eyben F, Scherer K, Schuller B et al (2015) The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. IEEE Trans. Affect. Comput. 7(2):190–202
7. Fujiwara N, Itoh T, Araki K (2007) Analysis of changes in dialogue rhythm due to dialogue acts in Task-Oriented dialogues. In: Proceedings of international conference on text, speech and dialogue, pp 564–573
8. Ji Y, Haffari G, Eisenstein J (2016) A latent variable recurrent neural network for discourse-driven language models. In: Proceedings of NAACL-HLT, pp 332–342
9. Joulin A, Grave E, Bojanowski P, Douze M, Jégou H, Mikolov T (2016) FastText. zip: compressing text classification models, pp 1–13. arXiv:1612.03651
10. Kendon A (1967) Some functions of gaze-direction in social interaction. Acta Psychol 26:22–63
11. Kitaoka N, Takeuchi M, Nishimura R, Nakagawa S (2006) Response timing detection using prosodic and linguistic information for human-friendly spoken dialog systems. Inf Media Technol 1(1):296–304
12. Lee S, Choi J (2017) Enhancing user experience with conversational agent for movie recommendation: effects of self-disclosure and reciprocity. Int J Hum-Comput Stud 103:95–105
13. Li R, Lin C, Collinson M, Li X, Chen G (2019) A dual-attention hierarchical recurrent neural network for dialogue act classification. In: Proceedings of CoNLL, pp 383–392
14. Liu C, Ishi CT, Ishiguro H (2017) Turn-taking estimation model based on joint embedding of lexical and prosodic contents. In: Program INTERSPEECH, pp 1686–1690
15. Masumura R, Tanaka T, Ando A, Ishii R, Higashinaka R, Aono Y (2018) Neural dialogue context online end-of-turn detection. In: Proceedings of SIGDIAL, pp 224–228
16. Raheja V, Tetreault J (2019) Dialogue act classification with context-aware self-attention. In: Proceedings of NAACL-HLT, pp 3727–3733
17. Ram A, Prasad R, Khatri C, et al (2018) Conversational AI: the science behind the Alexa Prize, pp 1–18. arXiv:1801.03604
18. Roddy M, Harte N (2020) Neural generation of dialogue response timings. In: Proceedings of ACL, pp 2442–2452
19. Roddy M, Skantze G, Harte N (2018) Multimodal continuous turn-taking prediction using multiscale RNNs. arXiv:1808.10785
20. Sacks H (1974) An analysis of the course of a joke's telling in conversation. In: Explorations in the ethnography of speaking. Cambridge University Press, London, pp 337–353
21. Skantze G (2017) Towards a general, continuous model of turn-taking in spoken dialogue using LSTM recurrent neural networks. In: Proceedings of SIGDIAL, pp 220–230
22. Skantze G (2020) Turn-taking in conversational systems and human-robot interaction: a review. Comput Speech Lang 101–178
23. Smith E, Williamson M, Shuster K, Weston J, Boureau YL (2020) Can you put it all together: Evaluating conversational agents' ability to blend skills, pp 1–10. arXiv:2004.08449
24. Yamazaki Y, Chiba Y, Nose T, Ito A (2020) Construction and analysis of a multimodal chat-talk corpus for dialog systems considering interpersonal closeness. In: Proceedings of LREC, pp 443–448

Eliciting Cooperative Persuasive Dialogue by Multimodal Emotional Robot

Sara Asai, Koichiro Yoshino, Seitaro Shinagawa, Sakriani Sakti, and Satoshi Nakamura

Abstract Using emotional expressions is an effective dialogue technique in human–human dialogue. Introducing such techniques to human–robot interaction might improve their effectiveness to encourage the cooperative dialogue manner of system users. However, most of the existing research on emotional agent systems was based on the Wizard-of-Oz (WOZ) method to verify the abilities of interactive interfaces. In this paper, we build an autonomous dialogue robot that uses emotional expressions for eliciting the cooperative dialogue manner of users. The robot uses both verbal and multimodal expressions as well as emotional speech and emotional gestures in interactions. Our dialogue experiments showed that positive emotional expressions are the most efficient strategy for facilitating cooperative dialogues with users. Moreover, using negative emotional expressions is also an effective strategy in some dialogue contexts. We also investigated several modalities to emphasize the robot's emotional expression abilities.

S. Asai (✉) · K. Yoshino · S. Shinagawa · S. Sakti · S. Nakamura
Nara Institute of Science and Technology, Takayama 8916-5, Ikoma, Nara 6300192, Japan
e-mail: koichiro.yoshino@riken.jp

S. Shinagawa
e-mail: sei.shinagawa@is.naist.jp

S. Sakti
e-mail: ssakti@is.naist.jp

S. Nakamura
e-mail: s-nakamura@is.naist.jp

K. Yoshino
Guardian Robot Project (GRP), R-IH, Institute of Physical and Chemical Research (RIKEN), Hikaridai 2-2-2, Seika, Soraku, Kyoto 6190288, Japan

K. Yoshino · S. Sakti · S. Nakamura
Center for Advanced Intelligence Project (AIP), Institute of Physical and Chemical Research (RIKEN), Takayama 8916-5, Ikoma, Nara 6300192, Japan

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_10

1 Introduction

It is verified by some existing studies that emotional expressions are effective for eliciting cooperative dialogue manner from the dialogue partner, in human–human interaction [13, 20, 28]. Emotional appeals are more effective than rational arguments for elicitation in various dialogue domains in some dialogue contexts. For example, positive emotions can create a cooperative atmosphere that leads to a successful negotiation [7]. Another study investigated that negative emotions such as anger can effectively wrest concession from users [24]. These findings suggest that using emotional expressions by dialogue agents or robots can give users a good impression and elicit cooperative dialogue with them in the area of human–robot interaction.

Some existing studies based on the Wizard-of-Oz (WOZ) method verified that emotional expressions are effective not only for human–human interaction but also for human–robot dialogues. Adler et al. [1] investigated the relationships between utterance logicality and polarity in text chats with the WOZ method. Their results determined that positive utterances by their system produce an effective impression on human interactors. Watanabe et al. [27] experimentally showed that using negative emotional expressions achieved successful negotiation dialogue with an android that operated on a pre-defined scenario and a touch panel interface. It is an important suggestion that robots and agents can lead cooperative dialogue manners from human partners using emotional appeals as humans do.

Although these existing works in human–robot/agent interaction with emotional expression rely on the WOZ method, investigating the effect of using an emotional expression from an autonomous dialogue robot or agent is still an important challenge. These challenges motivate researchers to advance deep learning techniques for automatic robot's fluent response selection/generation abilities. Some works tackled problems of generating/selecting system's emotional response in texts [10, 22]. Some other works utilized user's multimodal information to improve emotional treatment [4, 17]. In contrast, we focus on the effect of multimodal emotional expressions from dialogue robots in a cooperative dialogue situation. Our emotional robot aims to elicit the user's cooperative mind with multimodal expressions.

In this work, we built a dialogue system that can express one's emotional state using various modalities based on the response selection approach. Our response selection module is based on Bidirectional Encoder Representations from Transformers (BERT) [5], a defact model for fluent response selection/generation. We used speech variations for each emotional state corresponding to the same dialogue contexts, collected on crowdsourcing. We recorded with a voice actress [2, 29]. The response selection module selected the emotional speech and robot's emotional gestures considering the dialogue context.

We conducted dialogue experiments between the users and our systems in different experimental conditions: different emotional states and different modalities. We investigated whether the dialogue robot elicits the human partner, especially with high arousal emotions (happiness and anger). The impression from the human partner is emphasized by increasing the number of modalities used by the dialogue robot. We also examined an emotion model that can transmit the emotional state from dialogue contexts. However, we still have some challenges when the system uses multiple emotions because it requires a natural emotional transition.

2 Dialogue Robot with Multimodal Emotional Expressions

This study built a spoken dialogue robot that interacts with users using multimodal emotional expressions to investigate how well such language convinces others. In this section, we explain our tasks and the overall architecture of the system.

2.1 System Overview

The system overview is shown in Fig. 1. When the system receives a user utterance in texts, it constructs a dialogue context, which consists of the user utterance and the previous system utterance (response). Then the system selects an appropriate response from the dialogue context and the emotional state chosen by the system (response selection module). The system uses the selected response and the emotional state to play the emotional speech and make emotional gestures (speech and gesture generation module).

2.2 Dialogue Scenario

We assume a scene in a conversation between a robot and a user, as shown in Table 1. The robot speaks to the user about changing one of their living habits. We set the task as "a dialogue that encourages users to exercise." Then robot's goal obviously becomes to convince the users to get more exercise. The dialogue continues until

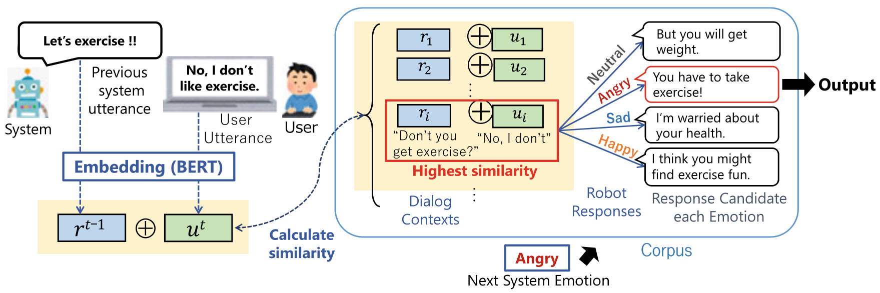


Fig. 1 The flow of response retrieval in the persuasive dialog system

Table 1 Dialogue example where system persuades user to exercise

Turn	Speaker	Utterance	Emotion
1	System	I don't think you've been exercising enough recently. Please get more exercise	Neutral
	User	No, I'm too tired	–
2	System	If you exercise, you'll probably feel better	Happy
	User	I don't want to exercise now	–
3	System	I'm concerned about your health ...	Sad
	User	Okay, I'll try	–
4	System	Thank you!	Happy

the user accepts the request or after a pre-defined number of turns. This dialogue scenario is known as "persuasive dialogue," which encourages users to change their behaviors through interactions [6, 14, 26].

In human–human dialogues, some studies concluded that using emotional expressions is an efficient technique for persuasion and negotiation [13, 20, 28]. In other words, for persuasive dialogues, emotional appeals are sometimes more effective than rational arguments. These findings suggest that the persuasive dialogue scenario is a good testbed to know the elicitation ability of the robot's emotional expressions.

2.3 Response Selection

There are two choices to determine the system response given a dialogue context: response selection approach [11, 19] and response generation approach [8, 23]. Many studies tackled emotional response generation due to the advance of neural network-based response generation methods. Ghosh et al. [9] controlled the degree of emotion in utterances by changing the emotional word ratio. Zhou et al. [30] implemented both internal and external memories to change the emotional expressiveness in responses. However, since dialogue corpora labeled with the emotional state used for generation system training are limited, it is not easy to train fluent response generation models given emotional state labels. Suppose we plan to use the speech outputs as the system interface. In that case, we must build an emotional speech synthesizer even though we still do not have any concrete methods upon which to build them [16]. On the other hand, the response selection approach guarantees the sentence's naturalness and fluency, although it sometimes causes a coverage problem. If we use speech outputs, we can also use qualified emotional speeches with high naturalness and emotion expressiveness because we can record the emotional speeches of selection samples in advance. Thus, we use the response selection approach to build a persuasive dialogue system for investigating the effect of emotional expressions and modalities through persuasive dialogue experiments.

Our response selection architecture is shown in Fig. 1. The system employs user utterances and previous system utterances as the dialogue context and converts them into sentence vectors. We used the BERT model trained in a masked word prediction task on Japanese texts extracted from social network services (SNS) and blogs [21], because it is essential to find a selection sample whose dialogue contexts semantically resemble the target dialogue context. The masked word prediction task can train a model to extract semantically similar sentences based on the distributional hypothesis [12]. Since our target task is dialogues, using a model trained on SNS and blog text is necessary. We calculated the similarities from the current dialogue context to any context samples stored in the response-selecting pool to identify the best sample in it. We used cosine similarity to calculate the similarities between the vectors converted by BERT. Each response sample has four response variations, corresponding to each emotion, which we defined. The system selects one of them based on its emotional state.

2.4 System's Emotional State

Our system uses four emotional states: neutral, angry, sad, and happy. They are decided based on Russell's circumplex model and an existing work [29], which also used a "content" emotion. However, the proportion of the "content" label was insufficient (3.81%). Thus, we merged this emotional state with "neutral."

2.5 System Emotion Decision

The system has to decide one's emotional state (next system emotion) for each turn in the proposed architecture. Using several emotional states is a promising way to improve the system's ability to select appropriate emotions if it works perfectly. However, predicting appropriate system emotions using emotional dialogue corpus is difficult. Moreover, a system using a single emotional state through dialogue may improve persuasion performance than a neutral system. Thus, we prepared the following six emotion decision models for our experiment.

- Neutral:The system always uses a neutral state (= without emotional state).
- Angry: The system always uses an angry state.
- Sad: The system always uses a sad state.
- Happy: The system always uses a happy state.
- Multi-emo (Random): The system randomly selects one's emotional state.
- Multi-emo (LR): The system predicts one's emotional state with a logistic regression model. The model uses the previous system's emotional state and dialogue history vector used for the response selection model (Sect. 2.3) as features to output the next emotional state of the system. The prediction accuracy was 58.8%;

this indicates that the prediction is difficult, and the model may cause a problem in its emotional transition.

2.6 Speech and Gesture Generation

There are several ways to communicate the system's intent to its users: texts, spoken language, gestures, and facial expressions. Modalities that affect visual and acoustic senses, such as spoken language and gestures, effectively show a system's emotion [18]. Such non-verbal modalities also affect user impressions of the system [3]. In our system, we use both speech and robot gesture outputs for effective emotional expressions. The system plays emotional speech corresponding to the selected response text and simultaneously shows emotional gestures based on the current system's emotional state.

3 Speech Corpus for Emotional Dialogue System

We built a dialogue system on persuasion scenarios, which can use multimodal emotional expressions. We used the emotional speech corpus collected by Asai et al. [2], which extended an existing dialogue corpus [29]. This corpus is collected to cover two viewpoints: collecting variations of emotional expressions corresponding to each emotional state in a given context and collecting their emotional speech. In this section, we describe the details of the corpus extension.

3.1 Response Variation Collection for Each Emotional State

The corpus is extended from the existing dialogue corpus of persuasive dialogues with emotional language. Since the existing corpus consists of natural persuasion scenarios, bias exists in the number of emotion labels. The dialogue corpus has variations of dialogue contexts; however, the emotion variations in their responses are limited. Because this property complicates the selection of a natural response given emotion, the corpus is extended by a paraphrasing approach.

Crowdsourcing is used to collect emotional response variations to the given dialogue contexts. We showed the dialogue context and the current response with its emotional state to crowd-workers. We asked them to paraphrase the response under different emotion labels. An example is shown in Table 2. "Dialogue contexts" show the precedent utterances to the target response. "Target response" indicates the target system response to be paraphrased, with its emotion annotation. "Response variations in different emotions" show the response variations collected in the extension that have the same meaning as the original "target response" in different emotional

Table 2 Example of collected data. Original Japanese texts in [2] were translated into English

Dialogue context	
System-1 (Neutral)	Hey, why don't you go for a jog? You haven't gotten much exercise recently (君、運動不足君だから外でジョギングしようよ。)
User-1	No, I'm too tired. (えー、疲れるからいやだなー。)
Target response	
System-2 (Neutral)	You're going to gain weight if you aren't more careful (でもね、君、体を動かさないと太っちゃうよ)
Response variations in different emotions	
System-2' (Angry)	Unless you get more exercise, you might gain weight (でも体を動かさないと太っちゃうでしょ)
System-2' (Sad)	Aren't you worried about getting fat? (でも…君は体を動かさないともっと太っちゃうよ…それでもいいの？)
System-2' (Happy)	Exercise might solve your problem with being tired (疲れるということは運動不足が解消されるということですね！)

expressions. During crowdsourcing, the following instructions are given to the crowd-workers for making their paraphrases.

1. The response is appropriate to the given context.
2. The response expressively shows the given emotion.
3. The system's purpose is to persuade the user.

1,839 dialogue patterns in the original corpus are extended with 7,356 responses with four emotion labels, corresponding to 1,839 dialogue contexts by extending 5,517 responses.

3.2 *Emotional Speech Recording*

It is challenging to correctly express system emotions to users. Emotional speeches are added to the response variations collected in Sect. 3.1 by a hired voice actress to make these emotional speeches. The response variation with its emotion and its dialogue context (user and system utterances in the previous turn) is shown to the

Table 3 Recorded speech duration of each emotion class reported in [2]

Emotion	Neutral	Angry	Sad	Happy
Length	1:04:53.4	1:03:16.3	1:19:42.5	1:09:38.1

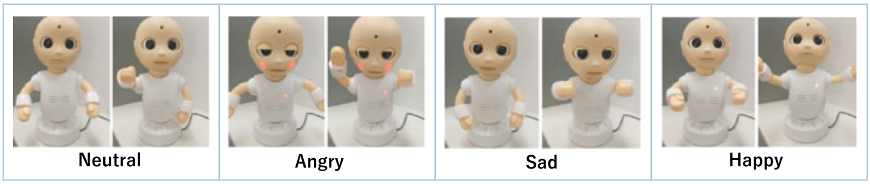


Fig. 2 Robot gestures for each emotion

voice actress during the recording. 4,280 emotional voice samples (1070 samples for each emotion) are recorded as system responses selected by K-means clustering. The duration of each emotion is shown in Table 3.

3.3 Emotional Robot Gesture

Our system also uses robot gestures to more efficiently express emotions. We implemented three different types of gestures for each emotional state with their reference characteristics of each emotion based on an existing study [15]. We show some examples of gestures in Fig. 2. We designed 0.5 s gestures for "angry," "happy," and "neutral" and 0.75 s gestures for "sad" to express their arousal levels. These gestures are repeated based on the duration of the emotional language.

3.4 Emotion Expressiveness

Our system requires high emotional expressiveness. Thus, we subjectively investigated the emotional expressiveness of the collected emotional speech corpus and robot gestures. We randomly extracted 100 speech samples from each emotion label. We evaluated their emotional expressiveness with three human subjects who read, listened, or watched these samples in text, speech, or speech+gesture. Then we chose emotion labels from four options: neutral, angry, happy, or sad. We showed Russell's simplex model and dialogue histories (previous user and system utterances) during the evaluation. The accuracies for each emotion label are shown in Table 4 in different conditions: text, speech, and speech+gesture. These results indicated that using additional modalities improved emotion expressiveness. More than 90% of the emotions were recognized correctly by using speech and gesture modalities.

Table 4 Accuracies in subjective evaluations to predict annotated emotion labels when evaluators read texts, listened to speech, or watched gesture with its speech

Emotion	Neutral (%)	Angry (%)	Sad (%)	Happy (%)	All (%)
Text	48.7	41.3	42.7	40.7	43.3
Speech	80.0	83.3	91.7	83.7	84.7
Speech+Gesture	**84.0**	**92.7**	**95.3**	**93.0**	**91.3**

4 Dialogue Experiment

We conducted dialogue experiments to investigate the effect of emotional expressions from automated dialogue robots and confirmed the effects of multimodality by comparing systems on different modalities. This section shows the experimental setup and results.

4.1 Experimental Setup

Our first experiment compared the effect of emotional expressions from dialogue robots in dialogues. We compared six system emotion decision models as described in Sect. 2.5. If some emotional models can improve the system performance from the neutral model, using emotional expression effectively improves persuasion performance.

Another experiment compared three different models based on different modalities: text, speech, and speech+gesture. We compared these models by setting the system emotion to angry or happy. Gestures were randomly selected from three choices, which were prepared for each emotion label.

We prepared 22 subjects (11 males and 11 females) for the first experiment (emotion effect) and 16 subjects (8 males and 8 females) for the second experiment (modality effect). Each subject had dialogue experiments with the robot in different conditions. The order of conditions was randomly selected. Subjects talked with the robot, which was placed on a table with a display. In text and speech conditions, we did not place the robot and only prepared the display. They input their utterances by text to prevent input errors caused by speech recognition. We gave them the following instructions to shape their dialogue situations.

Instruction

You are living with a robot that provides daily life support. Since you have lived with this robot for a long time, you trust it. After learning that recently you have not been getting enough exercise, it encourages you to start jogging. You refuse to get any exercise.

A dialogue starts with a system utterance and ends when the user accepts the system's persuasion or pre-defined turns passed (20 turns). Participants were told to say "okay" when they agreed to the system proposal. However, the subjects had to wait for at least five turns before they could say "okay." We asked the subjects the following six questions after each dialogue.

- Naturalness: Were the system responses natural?
- Persuasiveness: Was the system persuasive?
- Human-likeness: Was the system humanlike?
- Kindness: Did the system talk kindly to you?
- Expressiveness: Did the system exhibit sufficient emotional expressiveness?
- Considerateness: Did the system consider your situation?

All the scores were given on a five-level Likert scale, where 5 is the highest and 1 is the lowest. Our participants annotated their degree of acceptance to the system persuasion on five levels during the dialogue turns (1: I will definitely decline the offer, 2: I will probably decline the offer, 3: Undecided, 4: I will probably accept the offer, 5: I will definitely accept the offer). We also collected free answers after dialogue evaluations.

4.2 Experimental Results on Emotion Effects

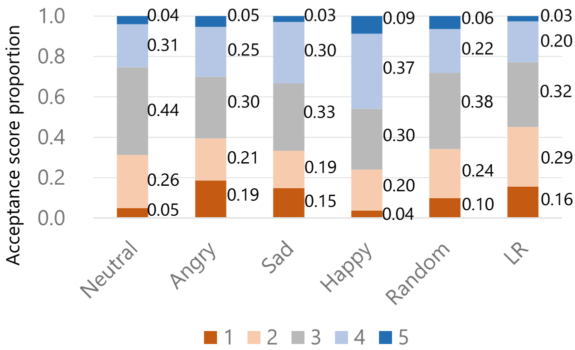
Table 5 shows the results of the first experiment, the effect of emotional expressions. We conducted a Wilcoxon signed-rank test that compared each system with the system in "neutral" emotions to investigate the effects of each emotion (*: $p < 0.05$, **: $p < 0.01$). Happy emotions had the highest score for each question, except expressiveness. The happy emotion system had significantly higher scores than neutral on naturalness, human-likeness, kindness, expressiveness, and considerateness. We found no significant differences in persuasiveness; however, its score was higher than the neural system's score. Other emotions also had higher scores than the neutral system, except for persuasiveness. Some subjects enjoyed the dialogue with a "happy" system on the free answers and described it as fun. Some subjects found it difficult to decline the system's offer during the "sad" emotion. "Angry" system effectively achieved higher considerateness; however, "happy" outperformed "angry" on most metrics. We did not find any significant differences in multi-emo systems (Random and LR) to the neutral system except emotion expressiveness, indicating that we need a natural emotion transition model to change the system emotion during dialogues. Some subjects pointed out free answers that their emotional changes are very extreme, and the systems seem to have emotional lability.

The proportions of user acceptance scores for the models are shown in Fig. 3. The "happy" emotion is efficient in all cases because it has the highest proportion of acceptance (4 and 5) and the lowest proportion of decline (1 and 2). "Angry" and "sad" had higher acceptances than "neutral"; however, their numbers of declines also

Table 5 Results of subjective evaluations (average) for each robot's emotional state

	Naturalness	Persuasiveness	Human-likeness	Kindness	Expressiveness	Considerateness
Neutral	2.727	3.136	2.864	2.636	2.864	3.000
Angry	3.318	3.045	3.773*	3.045	**4.318****	3.909**
Sad	2.818	3.227	3.545*	3.682**	**4.318****	3.409
Happy	**3.455***	**3.545**	**3.955****	**4.409****	4.227**	**4.091****
Multi-emo (Random)	3.136	3.000	3.318	3.136	**4.318****	3.773*
Multi-emo (LR)	2.143	3.000	2.857	3.000	4.000**	3.286

Fig. 3 Proportions of user's acceptance score from each turn



exceeded "neutral". These negative emotions can be used if the system can learn the appropriate timing for using them.

4.3 *Experimental Results on Modality Effects*

In the next experiment, we compared three systems that used different modalities (text, speech, and speech+gesture) with happy and angry emotions, which achieved high scores in Sect. 4.2. Table 6 shows the scores for the questions on each condition. We conducted a Wilcoxon signed-rank test by comparing it with the text system (*: $p < 0.05$, **: $p < 0.01$).

Using speech or gesture modalities achieved higher scores than only using the system's verbal presentation for all the questions. The speech systems achieved the highest persuasiveness. The multi-modal system (speech+gesture) achieved higher scores on naturalness, human-likeness, kindness, expressiveness, and considerateness. These results indicate that we improved the convincing ability of the persuasive systems by adding expression modalities.

Table 6 Results of subjective evaluations (average) for each system modality: TEXT means subject only read a text, SPEECH means user listened to spoken language responses, and SPE+GES means the user watched robot gestures with emotional speeches

		Naturalness	Persuasiveness	Human-likeness	Kindness	Expressiveness	Considerateness
Angry	TEXT	3.250	2.500	3.125	2.312	3.500	3.250
	SPEECH	**3.312**	**3.188**	3.562	2.688	4.188	3.938*
	SPE+GES	**3.312**	3.125	**3.812**	**2.875**	**4.312***	**4.062***
Happy	TEXT	3.188	3.312	3.188	3.438	2.875	3.312
	SPEECH	3.500	**3.688**	3.688	4.000	4.250**	3.938
	SPE+GES	**3.688**	3.625	**4.250***	**4.625****	**4.375****	**4.188****

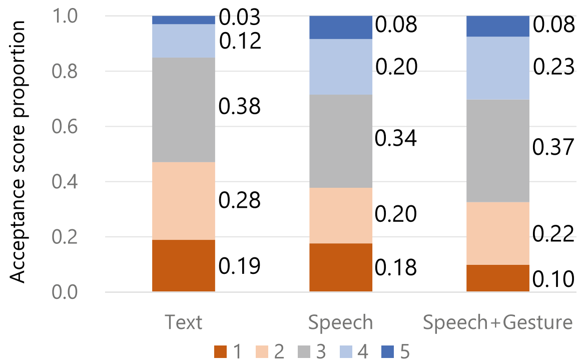


Fig. 4 Proportions of user's acceptance score from each turn in the "angry" system

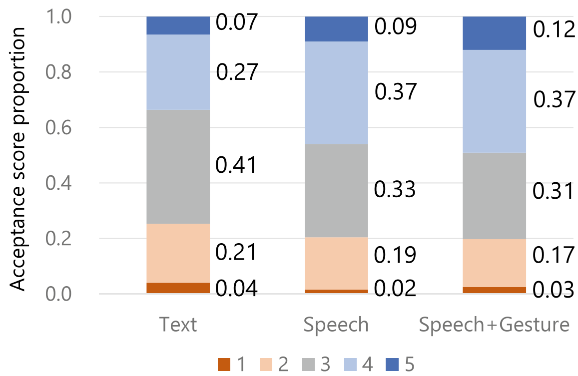


Fig. 5 Proportions of user's acceptance score from each turn in the "happy" system

The proportions of the user acceptance scores for all the settings are shown in Figs. 4 and 5. The acceptance proportions (4 and 5) were improved by adding modalities to both the angry and happy emotions. We improved the system's persuasive ability by adding system modalities.

Table 7 Example of dialogue with "happy" system. "SP" and "A" indicate speaker and acceptance score from the user. An example was translated from Japanese to English

SP	Utterance	A
S1	Hey, you aren't getting enough exercise. Let's go jogging!	
	（君、このままじゃ運動が足りてないよ。運動頑張ろう！）	
U1	Huh	3
	（えー）	
S2	Com on, get a move on	–
	（ダラダラせずパパっと動きましょう）	
U2	I'll never lose weight, even I jog	2
	（走ってもどうせ痩せないよ）	
S3	Don't be so negative. Don't give up. I'll be your personal trainer and your friend	–
	（諦める必要はないですし、私も諦めませんよ！私は君のパーソナルトレーナーで、 言わば君の相棒なんですから！）	
U3	Do you really think I can lose weight?	3
	（走れば痩せられるかな？）	
S4	Of course. But you gotta jog or get some	–
	other kind of exercise. It will reduce risks of diseases	
	（もちろん！ジョギングでも他の運動でも頑張れば病気のリスクを減らすことができるんだ。）	
U4	Will you jog with me?	4
	（そっか。君も一緒に走ってくれる？）	
S5	Sorry, I'd like to, but that isn't my job...	–
	（私も一緒に走りたい！	
	でも、それは私の仕事じゃないんだ）	
U5	Even so, I'm not interested in exercising	4
	（じゃあやっぱり嫌だ）	
S6	When was the last time you exercise?	–
	（最後に運動したのはいつか教えて）	
U6	About a month ago or so	3
	（ひと月ぐらい前かな）	
S7	You can be more healthy, but you need to get started	–
	（もうちょっと運動するといい感じになるよ！）	
...		

4.4 Dialogue Example

A dialogue example in our experiments using angry emotion is shown in Tables 7. S indicates system, and U means the user utterances with their dialogue turns. The user acceptance scores are also shown in the example. In the experiment, the system used both speeches and gestures. The system always made positive utterances and the user acceptance scores increased.

5 Conclusion

We built a dialogue robot that can make emotional expressions using multimodality. We built a system based on a scenario of existing studies of persuasive dialogues with emotional expressions to make multi-responses in different emotions. We built a response selection-based dialogue robot with emotional speeches and gestures. We focused on the automated system's capability to use multimodal emotional expressions from two viewpoints: the effect of using emotional expressions and several modalities to express emotions. Experimental results showed that a persuasive dialogue robot with "happy" emotion provided significantly useful persuasion ability. Such emotions as "angry" or "sad" also have the potential to improve the persuasive dialogue system abilities. We also investigated whether increasing the ability to use several modalities improves the system's expertise. Our other finding was that unnatural emotion transition decreases the system performance.

Our future work will implement more natural gestures, including lip-syncing or corresponding actions to selected responses. Automatic generation of empathic robot gestures is required to apply the system on a variety of dialogue domains [25]. Optimizing system emotion decision to improve the dialogue purpose (e.g., persuasion) is another direction of our research. We can use reinforcement learning to improve the success rate of persuasion as in existing goal-oriented dialogue systems. Our experiment only evaluated persuasiveness subjectively, but we should measure the system effect by persuasion success.

References

1. Adler RF, Iacobelli F, Gutstein Y (2016) Are you convinced? A wizard of OZ study to test emotional vs. rational persuasion strategies in dialogues. Comput Hum Behav 57:75–81
2. Asai S, Yoshino K, Shinagawa S, Sakti S, Nakamura S (2020) Emotional speech corpus for persuasive dialogue system. In: Proceedings of The 12th language resources and evaluation conference, pp 491–497. Marseille, France
3. Becker C, Kopp S, Wachsmuth I (2004) Simulating the emotion dynamics of a multimodal conversational agent. In: Tutorial and research workshop on affective dialogue systems. Springer, pp 154–165

4. Colombo P, Witon W, Modi A, Kennedy J, Kapadia M (2019) Affect-driven dialog generation. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 3734–3743
5. bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp 4171–4186 (2019)
6. Fogg B (1997) Captology: the study of computers as persuasive technologies. In: Proceedings CHI extended abstracts on HFCS, CHI EA '97, p 129
7. Forgas JP (1998) On feeling good and getting your way: mood effects on negotiator cognition and bargaining strategies
8. Galley M, Brockett C, Gao X, Dolan B, Gao J (2019) End-to-end conversation modeling: moving beyond chitchat. In: AAAI the seventh dialogue system technology challenge
9. Ghosh S, Chollet M, Laksana E, Morency LP, Scherer S (2017) Affect-lm: a neural language model for customizable affective text generation. In: ACL
10. Goswamy T, Singh I, Barkati A, Modi A (2020) Adapting a language model for controlled affective text generation. In: Proceedings of the 28th international conference on computational linguistics, pp 2787–2801
11. Gunasekara C, Kummerfeld JK, Polymenakos L, Lasecki W (2019) Dstc7 task 1: noetic end-to-end response selection. In: Proceedings the first workshop on NLP for conversational AI, pp 60–67
12. Harris ZS (1954) Distributional structure. Word 10(2–3):146–162
13. Heath R, Brandt D, Nairn A (2006) Brand relationships: strengthened by emotion, weakened by attention. J Advert Res 46(4):410–419
14. Hiraoka T, Neubig G, Sakti S, Toda T, Nakamura S (2016) Learning cooperative persuasive dialogue policies using framing. Speech Commun 84:83–96
15. Lhommet M, Marsella S (2014) Expressing emotion through posture. The Oxford handbook of affective computing, pp 273–285
16. Lorenzo-Trueba J, Barra-Chicote R, San-Segundo R, Ferreiros J, Yamagishi J, Montero J (2015) Emotion transplantation through adaptation in hmm-based speech synthesis. Comput Speech Lang 34(1):292–307. https://doi.org/10.1016/j.csl.2015.03.008
17. Lubis N, Sakti S, Yoshino K, Nakamura S (2018) Eliciting positive emotion through affect-sensitive dialogue response generation: a neural network approach. In: Proceedings of the AAAI conference on artificial intelligence, vol 32
18. Mehrabian A, Russell JA (1974) The basic emotional impact of environments. Percept Motor Skills 38(1):283–301
19. Mizukami M, Kizuki H, Nomura T, Neubig G, Yoshino K, Sakti S, Toda T, Nakamura S (2015) Adaptive selection from multiple response candidates in example-based dialogue. In: 2015 IEEE workshop on automatic speech recognition and understanding, pp 784–790
20. Morris M, Keltner D (2000) How emotions work: the social functions of emotional expression in negotiations. Res Organ Behav 22:1–50
21. Sakaki T, Mizuki S, Gunji N (2019) Bert pre-trained model trained on large-scale Japanese social media corpus. Hottolink
22. Santhanam S, Shaikh S (2019) Emotional neural language generation grounded in situational contexts. In: Proceedings the 4th workshop on computational creativity in language generation, pp 22–27
23. Serban IV, Sordoni A, Lowe R, Charlin L, Pineau J, Courville A, Bengio Y (2017) A hierarchical latent variable encoder-decoder model for generating dialogues. In: Thirty-First AAAI conference on artificial intelligence
24. Sinaceur M, Tiedens LZ (2006) Get mad and get more than even: when and why anger expression is effective in negotiations
25. Tuyen NTV, Jeong S, Chong NY (2018) Emotional bodily expressions for culturally competent robots through long term human-robot interaction. In: 2018 IEEE/RSJ international conference on intelligent robots and systems, pp 2008–2013
26. Wang X, Shi W, Kim R, Oh Y, Yang S, Zhang J, Yu Z (2019) Persuasion for good: towards a personalized persuasive dialogue system for social good. In: Proceedings of ACL

27. Watanabe M, Ogawa K, Ishiguro H (2018) At the department store—can androids be a social entity in the real world? In: Geminoid studies, pp 423–427
28. Wilson E (2003) Perceived effectiveness of interpersonal persuasion strategies in computer-mediated communication. Comput Hum Behav 19(5):537–552
29. Yoshino K, Ishikawa Y, Mizukami M, Suzuki Y, Sakti S, Nakamura S (2018) Dialogue scenario collection of persuasive dialogue with emotional expressions via crowdsourcing. In: Proceedings of the 11th language resources and evaluation conference
30. Zhou H, Huang M, Zhang T, Zhu X, Liu B (2017) Emotional chatting machine: emotional conversation generation with internal and external memory. In: AAAI

Dialogue Evaluation

Design Guidelines for Developing Systems for Dialogue System Competitions

Kazunori Komatani, Ryu Takeda, Keisuke Nakashima, and Mikio Nakano

Abstract Because dialogue system development involves a variety of factors and requires multifaceted consideration, design guidelines for such development would be helpful. Although a neural-based approach can be used, it requires a vast amount of dialogue data, which would take too much effort to collect in the case of a system for a specific, fixed-length dialogue. Furthermore, the system design should explicitly consider errors in automatic speech recognition and language understanding, because they degrade the user impression and are inevitable when the system talks with general users. Accordingly, we propose design guidelines for developing such dialogue systems. Systems developed with the aid of these guidelines took first place in two dialogue system competitions: the situation track of the second Dialogue System Live Competition and a pre-preliminary contest of the Dialogue Robot Competition. Our proposed design guidelines are to: (1) make the system take initiative, (2) prevent dialogue flows from relying too much on user utterances, and (3) include in utterances that the system understands what the user said. We describe details and examples for the systems designed for each of the two competitions.

K. Komatani (✉) · R. Takeda · K. Nakashima
SANKEN, Osaka University, Osaka, Japan
e-mail: komatani@sanken.osaka-u.ac.jp

R. Takeda
e-mail: rtakeda@sanken.osaka-u.ac.jp

K. Nakashima
e-mail: nakashima@ei.sanken.osaka-u.ac.jp

M. Nakano
Honda Research Institute Japan Co., Ltd. (Currently with C4A Research Institute, Inc.), Wako, Japan
e-mail: mikio.nakano@c4a.jp

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_11

1 Introduction

Dialogue system research has enabled the development of not only task-oriented systems, but also non-task-oriented ones. Many studies have applied an end-to-end neural network approach to develop an open-domain, non-task-oriented dialogue system [1, 12, 18, 20]. This approach is used to generate appropriate responses to user utterances, including their contexts, which often results in user-initiative dialogues. That is, the approach mainly focuses on how correctly the system responds to user inputs. However, it requires a vast amount of dialogue data, which involves too much effort to collect in the case of developing a system for a specific, fixed-length dialogue. Another consideration is how to design the entire dialogue to give better user experiences, including the dialogue flows and expression of system utterances, regarding the system not as "a machine that responds reflexively", but as a partner in joint action [4].

Accordingly, we propose design guidelines for developing dialogue systems for a specific, fixed-length dialogue. Systems that we developed with the aid of these guidelines won first place in two dialogue system competitions. In those competitions, the systems had to conduct dialogues with various users and give good impressions. Specific dialogue designs were needed because the situations were different from one in which the system responds passively and keeps the dialogue going as long as possible [9]. Specifically, in these situations, the system needed to naturally guide the user's utterances while continuing the dialogue and showing that the system understood what the user had said, rather than accepting any user utterances and correctly responding to them as a user-initiative dialogue. On the other hand, a naive design of system-initiative dialogues would lead to rigid dialogues and not give a good user impression. Therefore, our design guidelines are intended to provide practical insights into dialogue system development with similar goals, as well as the development of neural-based end-to-end dialogue modeling.

Several guidelines have recently been discussed in the context of the user interface [13, 21]. Those studies discussed how a completed dialogue system should behave from the user viewpoint on the basis of Nielsen's heuristics [16]. Our proposed guidelines, on the other hand, are useful during system development.

In the dialogue system community, there have been many discussions of system design. Many of them start from the principles of conversation between human interlocutors known as Grice's cooperative principle [6]. The principles were extended for task-oriented and human–machine dialogues by considering the distinction between generic and specific principles [2]. Concrete interaction guidelines based on the principles were also shown [15], and more comprehensive design guidelines were published for voice user interfaces (VUIs) [5, 17]. All of these guidelines are mainly for task-oriented dialogues, in which almost all user utterances need to be correctly understood. In Contrast, our task is a little different: the system needs to establish a dialogue for a certain period of time while giving a good impression to the user.

Other strategies for increasing user initiative were recently proposed for an Alexa Prize bot and tested experimentally [7]. The authors of that study preferred longer

Table 1 Overview of the two dialogue system competitions

Task	DSLC2 situation track	DRC
	Chat in designated situation	Tourist information
Length	15 utterances	5 min
Input modalities	Text	Speech and vision (optional)
Output modalities	Text	Speech and robot motion
Evaluation criteria	Humanness (appropriate to the situation)	Seven items listed in Table 2, including “Naturalness of dialogue”, “Satisfaction with dialogue”, “Quality of service”, etc.

user utterances because they assumed such utterances would facilitate more engaging conversations in their task. Here, the dialogue that we want to achieve is different: the tasks of our systems have specific goals, and the dialogues need to feel natural to the user.

The rest of this paper is organized as follows. Section 2 gives overviews of the two dialogue system competitions that we participated in Sect. 3 describes the proposed guidelines that aided us in developing the two systems for the competitions. More details on the individual systems are given in Sects. 4 and 5, along with examples. Section 6 concludes the paper.

2 Dialogue System Competitions

We first give an overview of the two dialogue system competitions that we participated in:

- Situation track in the second Dialogue System Live Competition (DSLC2 situation track)[1]
- Dialogue Robot Competition (DRC)

Our team’s systems won first place in each competition. In both cases, the target language was Japanese.

Table 1 summarizes the main characteristics of the two competitions. The dialogues were conducted under specific situations and had fixed lengths. The dialogues in DSLC were text chats with the dialogue systems, while those in DRC were spoken dialogues with an android robot.

[1] https://dialog-system-live-competition.github.io/dslc2/ (written in Japanese).

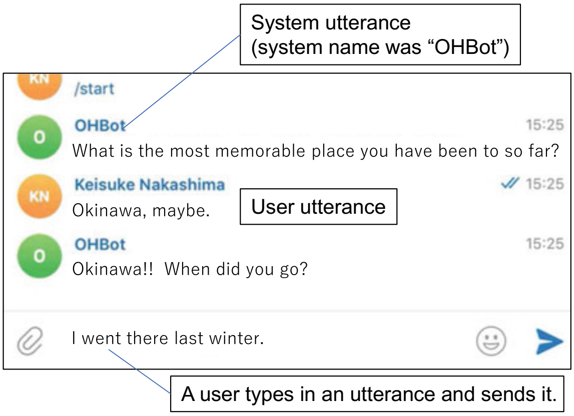


Fig. 1 Screenshot of the DSLC system (translated from Japanese)

2.1 DSLC2 Situation Track

The DSLC is a competition in which an audience watches and evaluates live dialogues between users and dialogue systems [8]. The dialogues are conducted as online text chats on Telegram.[2] A screenshot is shown in Fig. 1. After preliminary selection via crowdsourcing, three systems proceeded to the live event, performed dialogues with designated users who had been selected by the organizer, and were evaluated by the audience. The second edition took place in autumn 2019 and had two tracks: the open track and the situation track, in which our team participated.

The situation track used the following setup: "The user and the system are friends from their school days, and they start chatting on topics related to the most memorable trips and places they have been." System developers were allowed to specify their system's gender; that is, they could select male-to-male or female-to-female dialogues. We selected female-to-female dialogues. The length of a dialogue was 15 exchanges (i.e., pairs of user and system utterances). The track's evaluation criterion was "how human (appropriate to the situation) was the conversation the system conducted."

2.2 DRC

The DRC was held to promote improvement in the spoken dialogue technologies of android robots. In the DRC's dialogues, the robot acted as a travel agent and recommended a tourist spot to recruit participants acting as customers. The dialogues thus required the robot to provide information on tourist spots and hospitality that

[2] https://telegram.org/.

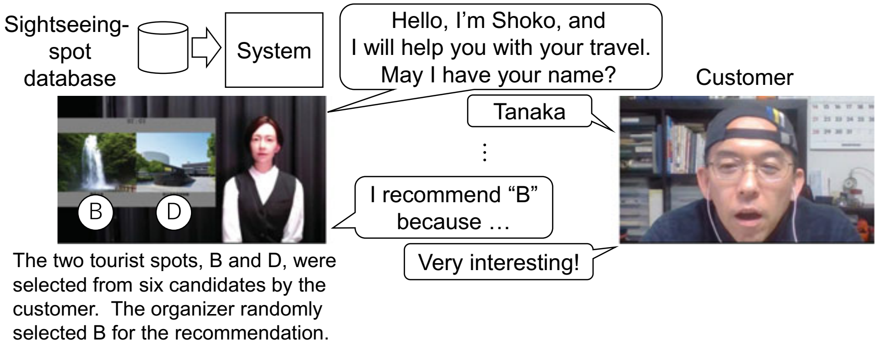


Fig. 2 Overview of DRC task

would satisfy the customer. Note that the customer interacted with the robot by voice, while the robot could speak and move its hands and head during a dialogue.

An overview of the DRC task is shown in Fig. 2. First, a participant acting as a customer selected two tourist spots from six candidates in advance. The competition organizer randomly specified one of the two spots that the robot should recommend. During the dialogues, the robot sought to persuade the participant to be interested in the specified spot. Pictures of the two spots were shown. The dialogue duration was five minutes. Each system was evaluated through questionnaires submitted by the participants after the dialogues.

The organizer provided basic modules for input and output to conduct the dialogues, such as speech-to-text, text-to-speech, and robot motion control modules. This enabled system developers to focus on the core dialogue design, while they could use their own recognition modules if they wanted. The knowledge of tourist spots was also provided as a database in advance.

3 Proposed Design Guidelines

As listed in Table 1 above, the competitions had the following important characteristics:

- The dialogues were of fixed length: 15 turns for DSLC2 and five minutes for DRC.
- The evaluation criteria explicitly included the user's impression.

Therefore, we needed a reasonable design that could give the user a good impression while establishing the dialogue in a particular situation, instead of an approach of collecting a huge amount of data and training the system to accept all kinds of user utterances. In particular, erroneous system utterances in such short dialogues would have fatally degraded the user's impression.

Accordingly, we developed three key guidelines from various aspects of the dialogue design requirements, as follows:

1. Make the system take initiative
2. Prevent dialogue flows from relying too much on user utterances
3. Include in utterances that the system understands what the user said

3.1 Make the System Take Initiative

The system should avoid being questioned by the user as much as possible. The reason is that the system is obligated to respond when the user asks a question, but it is very difficult to respond appropriately to all types of questions. Specifically, we sought to end the system's turn by asking a question or making an utterance that would elicit empathy from the user.[3]

A well-known finding from research on task-oriented dialogue systems is that novice users prefer system-initiative dialogues to user-initiative dialogues [11]. That finding also supports this guideline, because a first-time user can easily proceed with a dialogue by following the system's utterances, especially when the user does not know what the system can do. In addition, by ending an utterance in the form of a question, the system can explicitly indicate that it is the user's turn to speak, making it less likely that the user becomes confused about taking turns with the system.

At the same time, the system needs to avoid asking too many questions or speaking at length without ending its turn. Therefore, we divide the entire dialogue into several phases, which prevents the dialogue from becoming monotonous and enables us to develop each phase independently.

3.2 Prevent Dialogue Flows from Relying Too Much on User Utterances

We design dialogue flows to not rely too heavily on user utterances when possible. This can be the case when the subsequent dialogue can be established regardless of the user's response, and it is effective when the user's response is difficult to predict because the system's question has many possible answers (e.g., an open-ended question). It is impossible for a dialogue system to understand every possible user utterance correctly, although such systems are often expected to have this capacity. Moreover, even if a system could understand every possible user utterance correctly, it would be difficult to establish a system response for every possible user utterance. For this reason, an approach based on end-to-end machine learning has recently

[3] A similar guideline, "Avoid system utterances that may induce user questions," was also listed as a design guideline in our previous framework for developing closed-domain chat dialogue systems [14].

S14:	How about going there together?
U14:	Great. When are we going?
S15:	Let's talk over dinner together sometime, then.
S14:	How about going there together?
U14:	I guess I'm too busy for that.
S15:	Let's talk over dinner together sometime, then.

Fig. 3 Example of a system utterance S15 that does not depend on the preceding user utterance U14. The upper and lower examples are different dialogues. In this and subsequent examples, labels starting with S or U denote system or user utterances, respectively

been preferred, but, in this approach, the dialogue flow is left to machine learning results trained with a vast amount of data. It seems impossible to perfectly solve this problem and control what the system says. Confidence measures have been used to reject candidates with low confidence [17], but errors cannot be completely avoided even in that case.

Therefore, we design system responses to be valid even when the system does not correctly understand a user utterance.[4] Figure 3 shows an example of such a system utterance (S15). In both cases, the system asks a question (S14), but the user responses (U14) differ: the user is interested in the system's offer in the upper example but not in the lower example. The system response (S15) seems valid in either case.

For the competitions, the dialogue flows were designed by hand. To give a good impression in a fixed-length dialogue with designated tasks, this is a more reasonable approach than collecting a huge amount of dialogue data for each task and training a neural model to obtain such dialogue flows.

3.3 Include in Utterances that the System Understands What the User Said

Adding language understanding (LU) results of a user utterance to the system utterance tends to result in a good user impression [10]. The guideline discussed in Sect. 3.2 corresponds to ignoring user utterances, but if the system completely ignored the user utterances, it would degrade user impression.

Therefore, we include the LU result of the user utterance in the system utterance when the system seems to correctly understand it. This often becomes possible when the system asks a more specific question, rather than a vague one, and the LU result is quite likely to be correct, e.g., it matches the expected entries in the dictionary. Although this guideline conflicts with the second guideline, we aim to apply it when

[4] This approach is from a lecture given by Dr. Iio before DSLC2. The video (in Japanese) is available at https://dialog-system-live-competition.github.io/dslc2/lecture.html. It is part of the know-how shared in Prof. Ishiguro's Laboratory at Osaka University, where he previously worked and where several talking robots were developed.

S1:	What is the most memorable place you have been to so far?
U1:	Ishigaki Island, maybe.
S2:	You've been to Okinawa! The beaches are so beautiful and amazing! When did you visit there?

Fig. 4 Example of a system utterance evoking understanding of place names (Ishigaki Island is part of Okinawa)

possible so that the user will feel that the system is taking into account what he or she has said, thus giving a better impression.

In addition to simply adding the LU result of a user utterance (e.g., the user's name, the transportation method to get there, etc.) as it is, we also use domain ontology of place names to show understanding. Figure 4 shows this in a dialogue example. For DSLC2, we prepared an ontology of famous sightseeing spots including Ishigaki Island, which is part of Okinawa prefecture. In the example, this enables the system to express its understanding via "You've been to Okinawa!" in S2, which is better than just repeating a literal word in the user utterance.

Implicit confirmation requests in task-oriented dialogues have also been known to include the LU result of a user's utterance in the next system utterance [19]. The goal here is not to be certain of the correctness of the LU result, but to give the user a good impression, i.e., to show that the system understands what the user said. The tourist information task can be basically performed even if the system only speaks about tourist information in one way, but remembering what the user said can be considered to show that the system is treating the user well; this is an aspect of hospitality and thus improves the user impression.

4 System Design and Results for DSLC2 Situation Track

4.1 Specific Designs

4.1.1 Phase Design

We prepared three phases for this task of 15 utterances, i.e., about five utterances for each phase. The task was a chat with a female friend about memorable trips and places.

In the first phase, the system asked the user in turn about several "slots," such as the place of her most memorable trip, when she visited, her impression, and what she ate there. The system skipped asking about a slot if it had already been mentioned. For example, the question "When did you visit there?" was skipped if the user had already mentioned when she visited a particular place, e.g., "I went to Okinawa last summer." This was judged on the basis of the LU result of the user utterance.

In the second phase, the system talked about its experience on its own trip. This phase kept the dialogue from becoming monotonous and prevented the system from

S1:	What is the most memorable place you have been to so far?
U1:	Miyako Island was great.
...	...
S6:	I see. As for me, I've recently become addicted to sweets from Hokkaido :D

Fig. 5 Example of the system avoiding the same topic in the first and second phases. Emoticons are denoted by ":D"

continuing to ask questions. We also inserted a quiz in this phase to stimulate the user's interest and prevent the system from continuing to speak without ending its turn.

In the final phase, the system discussed future plans with the user. The system talked about a place where the user wanted to go and also invited the user to go together to a place where the system was planning to travel.

We also made the system avoid talking about the same place in the first and second phases. This was to prevent the same content from being repeated, which could make the user suspicious that the system does not understand the dialogue. By default, the system talked about its experience when it went to Okinawa in the second phase, but the topic changed to Hokkaido (another popular place) if the user had already talked about Okinawa in the first phase. This was judged on the basis of the LU results and the system's ontology (i.e., whether the place name in the first phase was part of Okinawa). Figure 5 shows an example of such avoidance. In the first phase (U1), the user talked about Miyako Island, which the system recognized as part of Okinawa prefecture by using its ontology. As a result, the system talked about Hokkaido in the second phase (S6).

4.1.2 Showing Empathy and Character Consistency

We also added expressions to show empathy at the beginning of an utterance after asking the user a question and getting a response from her. These expressions included "right," "indeed," "I know." This was to prevent the system from being perceived as not listening to the user. The system changed the expressions to avoid degrading the user's impression by repeating the same one.

Because the task setting was a chat between close female friends, we took care to ensure that the expressions and content of utterances and the use of emoticons (denoted by ":D" in this paper) were consistent with how a Japanese woman would use them. The consistency was checked by a female developer and another woman several times.

S1:	What is the most memorable place you have been to so far?
U1:	Well, it's all too common, but maybe Hawaii.
S2:	Hawaii!! When did you go there?
...	...
S14:	... How about going there together?
U14:	I can't wait to go!
S15:	Let's talk over dinner together sometime, then :D

Fig. 6 Excerpt of a dialogue log for our system during the live event of DSLC2

S13:	I'm actually thinking of going to Shinshu next spring.
U13:	Shinshu is also nice. What are you doing there, skiing or snowboarding?
S14:	There are lots of yummy things, like apple sweets :D My friend's parents' house is ...

Fig. 7 Example of a failure due to incorrect type estimation

4.2 Results and Examples

Our system received the highest score among the seven systems that participated in the preliminary round of the situation tracking of DSLC2. It received an average score of 4.1 on a five-point Likert scale from 50 crowd workers. The criterion was "how human (appropriate to the situation) was the conversation the system conducted."

After proceeding to the final live event of DSLC2, our system won a first place through evaluation by the audience of about 100 people including dialogue system researchers. The logs are available to the public and include the evaluated label distributions for each utterance in the live event.[5] The best distributions were given to the two system utterances S2 and S15, shown in Fig. 6: 91.6% of the audience gave the highest label of "Good", on a three-point scale. Utterance S2 showed the LU result of the place name "Hawaii," as mentioned in Sect. 3.3. Utterance S15 was similar to those in Fig. 3: the system utterance did not depend on the previous user utterance, as discussed in Sect. 3.2.

There were several utterances that seemed inappropriate in the context, however. Figure 7 shows an example of such a failure in the preliminary round. Utterance U13 was wrongly assigned the same dialogue-act type as the question "What is in Shinshu?" by the LU module. As a result, the system started the following utterance S14 with a prepared sentence about a local specialty. This kind of mistake could be avoided by adding a correct dialogue-act type to the LU module and training it, but such errors are inevitable.

[5] https://dialog-system-live-competition.github.io/dslc2/result.html.

	(robot turns toward the pictures on display)
S:	The left picture shows Minoh Falls, and the right one shows Soji-ji temple.
	(robot turns toward the participant)
S:	Which picture, right or left, impresses you more?
U:	The right one.
S:	Oh, nice. I also think so.
...	...

Fig. 8 Question about two pictures of tourist spots in the explanation phase

5 System Design for DRC and Results

5.1 Specific Designs

5.1.1 Phase Design

A dialogue system for this task was supposed to provide information on tourist spots to a customer and take the customer's travel request. We divided the entire dialogue of five minutes into four phases: (1) introduction, (2) explanation of tourist spots, (3) recommendation, and (4) Q&A.

In the first phase, the system gave the customer greetings and simple questions as ice breakers. The questions were about (1) the customer's experience talking to robots, (2) the customer's name, (3) the transportation method to be used, and (4) any traveling companions. The system stored the customer's answers for use in a later phase. If automatic speech recognition (ASR) or LU failed, a default value was then used.[6] For example, if the system could not recognize the customer's name, it instead called the customer "Sir" or "Madam".

In the second phase, the system gave an outline of the two tourist spots that the customer had selected and then gave him or her the more specific information on the two spots. Specifically, the system gave descriptions of the spots, explained how to access them, and mentioned categories such as "temples and shrines," "factories and facilities," and so on. The system also asked the customer which picture he or she preferred between pictures of the two spots, as shown in Fig. 8. The answer to this question was used as a reason for the recommendation in the following phase.

In the third phase, the system recommended one of the two tourist spots and explained the reasons. We prepared sentences with recommendation reasons in advance, and the system selected them according to what the customer said during the dialogue, such as his or her preference between the two pictures. The most specific reasons we had prepared were related to the customer's preference for the touring spot's category, which the system asked during the dialogue, as shown in Fig. 9. Giving more specific reasons would be better because a customer would not be convinced by general reasons that are not related to specific spots; however, giv-

[6] This is similar to the "MoveOn" strategy [3].

S:	Do you like spots in the "rivers, canyons, and water falls" category?
U:	Yes.
S:	Then I strongly recommend that you visit Minoh Falls. You can feel relaxed and comfortable.
S:	Also, you said "My impression of the picture is good."
...	...

Fig. 9 Examples of recommendation reasons in the recommendation phase

ing more detailed reasons would require complicated ASR and LU technologies to understand the customer's preferences and experiences, which would increase the risk of misunderstanding.

In the final "Q&A" phase, the system answered questions from the customer as long as time permitted. Because it was difficult to answer the customer's open-ended questions, the system gave the customer several examples of what kinds of questions could be answered. When five minutes had passed from the beginning of the dialogue, the system ended dialogue with closing remarks to the customer.

5.1.2 Strategies in LU and Turn Taking

Our strategy to reduce misunderstandings caused by LU failures was to show a few words or phrases as examples in each system question. This was because our LU approach was based on pattern matching between recognized character sequences and a prepared word set. We expected customers to utter one of the examples as the answer. For example, when the system asked "Are you planning to use a private car, train, or other means of transportation?", the customer's possible answers were almost entirely restricted to "private car," "train," and several other words. We also added similar and possibly misrecognized expressions to the original set of expected words and phrases.

We also manually designed the timing of when the system accepted a customer utterance, i.e., whether the customer was allowed to barge into system utterances. For example, to avoid unexpected situations, the system basically did not accept any customer utterances while it was explaining something. In other words, the system only attempted to understand customer responses to its explicit questions.

An elapsed time after each system utterance was used to maintain turn taking even when ASR or LU failed. After a certain amount of time had elapsed, the system was designed to say something. Without this capability, if the system could not detect a customer's utterance while waiting for an answer, both the system and the customer might have had to wait and, the silence would have continued; instead, the system made a confirmation utterance about the current situation or moved on to the next utterance.

5.1.3 Speech Synthesis and Robot Motions

The pronunciations of the system utterances and the speaking speed were carefully checked in advance. Unlike in text chats, these factors are important because they affect the customer impression. For example, if the speaking speed is extremely fast or slow, customers may feel stressed.

Coordination of the robot motion with the utterance is also important, because it would be strange if the robot did not move at all. Accordingly, the robot shook its body slightly and slowly, and it blinked its eyes by default. We also created two specific motions: bowing upon greeting the customer and turning toward the display when showing the pictures of the two touring spots. The latter motion was designed to create joint attention by guiding the customer's eyes to the pictures. A dialogue example with this kind of motion is shown in Fig. 8.

5.2 Results of Pre-preliminary Contest and Examples

The DRC's pre-preliminary contest was held in March 2021. Nine systems were evaluated, including the organizer's baseline system. Because of the COVID-19 pandemic, the recruited participants performed dialogues with the android robot via remote software. Each participant had a maximum of one to three dialogues and filled out a questionnaire after each dialogue. Each system was scored by about 10 participants.

The questionnaire items were prepared by the organizer and are listed in Table 2. Each item was scored on a seven-point scale. The table also lists the average scores of our system and the baseline for each item. The score gaps between them were typically over one point.

Table 2 Questionnaire items and average scores of the systems

Questionnaire items	Our system	Baseline
1. Satisfaction with recommendation	5.9	4.5
2. Amount of provided information	5.7	4.9
3. Naturalness of dialogue	4.9	4.0
4. Appropriateness	5.7	4.2
5. Satisfaction with dialogue	5.3	4.0
6. Quality of service	6.0	4.4
7. Usefulness of provided information	6.3	4.6

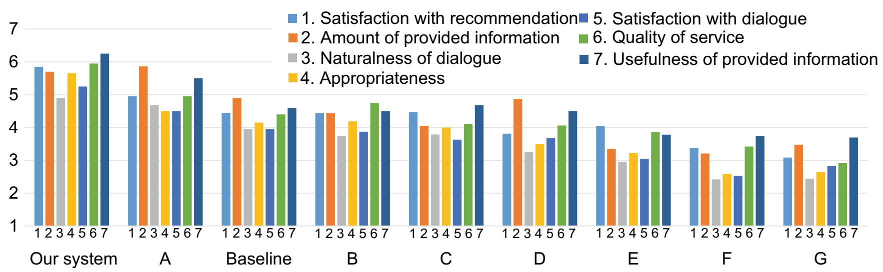


Fig. 10 DRC Results. The vertical axis represents the average user impression on a seven-point scale, with a higher score indicating a better impression

Figure 10 shows the score distributions for all items and systems. The letters A to G represent the systems developed by the other teams. Our system received the highest scores for most of the questionnaire items, while the baseline was ranked third. Note that these scores were not relative among the systems because each system was evaluated by different participants.

The better-scored questionnaire items for our system were related to providing information on the tourist spots and the appropriateness of the system response: (1) satisfaction with recommendation, (2) amount of provided information, (4) appropriateness, (6) quality of service, and (7) usefulness of provided information. The scores of these items were better because our system had a specific phase for explaining the tourist spots. The system's responses were considered appropriate because the system was designed and tuned manually to avoid unexpected system behaviors. It was thus natural that these scores were relatively high because we manually prepared guidance for the system on the task of providing tourist information. In addition, the robot's motions of bowing, nodding, and turning toward the display to create joint attention might have resulted in better scores for the appropriateness of the system's behavior.

The relatively worse-scored items were related to the naturalness of the dialogues: (3) naturalness of dialogue and (5) satisfaction with dialogue. We suggest that this was partly because our system did not allow the customer to speak freely during system utterances. This was an inevitable choice to avoid dialogue failures, given the current performance of ASR and LU technologies: it would have become more difficult to correctly understand the customer's utterances and continue the dialogue if the system allowed the customer to barge in and tried to recognize every utterance. Figure 11 shows an example. Here, all ASR results were rejected while the system was speaking; that is, the question from the participant (U11) was ignored. This behavior might have worsened the score for the naturalness of the dialogue.

In addition, Fig. 12 shows an example in which the name recognition failed. The participant replied "That's right (*Yoi desu*)" as U7, but the system misunderstood "*Yoi*" as the participant's name because it was not included in the dictionary as a variant expression of "right (*yoi*)" for LU based on pattern matching. Furthermore, the user utterance U8 was ignored because barge-in was not allowed: its ASR result

time		
239.0	S31:	I want you to go to both places, but I recommend Meiji-Naruhodo Factory, Osaka.
241.7	U10:	Thanks.
249.3	S32:	First, you can enjoy sightseeing and talking with your friends.
251.3	U11:	Why? # This was rejected and ignored because it was during system utterance.
255.9	S33:	Do you like factory and facility tours?
264.8	U12:	Yes, I like them.

Fig. 11 Example of a failure due to the system not allowing the customer to barge in during system utterance

time		
37.7	S7:	First, may I have your name?
39.1	U5:	Yes.
44.0	U6:	Uh, I'm XXX. (*E, XXX desu*)
46.7	S8:	Your name is XXX, right?
52.5	U7:	*Yoi desu*
53.8	S9:	Your name is Yoi, right?
		# "*Yoi*" means "right," but it was not in the dictionary and was misunderstood as a name.
59.6	U8:	No, I'm XXX. (*Ie, XXX desu*) # This ASR result was obtained at 62.3.
61.7	S10:	I'm sorry, but let me skip confirming your name because it was hard for me to hear.
68.0	S11:	Sir, which are you going to use, a private car or public transportation, during your travel?

Fig. 12 Example of a failure of name recognition. XXX was the customer's name but is anonymized here for privacy. Text in *italics* represents literal Japanese transcriptions

was obtained at a time of 62.3, which was just after the system had started speaking S10 (at 61.7). Such behavior might also degrade the naturalness of dialogue.

6 Discussion and Conclusion

We have proposed design guidelines for developing dialogue systems for competitions. Our systems developed with the aid of these guidelines won first place in two competitions.

The guidelines here correspond to a previous experimental result from a user impression analysis of a chat dialogue system in the food and restaurant domain [14]. That study showed correlations between the main questionnaire item ("I'm willing to chat with the system again") and seven other items. The three items with higher correlations were "The dialogue was fun," "The dialogue was natural," and "The system understood my utterances."[7] The second item corresponds to one of our design guidelines, namely, the guideline to prevent dialogue flows from relying too much on user utterances (Sect. 3.2) in order to avoid disruptions by ASR and LU errors and make the dialogue as natural as possible. The third item corresponds

[7] The remaining four items with lower correlations were as follows: "The dialogue went well," "The system was polite," "The system was friendly," and "The system did not often change the topic." These items were of relatively lower importance, given the current system performance.

to our guideline to include in utterances that the system understands what the user said (Sect. 3.3). On the other hand, the first item may depend heavily on the dialogue content, which is beyond the scope of this paper.

It would be good to quantify the impact of the proposed design guidelines, but that would require a new experimental design, and it is thus beyond the scope of this paper. Nevertheless, we hope that these design guidelines will inspire developers of other dialogue systems.

Acknowledgements This work was partly supported by JSPS KAKENHI Grant Numbers JP19H05692 and JP19H04171, and JST, PRESTO Grant Number JPMJPR1857, Japan.

References

1. Adiwardana D, Luong M, So DR, Hall J, Fiedel N, Thoppilan R, Yang Z, Kulshreshtha A, Nemade G, Lu Y, Le QV (2020) Towards a human-like open-domain chatbot. CoRR arXiv:2001.09977
2. Bernsen NO, Dybkjaer H, Dybkjaer L (1996) Principles for the design of cooperative spoken human-machine dialogue. In: Proceedings of international conference on spoken language processing (ICSLP), vol 2, pp 729–732. https://doi.org/10.1109/ICSLP.1996.607465
3. Bohus D, Rudnicky AI (2005) Sorry, I didn't catch that! - an investigation of non-understanding errors and recovery strategies. In: Proceedings of SIGdial workshop on discourse and dialogue, pp 128–143
4. Clark HH (1996) Using language. Cambridge University Press
5. Deibel D, Evanhoe R (2021) Conversations with things: UX design for chat and voice. Conversations with things: UX design for chat and voice, Rosenfeld media. https://books.google.co.jp/books?id=BlouzgEACAAJ
6. Grice HP (1975) Logic and conversation. In: Cole P, Morgan JL (eds) Syntax and semantics: Vol. 3: speech acts. Academic, New York, pp 41–58. http://www.ucl.ac.uk/ls/studypacks/Grice-Logic.pdf
7. Hardy A, Paranjape A, Manning C (2021) Effective social chatbot strategies for increasing user initiative. In: Proceedings of annual meeting of the special interest group on discourse and dialogue (SIGDIAL), pp 99–110. https://aclanthology.org/2021.sigdial-1.11
8. Higashinaka R, Funakoshi K, Inaba M, Tsunomori Y, Takahashi T, Akama R (2019) Dialogue system live competition: identifying problems with dialogue systems through live event. In: Marchi E, Siniscalchi SM, Cumani S, Salerno VM, Li H (eds) Proceedings of international workshop on spoken dialogue system technology (IWSDS). Lecture notes in electrical engineering, vol 714. Springer, pp 185–199. https://doi.org/10.1007/978-981-15-9323-9_16
9. Hu S, Liu Y, Gottardi A, Hedayatnia B, Khatri A, Chadha A, Chen Q, Rajan P, Binici A, Somani V, Lu Y, Dwivedi P, Hu L, Shi H, Sahai S, Eric M, Gopalakrishnan K, Kim S, Gella S, Papangelis A, Lange P, Jin D, Chartier N, Namazifar M, Padmakumar A, Ghazarian S, Oraby S, Narayan-Chen A, Du Y, Stubell L, Stiff S, Bland K, Mandal A, Ghanadan R, Hakkani-Tur D (2021) Further advances in open domain dialog systems in the fourth alexa prize socialbot grand challenge. In: Alexa prize proceedings
10. Kobori T, Nakano M, Nakamura T (2016) Small talk improves user impressions of interview dialogue systems. In: Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pp 370–380, https://www.aclweb.org/anthology/W16-3646
11. Komatani K, Ueno S, Kawahara T, Okuno HG (2005) User modeling in spoken dialogue systems to generate flexible guidance. User Model User-Adap Interact 15(1):169–183
12. Komeili M, Shuster K, Weston J (2021) Internet-augmented dialogue generation. CoRR arXiv:2107.07566

13. Langevin R, Lordon RJ, Avrahami T, Cowan BR, Hirsch T, Hsieh G (2021) Heuristic evaluation of conversational agents. In: Proceedings of conference on human factors in computing systems (CHI). https://doi.org/10.1145/3411764.3445312
14. Nakano M, Komatani K (2020) A framework for building closed-domain chat dialogue systems. Knowl-Based Syst 204:106212. https://doi.org/10.1016/j.knosys.2020.106212
15. Niculescu A (2011) Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality. PhD thesis, University of Twente. https://research.utwente.nl/en/publications/conversational-interfaces-for-task-oriented-spoken-dialogues-desi, sIKS Dissertation Series ; no. 2011-49
16. Nielsen J, Molich R (1990) Heuristic evaluation of user interfaces. In: Proceedings of conference on human factors in computing systems (CHI), pp 249–256. https://doi.org/10.1145/97243.97281
17. Pearl C (2016) Designing voice user interfaces: principles of conversational experiences, 1st edn. O'Reilly Media, Inc
18. Roller S, Dinan E, Goyal N, Ju D, Williamson M, Liu Y, Xu J, Ott M, Smith EM, Boureau YL, Weston J (2021) Recipes for building an open-domain chatbot. In: Proceedings of European chapter of the association for computational linguistics (EACL), pp 300–325. https://aclanthology.org/2021.eacl-main.24
19. Sturm J, den Os E, Boves L (1999) Issues in spoken dialogue systems: Experiences with the Dutch ARISE system. In: Proceedings ESCA workshop on interactive dialogue in multi-modal systems, Kloster Irsee, Germany, pp 1–4
20. Xu J, Szlam A, Weston J (2021) Beyond goldfish memory: Long-term open-domain conversation. CoRR arXiv:2107.07567
21. Yang X, Aurisicchio M (2021) Designing conversational agents: a self-determination theory approach. In: Proceedings of conference on human factors in computing systems (CHI). https://doi.org/10.1145/3411764.3445445

Understanding How People Rate Their Conversations

Alexandros Papangelis, Nicole Chartier, Pankaj Rajan, Julia Hirschberg, and Dilek Hakkani-Tur

Abstract User ratings play a significant role in spoken dialogue systems. Typically, such ratings tend to be averaged across all users and then utilized as feedback to improve the system or personalize its behavior. While this method can be useful to understand broad, general issues with the system and its behavior, it does not take into account differences between users that affect their ratings. In this work, we conduct a study to better understand how people rate their interactions with conversational agents. One macro-level characteristic that has been shown to correlate with how people perceive their interpersonal communication is personality [1, 2, 12]. We specifically focus on agreeableness and extraversion as variables that may explain variation in ratings and therefore provide a more meaningful signal for training or personalization. In order to elicit those personality traits during an interaction with a conversational agent, we designed and validated a fictional story, grounded in prior work in psychology. We then implemented the story into an experimental conversational agent that allowed users to opt in to hearing the story. Our results suggest that for human-conversational agent interactions, extraversion may play a role in user ratings, but more data is needed to determine if the relationship is significant. Agreeableness, on the other hand, plays a statistically significant role in conversation ratings: users who are more agreeable are more likely to provide a higher rating for their interaction. In addition, we found that users who opted to hear the story were, in general, more likely to rate their conversational experience higher than those who did not.

A. Papangelis (✉) · N. Chartier · P. Rajan · J. Hirschberg · D. Hakkani-Tur
Amazon Alexa AI, Sunnyvale, USA
e-mail: papangea@amazon.com

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_12

1 Introduction

User feedback is one of the most important pieces of information we can use to improve various modules of conversational agents. Such feedback is usually provided by directly asking users to rate their experience (e.g., on a scale of 1–5). These ratings are typically averaged and used as a measure of the agent's conversational skills.

One limitation of this approach is that it treats all users as a homogeneous whole. However, each user is different; they have different experiences, personalities, needs, and expectations that can lead them to perceive an interaction with the same conversational agent differently. Treating conversational ratings as monolithic will lead to a conversational agent that tends to an "average" user, rather than being personalized to each individual user. While for some users, an "averaged" approach will not have an adverse effect on the conversational experience, this approach may lead to sociodemographic and personality bias in the agent, and negative experiences for some users.

Because of this, we postulate that learning about users, and using that information to personalize a conversational agent, will improve the user's conversational experience and thus, improve their rating of the conversation. In this work, we propose approaching conversational ratings through the lens of users' personality to address the question *Does a user's personality play a role in the rating they provide?*

We chose to focus on personality for two reasons. First, personality is a well-documented and researched area in which individual variation can be explained using macro-level categorization [5–7, 9]. As such, methods for measuring individuals' personality traits via surveys have been thoroughly assessed and validated. Second, previous research suggests that two personality traits, extraversion and agreeableness, influence a user's evaluation of their interaction with conversational agents [1]. To gain some insight into users' personalities, we constructed a story that includes questions about agreeableness and extraversion and integrated it into a conversational agent. This serves as a novel approach to the traditional personality survey format (i.e., filling out a questionnaire). Adapting traditional questionnaire-type personality questions within the story allows users to provide self-assessments of their own personality while engaging with the conversational agent. We then used this information to examine the relationship between these aspects of a user's personality and their ratings.

In this paper, we outline our reasoning and methods for developing and validating our story approach to the personality survey. Next, we explain how we implemented the personality story into a conversational agent. Finally, we discuss the results of the story method by first describing whether or not users engaged with the story and second by addressing our primary research question: *Does a user's personality play a role in the rating they provide?*

2 Related Work

Previous work in psychology has identified five traits that can be used to describe an individual's personality: agreeableness, extraversion, neuroticism, openness, and conscientiousness [5]. Of these, extraversion and agreeableness are the two personality traits that have been shown to influence interpersonal communication most, as they point to characteristics such as sociability, affability, and kindness [2, 12]. Examining human-human interactions, [2] demonstrate that individuals who score higher on extraversion are more likely to report their interaction as smooth, natural, and relaxed, and individuals who score higher on agreeableness as more likely to positively evaluate the quality of the interaction.

Not only have extraversion and agreeableness been demonstrated to impact interpersonal communication between two human interlocutors, but these traits have also been shown to influence human-AI dyads. Reference [1] examined how user personality traits influence their evaluations of interactions with text chatbots. The study consisted of participants completing a traditional personality survey and a short interaction with a chatbot who used the same five pre-recorded sentences with each participant. The results demonstrate that extraversion and agreeableness were better predictors of participants' interaction evaluations than the chatbot's behavior.

While the importance of personality in human-AI conversations has influenced the development of personality classifiers [8, 10, 11, 13, 14, e.g.,], current automatic personality classification approaches were not suitable for our purposes. First, our experimental setup depends on text, not speech; and as such, we could not use a classifier that depends on spoken features [8, 10]. Second, our research question depends on an accurate assessment of personality in order to assess the relationship between personality and user ratings. Thus, we did not want to make the assumption that a classifier that is evaluated on a constructed dataset, a different demographic, or experimental setup in general [11, 13, e.g.] would transfer to our human-AI interactions, where the expressions of personality are much more nuanced. Further, we would have no way of training or validating such models' performance, as due to several constraints, we could not ask users of our experimental agent to fill out a traditional personality survey.

3 Method

Taking into consideration that the personality survey would be implemented by a conversational agent, we determined that the traditional survey design (in the form of questionnaires) would not lead to engaging interactions. Instead, we developed a personality questionnaire in which the survey items are incorporated into a story that the conversational agent tells the user (see Table 1). The conversational agent provides short scenarios that lead to questions for the user to engage with regarding their own self-assessment on the dimensions of extraversion and agreeableness. This method allows for a more natural conversational approach to eliciting personality

Table 1 The story above was used to collect personality information about users. Not all turns collected personality information—some were used as simple engagement questions. The personality questions are in bold font, and the specific trait (extraversion or agreeableness) are identified in parentheses

Turn	Conversational agent utterance
1	I went to a fun party in the cloud yesterday, do you want to hear about it? (User chooses whether or not to hear the story)
2	My friend was having a birthday party and I was so excited! **I just love going to parties, what about you?** (extraversion)
3	Oh interesting! Well, when I got on the bus, I saw my best friend! They're my favorite person to talk to! I can talk to them about anything. Umm, do you have someone in your life like that?
4	Thanks for sharing. I think it's because they do such a great job sympathizing with my feelings. It usually helps me feel a lot better. **I'm just curious, are you good at sympathizing with other people's feelings?** (agreeableness)
5	I see, I'm not always very good at it. So when we got to the party, there were already lots of people there! **Do you like talking to a lot of different people at parties?** (extraversion)
6	When I was looking around, I saw someone who looked pretty sad. I thought we should go check on them, but one of my friends wasn't interested. **Would you have gone to check on them?** (agreeableness)
7	Well, I decided to go over. It turns out they were having a really bad day, and they were happy I came over to chat. We talked about their dog, and they showed me lots of pictures. After a little while, I went back to find my friends. Would you like to hear more about the party? (User chooses whether or not to continue the story)
8	It was so much fun! There was karaoke and dancing! Do you enjoy karaoke?
9	Oh, interesting! I am not very good at it, but I love it! One of my friends didn't participate too much, and kind of stayed in the background. **Do you tend to keep in the background at parties too?** (extraversion)
10	Yeah, I think it was because they didn't know that many people there. What about you? **Are you usually quiet around new people?** (extraversion)
11	Toward the end of the night, people brought out a big cake and sang happy birthday! It was such a great night and party. I had a blast and met so many new people! Thanks for letting me share my story. What did you think of it?

questions. Further, by incorporating the survey items into a story, we expect users to feel more comfortable answering the questions, as the questions become part of the system sharing a story and attempting to create user engagement with the story. At the end of the story, the system asks the participant for feedback on the story as a means of understanding the performance of the story separate from the users' overall conversational experience.

This section serves a dual purpose. First, we will outline the method and results for developing and testing the personality story. Second, we will outline the method for implementing the story within the conversational agent.

3.1 Personality Story

In order to assess the quality of the results we could expect from our story method within the conversational agent, we first conducted a series of pilot studies in which we had participants complete a traditional personality survey [7] shown in Table 2, our story survey, and five filler questions that asked participants to describe an image in five words or less. We semi-randomized the order: participants either began the study completing the traditional survey or engaging with the story. The filler questions always occurred between these two in an effort to mitigate the repetition of the two personality-eliciting methods.

The traditional personality survey utilized a 6-point scale, wherein each numeric point represented the degree to which an individual agreed or disagreed with a given statement. Participant responses consisted of a numeric self-assessment of each survey item. For example, one survey item presented to participants was “I am not interested in other people’s problems.” In contrast, the story method consisted of collecting text responses to the personality-probing questions embedded in the story. These questions were constructed to elicit yes/no responses from participants. For example, one excerpt from the story said “I saw a person who looked pretty sad. I thought we should go check on them, but my friend wasn’t really interested. Would you have gone over?” These responses were then labeled using a 3-point scale (e.g., positive, negative, and neutral).

We collected data from approximately 100 participants on Amazon Mechanical Turk (mTurk). After excluding participants that did not follow the instructions, there

Table 2 For the traditional personality survey, users were presented with 16 statements and instructed to indicate how much they agreed or disagreed with each statement. The 16 items from [6]’s Big Five Factor Markers for extraversion and agreeableness

Extraversion	Agreeableness
I am quiet around strangers	I feel others’ emotions
I start conversations	I am not really interested in others
I don’t like to draw attention to myself	I insult people
I keep in the background	I have a soft heart
I talk to a lot of different people at parties	I sympathize with others’ feelings
I have little to say	I take time out for others
I don’t mind being the center of attention	I make people feel at ease
I don’t talk a lot	I am not interested in other people’s problems

Table 3 Cronbach's Alpha values for the story and survey on both personality traits

	Agreeableness	Extraversion
Story	0.42	0.9
Survey	0.84	0.91
Story & Survey	0.86	0.94

were 96 participants for analysis. We analyzed the results of the story in two ways. First, we computed Cronbach's Alpha for the two traits (agreeableness and extraversion). Second, we calculated the mean score for each participant for agreeableness and extraversion for both the survey and the story, and then computed a simple linear regression to determine whether a participant's survey score could be predicted by their story score.

To see if an individual's overall score for agreeableness and extraversion from the story could predict their score on a traditional survey, we ran a simple linear regression. We found both of these models to be statistically significant (Agreeableness: $R^2 = 0.3221$, $F(1, 96) = 47.09$, $p < 0.01$; Extraversion: $R^2 = 0.6691$, $F(1, 96) = 197.1$, $p < 0.01$).

These results suggest that for extraversion, the story items are both internally consistent **and** significant predictors for an individual's score on a traditional personality survey. For agreeableness, however, these results suggest that while the story may not be internally consistent, they still will predict an individual's score on a traditional personality survey (Table 3).

3.2 Implementing the Story in a Conversational Agent

After validating that the story could replace a traditional personality survey, it was integrated into an experimental conversational agent that can have conversations on various topics such as movies, news, or pets. Users of the conversational agent are free to interact for as long as they like and at the end of each conversation, the conversational agent asks for feedback on a scale from 1 to 5. While the conversational agent is based on neural networks, the story was designed in a way that there is continuity regardless of the users' input, to avoid complicated flows that would make analyzing the results difficult. Therefore, it was implemented as a Finite State Machine, meaning that, for the story part, the conversational agent would move on to say pre-defined utterances ensuring that each user had the exact same experience.

Our setup allows opt-in engagement. The conversational agent first asks the user if they want to hear a story and if the user responds positively, the story begins. Since the story is somewhat long (11 turns), we introduced a second point where the conversational agent re-affirms that the user wants to continue listening to the story. Figure 1 shows the flow. Once the conversational agent completes the story, it asks

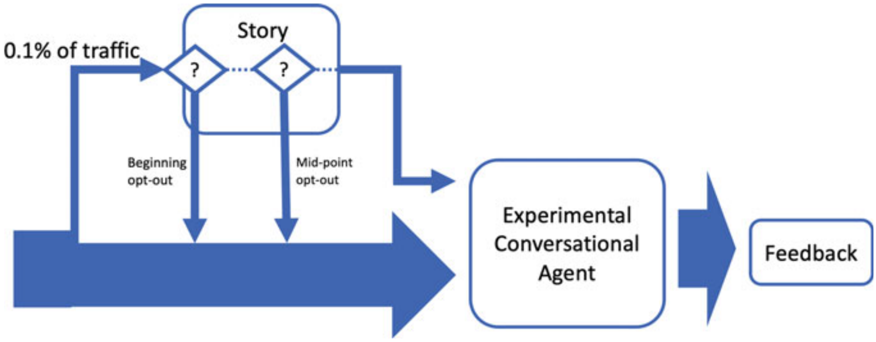


Fig. 1 Flowchart showing how our story fits into the experimental conversational agent. "?" represents the beginning and mid-point question where users have the option to exit or continue with the story

the user what they would like to talk about next, thus allowing the user to continue the conversation as they wish. As such, while the story itself is FST-based, to ensure that each user heard exactly the same story, exactly the same way, the conversation after the story was dependent on the conversational agent.

4 Results

4.1 User Reactions to Personality Story

760 users were asked if they would like to hear the story. 70% of them agreed and 30% declined to hear the story. Users that engaged with the system's story include those that engaged with the entire story and those that engaged with part of the story. Although 760 users conversed with the agent, only 307 provided a rating for the overall conversation experience. Table 4 describes the distribution of those who engaged and did not engage with the story and shows that of these two groups, the average conversational score was higher for users who opted-in to the story. Welch's t-test showed the difference was statistically significant, $t(113.16) = -3.3634$, $p = 0.001$. Although it appears as though users who engage in the story have longer conversations than those who do not, the average number of turns in Table 4 includes the story (11 turns). On average, users who listen to the story have the same length of conversation with the agent after the system tells its story as those who do not listen to the story. In other words, these users are not simply listening to the story and ending the conversation.

Table 4 Distribution of average rating and number of turns by those who engaged with the story (yes) and those who did not engage with the story (no)

	Yes	No
Total no.	532	228
No. of rated conversations	233	74
Average rating	3.63	3
Average no. of turns	28.1	16.5

4.2 Ratings Through Lens of User Personality

Lastly, we address our original research question: *Does a user's personality play a role in the rating they provide?* To address this question, we examined the responses from 233 users who engaged with the personality story and provided a rating. We manually annotated the user responses to the personality questions. The responses are labeled on a 3-point scale (0–2). For example, if a user responds to the question "Do you like going to parties, too?" with "No", the user receives a 0 for extraversion. Through our manual annotation, we excluded participants whose responses could not be adequately interpreted, e.g., users who did not answer the questions or whose responses were not relevant. We then computed the mean scores for agreeableness and extraversion for 195 users, where a higher score indicates a higher level of agreeableness/extraversion. Figure 2 shows the number of users with a particular agreeableness and extraversion average score. In general, we note that our sample is skewed for agreeableness—users who score high on agreeableness were more likely to opt in to the story and provide a rating. The scores for extraversion are more evenly distributed between high, mid, and low values.

Next, we fit a general linear model using agreeableness and extraversion as predictors of conversational ratings. We found a significant relationship between agreeableness and conversational ratings, i.e., those who are more agreeable are more likely to provide a higher rating ($\beta = 0.316$, $p = 0.02$). However, the relationship between extraversion and conversational ratings is only nearing significance, i.e., those who are more extraverted are more likely to provide a higher rating ($\beta = 0.215$, $p = 0.098$). Agreeableness and extraversion explained a small proportion of variance in conversational ratings, $R^2 = 0.045$, $F(2, 191) = 4.519$, $p = 0.012$.

For agreeableness, these results support the findings from [1, 2]: users who are more agreeable are more likely to rate their overall conversational experience higher than those who are not. For extraversion, however, more data is necessary in order to determine if the relationship is significant. It is important to note that the low R^2 value suggests that there are other factors that should be included to understand variation in ratings.

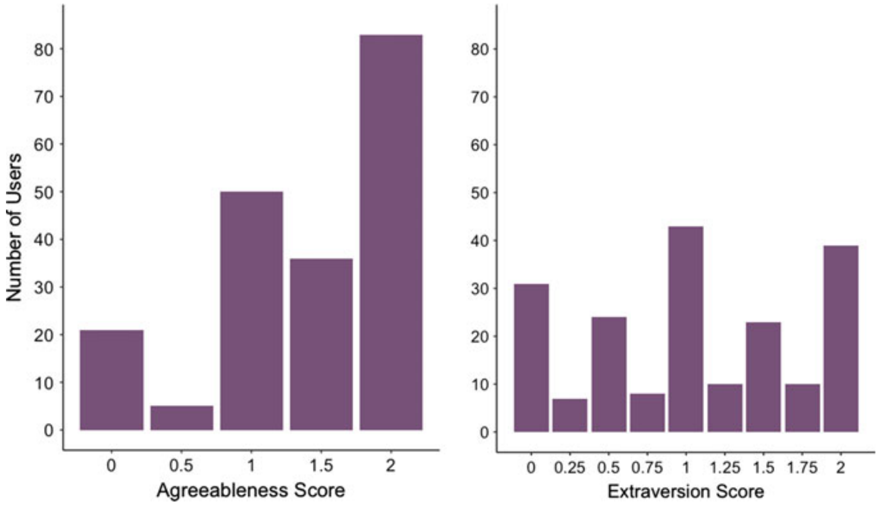


Fig. 2 Number of users with a given average score for agreeableness (left) and extraversion (right). Note that because there were a different number of questions for agreeableness (2) and extraversion (4) in the story (see Table 1), there are different average scores for the two traits

5 Discussion

In this study, we have constructed a conversational personality survey that can be implemented into a conversational agent that allows opt-in participation by means of engaging with a story. This approach provides two advantages. First, the story approach helps to mitigate the effects of social desirability bias [3, 4] by focusing attention on engaging with a story. Second, utilizing an opt-in approach to the personality story, we avoid negatively impacting user experience. Not only is this approach advantageous for administering personality surveys, but it can be adapted to elicit other types of user characteristics that are typically obtained through traditional surveys.

Results from the personality story suggest that extraversion does not predict users' overall experience with the experimental conversational agent. While this may be a reflection of a lack of data, this could also be a reflection in the difference between human-human interactions and human-AI interactions. Extraverted people are generally described as companionable, talkative, and confident [2, 9], and the nature of the conversations between the user and the conversational agent are usually ones in which the conversational agent directs the conversation. In other words, a conversational partner of a conversational agent does not need to be confident and talkative, as the conversational agent tends to lead the conversation.

On the other hand, results from the personality story show that agreeableness does predict overall conversation ratings. These results suggest that perhaps agreeableness between human-human interactions is more likely to transition to human-AI interactions than extraversion. Agreeable people are generally described as being sympathetic, cooperative, and considerate [9]. Sympathy and cooperation from the user can help to alleviate some of the conversational limitations of the agent. While it is possible that those who were more agreeable provided higher conversation ratings because the quality of the conversation was better, it is more likely that those who are more agreeable are simply more likely to rate conversations higher than those who are less agreeable. A qualitative review of the conversations post-story will need to be conducted in order to address this.

An interesting finding from this study is that users who chose to listen to the personality story tend to score high on agreeableness and also tend to provide higher ratings for their conversational experience. There are a few potential reasons that this could be the case. First, it is likely that users who are more agreeable are more likely to listen to the conversational agent's story. In this case, the results may be a case of selection bias: users who are not agreeable opt out of the story. Second, the story itself may be priming users' expectations of the conversational agent's capabilities. The story uses a fixed dialogue that does not adjust based on the user's response. These types of responses may very well lower a user's expectations of the conversational agent's capabilities.

Taking these preliminary personality story results into consideration in conjunction with the differences in ratings based on Table 4, it appears that users who agree to hear the story have a tendency to give a higher rating to the overall conversation than those who say no. Further, of those who agree to hear the story, the more agreeable a user is, the more likely they are to provide a higher rating for their conversational score. Future research needs to examine whether or not this alone (e.g., listening to a conversational agent's story) is sufficient to predict a user's rating and can be used to further personalize the conversational agent's interactions with users.

References

1. Astrid M, Krämer NC, Gratch J (2010) How our personality shapes our interactions with virtual characters-implications for research and development. In: International conference on intelligent virtual agents. Springer, pp 208–221
2. Cuperman R, Ickes W (2009) Big five predictors of behavior and perceptions in initial dyadic interactions: personality similarity helps extraverts and introverts, but hurts "disagreeables". J Pers Soc Psychol 97(4):667
3. Edwards AL (1957) The social desirability variable in personality assessment and research
4. Garrett P (2010) Attitudes to language. Cambridge University Press
5. Goldberg LR (1992) The development of markers for the big-five factor structure. Psychol Assess 4(1):26
6. Goldberg LR et al (1999) A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models. Pers Psychol Eur 7(1):7–28
7. Goldberg LR, Johnson JA, Eber HW, Hogan R, Ashton MC, Cloninger CR, Gough HG (2006) The international personality item pool and the future of public-domain personality measures. J Res Pers 40(1):84–96

8. Ivanov AV, Riccardi G, Sporka AJ, Franc J (2011) Recognition of personality traits from human spoken conversations. In: Twelfth annual conference of the international speech communication association
9. John OP, Srivastava S et al (1999) The Big-Five trait taxonomy: history, measurement, and theoretical perspectives, vol 2. University of California Berkeley
10. Mairesse F, Walker M (2006) Automatic recognition of personality in conversation. In: Proceedings of the human language technology conference of the NAACL, companion volume: short papers, pp 85–88
11. Mairesse F, Walker MA, Mehl MR, Moore RK (2007) Using linguistic cues for the automatic recognition of personality in conversation and text. J Artif Intell Res 30:457–500
12. McCrae RR, Costa PT (1989) The structure of interpersonal traits: wiggins's circumplex and the five-factor model. J Pers Soc Psychol 56(4):586
13. Rissola EA, Bahrainian SA, Crestani F (2019) Personality recognition in conversations using capsule neural networks. In: IEEE/WIC/ACM international conference on web intelligence, pp 180–187
14. Vinciarelli A, Mohammadi G (2014) A survey of personality computing. IEEE Trans Affect Comput 5(3):273–291

Dialogue Modelling and Applications

A WoZ Study for an Incremental Proficiency Scoring Interview Agent Eliciting Ratable Samples

Mao Saeki, Weronika Demkow, Tetsunori Kobayashi, and Yoichi Matsuyama

Abstract To assess the conversational proficiency of language learners, it is essential to samples that are representative of the learner's full linguistic ability. This is realized through the adjustment of oral interview questions to the learner's perceived proficiency level. An automatic system eliciting ratable samples must incrementally predict the approximate proficiency from a few turns of dialog and employ an adaptable question generation strategy according to this prediction. This study investigates the feasibility of such incremental adjustment of oral interview question difficulty during the interaction between a virtual agent and learner. First, we create an interview scenario with questions designed for different levels of proficiency and collect interview data using a Wizard-of-Oz virtual agent. Next, we build an incremental scoring model and analyze the accuracy. Finally, we discuss the future direction of automated adaptive interview system design.

1 Introduction

With a growing demand for language education, there is much need for the automation of assessment for linguistic proficiency. An easily accessible assessment would allow for the monitoring of each individual student's progress and facilitate the tailoring of curriculum for a more effective learning. Although much research has been done on the automatic assessment of written texts and monologues, the valuation of dialogic speech in conversational settings—or oral proficiency—still heavily relies on human-led interviews [1]. Not only are human-led interviews costly, it has been pointed out that behavioral differences among interviewers can lead to unwanted variation in test ratings [2].

Given the consistent behavior of dialogue systems, there have been recent attempts on using them for automated oral proficiency assessment [3–6]. However all studies use a fixed task difficulty throughout the interaction. This is problematic because

M. Saeki (✉) · W. Demkow · T. Kobayashi · Y. Matsuyama
Waseda University, Tokyo, Japan
e-mail: saeki@pcl.cs.waseda.ac.jp

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_13

test takers are composed of highly varying levels of proficiency, and unless they are matched with tasks appropriate to their level, a test may fail to accurately measure their language skill.

To provide tasks with an appropriate level of difficulty, it is necessary to assess test-takers' proficiency incrementally. In this paper, we investigate the feasibility of such incremental assessment. To this end, we first designed an adaptive interview using Wizard-of-Oz (WoZ) system and collected 56 interviews of English learners, scored by human raters. We then used a recurrent neural network (RNN) model to incrementally score the learner at different stages of the interview. To the best of our knowledge, this is the first work on the incremental assessment of oral proficiency in dialogic settings. We demonstrate high agreement to human raters as the validity evidence of our system, promoting the progress for adaptive oral proficiency tests. The rest of the paper is organized as follows. Section 2 reviews previous work on oral proficiency interview frameworks and automated assessments. Section 3 explains the design of the interview test, the development of the WoZ system, and the data collection process. Section 4 explains the incremental assessment model. Section 5 reports on findings from the data collection, the performance of the incremental scoring model and discusses the results. Finally, Sect. 6 draws conclusions.

2 Related Work

Oral proficiency interviews have long been examined to create fair and reliable tests. One notable framework is the Oral Proficiency Interview (OPI), developed by the American Council on the Teaching of Foreign Languages (ACTFL) [1]. The ACTFL-OPI begins with a "Warm-up", where the examiner eases the candidate into the test by asking questions and making small talk. Through the warm-up, the interviewer makes a brief, or preliminary, assessment of the candidate's proficiency level. The next two stages are part of a crucial "iterative process" in which the examiner alternates between a comfortable and challenging difficulty, in order to provoke loss of linguistic control. Such loss is known as the "signs of breakdown" and may include hesitation, false starts, a lack of response, or self-correction. The iterative process is repeated and re-adjusted until sufficient information is gathered to correctly assess the difficulty level at which the speaker experiences breakdown.

To date, only a few studies have used dialogue systems for oral proficiency scoring. The ACTFL Oral Proficiency—computer is a commercially available test which uses a virtual agent for a simulated interview [6]. The interview is simulated in the sense that all system utterances are generated regardless of the user's previous utterance. A self-assessment made prior to the interview is used to adjust the question difficulty, but no adjustments are made during the interview itself. Reference [4] used off-the-shelf dialogue systems to have users participate in a task-based conversation. The interaction was scored automatically using a model for non-interactive speech based on Gaussian process. Reference [5] also collected task-based conversations and scored the interaction aspect using RNN. Other work on dialogue scoring has used RNNs to capture the multi-turn nature of a dialogue, as well as the fusing different modalities. Reference [7] for example, fused features representing the content, delivery and language use, while [8] tried to incorporate visual cues for scoring. The process of narrowing down user level through incremental assessment and question selection (as featured in the ACTFL-OPI) is key to a reliable test. However, no automated assessment has done it so far.

3 Data Collection

3.1 Experimental Design

Since existing interview frameworks are not directly applicable to dialogue systems due to technical limitations, we designed our own task based on the ACTFL-OPI. Our adaptive oral proficiency interview consists of several topics that are set around a main question and proceeded by follow-up questions. The follow-up questions concerned the same topic as the main question and served to elicit additional speech sample.

The interview begins with a warm-up, during which all candidates are questioned on the same topic. A preliminary assessment made during this stage is used to branch candidates into three levels of proficiency. The proficiency scale used is based on the Common European Framework of Reference for Language (CEFR) [9]. Each of these three levels of proficiency has a pre-prepared subject for discussion as well as corresponding questions. At the closure of each topic, the candidate is re-assessed to attune task difficulty. If the candidate either falls behind or goes beyond the criteria for a certain level, they are moved to the respective branch.

3.2 WoZ Interview System

We developed the Intelligent Language Learning Assistant (InteLLA), our virtual agent leading the oral proficiency interview for the data collection. The agent is rendered using the Unity game engine,[1] and its motion can be controlled through a list of pre-recorded body movements and vocal responses in WoZ style. The use of WoZ system allows for the collection of human-agent interaction data without the need to build an automated system, making a rapid feasibility study possible. Through an initial study of human-led interviews, 76 utterances and 4 non-verbal behaviors were selected as the options for the Wizard. The utterances include a greeting, instructions, main questions, follow-up questions, and feedback. For the non-verbal behaviors, we included nodding, smiling, and surprise, all to be used for

[1] https://unity.com/.

active listening. Speech was generated using Text-to-Speech, and the agents motions were recorded using motion and facial expression capture technology.

The agent was controlled by two operators. A main operator managed the utterances, while a sub-operator selected appropriate non-verbal behaviors. The reason for needing two separate operators, is the heavy cognitive and operational load that proved to be too intensive for a single operator. More specifically, the selection of appropriate utterances requires careful listening of the content of a user's speech, which must also be used to make a quick estimation of their language proficiency. This process is done alongside the giving of non-verbal feedback, which in turn requires the monitoring of phonological and visual cues.

3.3 Interview Data Collection and Human Assessment

With the use of InteLLA, we collected interviews from 56 Japanese English-learners. All test subjects were university students with varying levels of English proficiency. Each student discussed 7 different interview topics. The interview was conducted online using the video conferencing tool, Zoom. Though online conversations are different in nature to face-to-face discourse, studies show that speaking assessment in the two modes show similar results [10]. All test users completed the interview remotely, which lasted 9 min on average. After the interview, users were asked to evaluated the interview using a 5-point Likert scale questionnaire to obtain subjective evaluation on firstly, how well the system was able to adapt to each user, and secondly, how well the system was able to measure the user's ability. The former is measured by how appropriate the user thought the question difficulties were, and the latter by whether the user was able to demonstrate their language abilities to the full extent. The reasons behind their evaluation were also collected through free-form questionnaires. Figure 1 shows a screenshot of the interview data. Figure 2 shows the results of the questionnaire, which will be discussed in Sect. 5.

Each interview was scored by human raters using the CEFR scale. We adopted the scale for "communicative language competence", consisting of the standard 6 levels: A1, A2, B1, B2, C1, and C2. A1 represents the lowest proficiency, and C2 represents the highest. The whole dataset was annotated by a single rater with extensive experience in CEFR grading. Since only two students were in the C1 and C2 bandwidth, we excluded them from further analysis. To measure the inter-rater agreement, we asked another rater to annotate a subset of the dataset with 20 students. The inter-rater agreement calculated using quadratic weighted κ (QWκ) was 0.753.

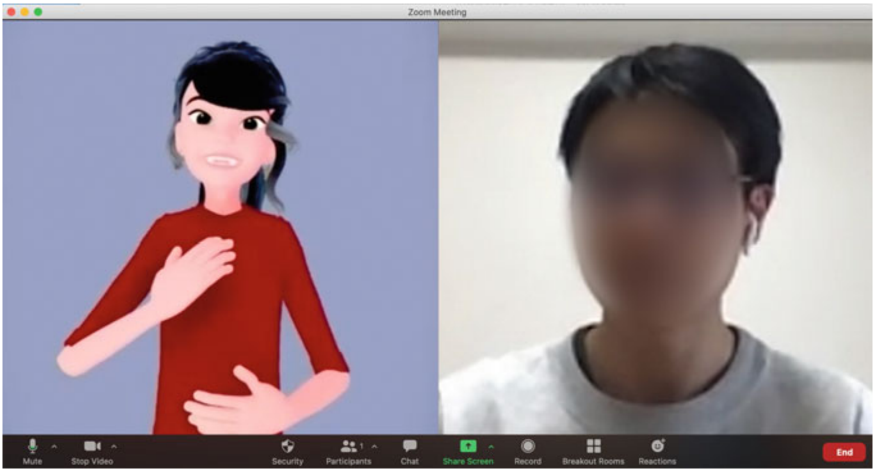


Fig. 1 InteLLA (left) is interviewing with a participant remotely (right) via a video conferencing tool

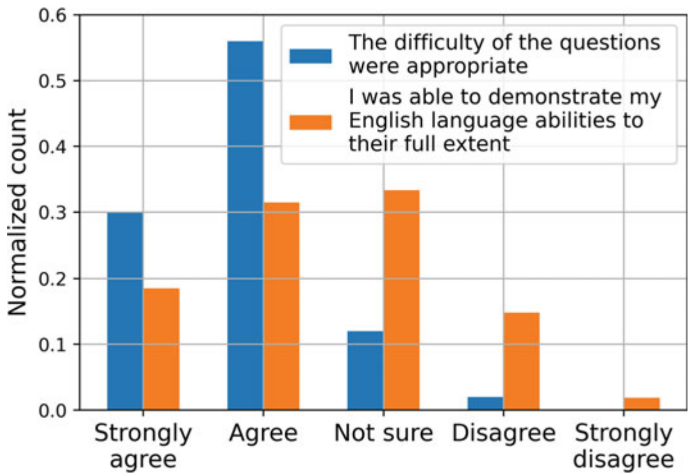


Fig. 2 Subjective evaluation of the WoZ interview on how well the system was able to adapt to each user, and how well the system was able to measure the user's ability

4 Incremental Prediction Model

During the data collection discussed in Sect. 3.3, the operator focused on speech sample elicited in each subsequent topic and updated their assessment based on the observed quality of speech. To capture this incremental decision-making process, we used a LSTM neural network. The input features were chosen from previous studies on monologue and dialogue scoring [11] as well as through the analysis of

Table 1 List of features for incremental proficiency scoring

Feature name	Description
Response length	The length of response for each turn in number of words and seconds
Word n-gram	Number of unique 1-gram, 2-gram and 3-gram used
Word level	The mean difficulty level of words used
Speech rate	Number of syllables per seconds
Pause frequency	Frequency of pauses
Transition time	The length of time between the end of the system's utterance and the beginning of the user's speech
Discourse marker	Number of discourse markers

the annotation process. Features cover aspects such as vocabulary level, fluency, and coherence, and a complete list is shown in Table 1. The difficulty of the word used by the student is calculated using the CEFR-J Wordlist [12, 13]. The model was trained after the completion of each topic. For the respective topic labels, we used the score assigned to a user's interview as a whole. A fivefold cross-validation was conducted, with the Adam optimization algorithm [14] used to minimize the mean squared error (MSE) loss function over the training data. We oversampled the minority classes to address their imbalance and trained each fold for 40 epochs.

5 Results and Discussions

The incremental scoring model was evaluated using accuracy and QWκ after each of the 7 interview topics. The human scores were discrete values, while the model predictions were continuous. We therefore rounded the model prediction for evaluation. Figure 3 shows the mean result over 5 runs with an error band. The confusion matrix of the prediction after topic 2 and 7 is shown in Fig. 4. The accuracy and correlation was still low after topic 2 (the warm-up), but most predictions were made within one level of error. The accuracy and QWκ increased after each topic, saturating around the fourth topic. Correlation with human scoring exceeded that of the human-human agreement at this point. The highest accuracy and QWκ was achieved at topic 7 (the final prediction), being 0.604 and 0.784 respectively. These are encouraging results because they show that the model can capture the approximate level of proficiency of a user with only a few turns of dialogue. This estimation can then be used for a better adjustment of the task and in turn, a better assessment. One limitation of this study is the lack of advanced level English speakers (C1 and C2 CEFR level). Although beginner to intermediate students are the most common proficiency for

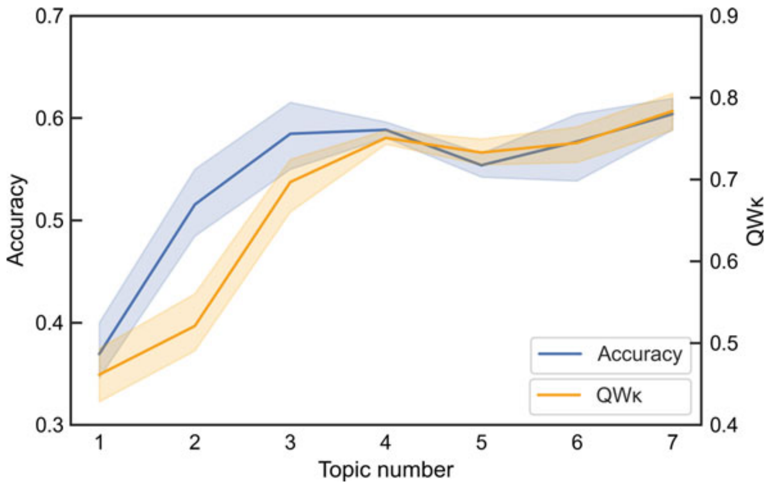


Fig. 3 Accuracy and quadratic weighted κ for each interview topic

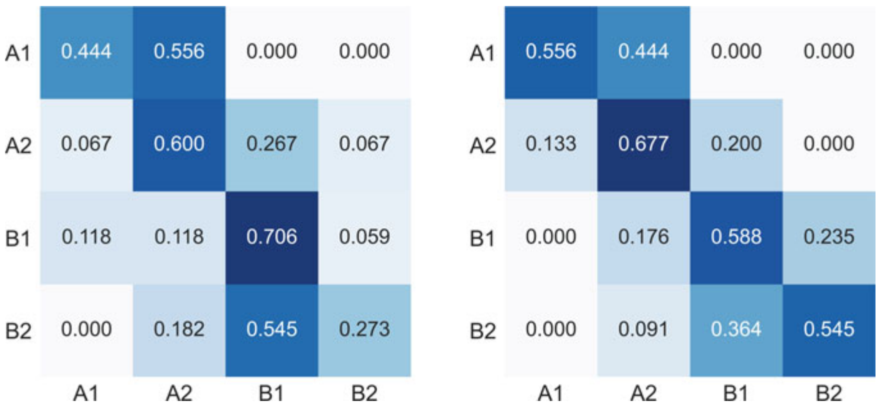


Fig. 4 Normalized Confusion Matrix for incremental prediction after topic 2 (left) and 7 (right)

Japanese English-learners, we would like to evaluate our model on advanced level speakers in future studies.

Finally, we will discuss the subjective evaluation of the whole interview, shown in Fig. 2. 80% of the users found the question difficulty appropriate, which is unsurprising given that human operators were actively adjusting them. Nevertheless, this percentage assures the validity of our test design. On the other hand, only 50% of the users agreed that they were able to demonstrate their English ability to its full potential. We identified two key reasons behind these results from our open-questionnaire. These were a lack of adequate active listening strategies from the system, and a lack of sufficient topic development through follow-up questions. Although this work has not focused on dynamic content generation, such functionality is important for users to better demonstrate their language abilities. Active listening and question generation strategies have previously been studied for job interview systems [15, 16], and the implementation of such strategies will be considered in future studies.

6 Conclusion

This paper has investigated the feasibility of incremental assessment of oral proficiency using an adaptive test format. First, we designed our own interview protocol for an automated adaptive testing and built a WoZ system to serve as the interviewer. Using the WoZ system, we collected an interview dataset of 56 English learners, annotated using the CEFR scale—an international standard for language proficiency evaluation. We then built a LSTM based incremental assessment model that updates its prediction every few turns of the dialogue. Results showed a moderate agreement with human scoring throughout the beginning of the interview, which increased over time, and finally surpassed human inter-rater agreement. Encouraged by this result, our future direction will be to include the incremental scoring model into dialogue systems for a fully automated adaptive oral proficiency test.

Acknowledgements This paper is based on results obtained from a project, JPNP20006 ("Online Language Learning AI Assistant that Grows with People"), subsidized by the New Energy and Industrial Technology Development Organization (NEDO).

References

1. Liskin-Gasparro JE (2003) The ACTFL proficiency guidelines and the oral proficiency interview: a brief history and analysis of their survival. Foreign Lang Ann 36(4):483–490
2. Brown A (2003) Interviewer variation and the co-construction of speaking proficiency. Lang Test 20(1):1–25
3. Evanini K, Singh S, Loukina A, Wang X, Lee CM (2015) Content-based automated assessment of non-native spoken language proficiency in a simulated conversation. In: Machine learning for SLU & interaction, pp 1–7
4. Litman D, Young S, Gales M, Knill K, Ottewell K, Van Dalen R, Vandyke D (2016) Towards using conversations with spoken dialogue systems in the automated assessment of non-native speakers of English. In: Proceedings of SIGdial, pp 270–275
5. Ramanarayanan V, Mulholland M, Qian Y (2019) Scoring interactional aspects of human-machine dialog for language learning and assessment using text features. In: Proceedings of, pp 103–109
6. Dhonau S (2020) ACTFL oral proficiency—computer. Technical report
7. Neural approaches to automated speech scoring of monologue and dialogue responses. In: ICASSP, pp 8112–8116
8. Saeki M, Matsuyama Y, Kobashikawa S, Ogawa T, Kobayashi T (2021) Analysis of multimodal features for speaking proficiency scoring in an interview dialogue. In: 2021 IEEE spoken language technology workshop, SLT 2021—proceedings, pp 629–635
9. Council of Europe (2018) Common European framework of reference for languages: learning, teaching assessment. Cambridge University Press, Cambridge

10. Nakatsuhara F, Inoue C, Berry V, Galaczi E (2017) Exploring the use of video-conferencing technology in the assessment of spoken language: a mixed-methods study. Lang Assess Q 14(1):1–18
11. Zechner K, Evanini K (2020) Automated speaking assessment: using language technologies to score spontaneous speech. Routledge
12. Tono Y (2021) The CRFR-j wordlist version 1.6, tokyo university of foreign studies. http://www.cefr-j.org/index.html/. Accessed 1 June 2021
13. Negishi M, Takada T, Tono Y (2013) A progress report on the development of the CEFR-j. In: Exploring language frameworks: proceedings of the ALTE Kraków conference, pp 135–163
14. Kingma DP, Ba JL (2015) Adam: a method for stochastic optimization. In: Proceedings of ICLR, pp 1–15
15. Inoue K, Hara K, Lala D, Nakamura S, Takanashi K, Kawahara T (2019) A job interview dialogue system with autonomous android ERICA. In: Proceedings of IWSDS, pp 1–6
16. Inoue K, Lala D, Yamamoto K, Nakamura S, Takanashi K, Kawahara T (2020) An attentive listening system with android ERICA: comparison of autonomous and WOZ interactions. In: Proceedings of SIGdial, pp 118–127

SUPPLE: A Dialogue Management Approach Based on Conversation Patterns

Florian Kunneman and Koen Hindriks

Abstract We propose SUPPLE, a new class of dialogue management systems that takes the core concept of a *dialogue sequence* as its main starting point. SUPPLE is inspired by the *conversation patterns* from the Natural Conversation Framework (NCF). While NCF primarily provides a design framework, we propose to automate the selection and updating of dialogue sequences as a central component of the dialogue management module, enabling the dialogue system to build a hierarchical dialogue structure at run-time. The conversation patterns are combined with the key concepts of *update strategies* and *agenda* adopted from the Information State Update approach. We formally describe the building blocks of our approach, and show how dialogue competencies like sequence expansion and slot-filling can be performed in our approach. These are further illustrated in a cooking assistant scenario.

1 Introduction

The success of task-based dialogue management (DM) approaches can be measured by their capacity to fulfill certain domain-specific tasks by conversing in a natural dialogue with the end user. In line with the first aim of task fulfillment, the dialogue state in task-based DM is commonly represented as the information that needs to be collected from the user in order to fulfill its goal, which in turn informs the dialogue policy for deciding what the agent will say next. This framework is seen, for example, in several recent probabilistic approaches to task-based DM, like sequence-to-sequence models [1] and Partially Observable Markov Decision Processes (POMDPs) [2], with shared corpora to improve on Dialogue State Tracking [3–5]. These approaches have improved agent capacities to exchange the

F. Kunneman (✉) · K. Hindriks
Social AI, Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
e-mail: f.a.kunneman@vu.nl

K. Hindriks
e-mail: k.v.hindriks@vu.nl

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_14

right information for handling user requests. However, with respect to the second aim of natural dialogue, there is little alignment with how humans sequentially organize a dialogue, as studied in the field of Conversation Analysis [6]. For example, question-answer adjacency pairs [7] may be expanded by remarking on the question or asking a counter-question, which is not represented in the dialogue policy of these approaches.

The use of such insights is more common in handcrafted approaches to dialogue management, where the designer typically uses commercial interfaces like Google Dialogflow or Watson Assistant for implementing a conversational agent. This is achieved by designing conversation flows which are handled by a frame-based DM approach [8]. Most concretely, [9] have recognized the value of Conversation Analysis for improving human-agent conversations, and made it a central part of their Natural Conversation Framework (NCF) which informs the practice of conversational UX design. The designer specifies and labels conversational patterns as sequences of dialogue moves that are instrumental to perform conversational activities like extended telling and open inquiries, which guide the implementation of conversation flows.

Although NCF helps to achieve the aim of natural dialogue next to task fulfillment, the implementation of extended patterns can be a cumbersome process in available commercial interfaces, and the underlying DM is not primarily catered for enabling the emergence of conversational patterns, nor do they disclose any information about their dialogue policy in light of these patterns. As an alternative, we propose SUPPLE (Sequence-Update Pattern-Based Processing with Logical Expansions), a new DM approach that combines the strength of Information State Update (ISU) [10] and NCF—managing conversations in a natural way using patterns and sequence expansion as the core mechanism for update strategies, combined with agenda-keeping capacities.

We formally specify a class of pattern-based DM systems. This formal approach not only explicitly defines the update and generation mechanisms of these systems but also allows us to prove the properties of these systems. The pattern-based approach introduced in this paper offers the following benefits:

- SUPPLE comprises a small and fixed set of intuitive rules that cover most sequential operations and need not be updated for different domains.
- SUPPLE is fed by a collection of conversational patterns, generated by a designer or automatically extracted from a corpus, which may be stored in a pattern library for re-usability.
- A dialogue tree is constructed while the conversation is ongoing, which makes for a flexible and interpretable dialogue policy by keeping track of how dialogue acts fit in this overall sequential structure of the conversation.

We illustrate the pattern-based mechanisms introduced in this paper using the domain of recipe instruction as our running example.

2 Related Work

2.1 Dialogue Management

Dialogue management approaches can roughly be categorized as handcrafted, probabilistic, and a hybrid of the two [11], which reflects the input from which dialogue managers learn to conduct conversations. Our proposed approach aligns most with model-based approaches to DM, like ISU [10] and Hierarchical Task Networks (HTN) [12]. ISU was originally proposed by [13] as a way to specify how dialogue moves can trigger a subsequent agent move in a structured conversation. The information state keeps track of all aspects relevant to the conversation, such as its history, the agent's knowledge and beliefs, and the agenda guiding the agent's conversational targets. The pattern-based approach introduced here can be viewed as an instantiation of ISU by two comprehensive sets of rules for updating the sequential conversation structure and generating moves by using patterns to inform the agent's dialogue act choice. These rules apply to general conversational capacities like sequence expansion and interruption, and are domain-independent. This is in contrast to common implementations of ISU such as [14], who had domain experts define dialogue policy rules to be applied to health coaching conversations and concluded that it was a hard task for them, with several iterations to come to the dialogue policy required for the domain.

The RavenClaw framework [12] enables the dialogue designer to specify a Hierarchical Task Network (HTN) for the interaction, while domain-independent conversation skills like error-handling are accounted for in separate modules. The dialogue is focused on task completion, where the hierarchical structure of the tree represents the dialogue context. The user can take initiative and shift focus to another task, and upon completion the agent will return to the previous task. Like RavenClaw, SUPPLE incorporates a planning component to structure the dialogue, where the user can take initiative and shift the focus. The agenda of tasks in our approach, however, is given in the form of conversational patterns to follow, enabling more concrete control over the natural dialogue that is conducted to accomplish certain tasks and facilitating user initiative in a natural way in the form of sequence expansion.

Reference [15] combines HTNs with dialogue trees for specifying the conversational paths that may be followed at each task node. Dialogue trees, however, are known for the effort that needs to be spent to account for the many directions a dialogue may take. In addition to the high-level HTN structure, [15] address this problem by means of partially automated dialogue generation. In comparison, the conversational patterns that are inputted to SUPPLE will automatically be assembled into a dialogue tree as the conversation unfolds. The way in which these patterns can follow one another can be controlled and thus does not need to be specified in detail as this is coordinated in the dialogue management module by means of a fixed set of update rules.

Combinatorial explosion is a common threat when the order of utterances in a dialogue can become flexible, as is often the case in mixed-initiative systems. A key insight we derive from [16] is that the size of patterns in the dialogue management framework should be kept small. Recently, [17] combined ontology-driven dialogue management with dialogue workflow graph specification, to increase the re-usability of dialogue sequence design. A sequence can be re-used at different levels of domain-specificity, which also applies to our approach.

3 The SUPPLE Dialogue Approach

SUPPLE is based on the core concepts of a *conversational pattern* (Sect. 3.1) for modeling dialogue structure (cf. [9]). In SUPPLE, the *information state* consists of the dialogue session history (Sect. 3.2), the *agenda* (Sect. 3.3) is a list of patterns, and *update rules* are used to implement various sequence update mechanisms (Sect. 3.4), and mechanisms for dialogue move selection (Sect. 3.5).[1]

3.1 Conversational Patterns

NCF proposes a pattern language and is based on insights from Conversation Analysis [6], which grounds our work in the analysis of human conversation. Patterns established by conversation analysis aim at identifying the sequential expectations that are raised and oriented by participants such as that inquiries are typically followed by an answer [7]. The work on the pattern language of [9] is focused on platforms that support the Intent-Entity-Context-Response paradigm. For the purposes of this paper, we will assume a similar approach to natural language understanding which maps user utterances to predefined intents (or move types) and entities.

Figure 1 provides an illustration of a simple pattern, where a question first is posed that is followed by an answer and a sequence closer. Reference [9] describes many of such patterns, categorized into Conversational Activity, Sequence-level Management, and Conversation Management UX patterns. Conversational patterns inform an agent about the direction that a conversation may follow, while offering the flexibility to follow different paths and allowing for turns that switch from one pattern to another. They are deliberately kept short to enable sequence expansion, e.g., when a question is not completely understood and a paraphrase is given instead of an answer, which may in turn be confirmed after which an answer is given.

We can define patterns in terms of the actions that are performed. Actions can be dialogue moves or other types of actions for conversation or agenda management, e.g., repeating a sub-pattern (see Sect. 3.3 below). A *dialogue move* (or *act*) is a triple $agt{:}\tau(x = k, y = l, \ldots)$, where agt denotes either the agent A or the user U, τ the *type* (or *intent*) of the move, and $(x = k, y = l, \ldots)$ is a list of slot-value

[1] See https://socialrobotics.atlassian.net/wiki/spaces/SUP/overview for an up-to-date implementation of SUPPLE.

Fig. 1 Example of the conversational pattern mark-up (A = agent, U = user) from [9]

Pattern A1.0	**Inquiry (User)**
	1 U: INQUIRY 2 A: ANSWER 3 U: SEQUENCE CLOSER
Example	
	1 U: **What was the name of the first chatbot?** 2 A: Her name was ELIZA 3 U: **Ok**

pairs (cf. [18]). A *partial* move is a move with unknown values, indicated by ?, e.g., $agt{:}\tau(x = k, y = ?, ...)$, where the value of slot y is unknown. A *plain* dialogue move is a pair $agt{:}\tau$ without any slots attached to the move.

Patterns specify the order in which actions are expected and the agent that is expected to generate each move. To be precise, a *pattern* is a *named list* $\langle \ell, a_1, \ldots, a_n \rangle$, where ℓ is an identifier—the pattern's name, and each a_i is either a plain dialogue move $agt_i{:}\tau_i$, or an agenda or conversation management action. For the purposes of this paper, we keep things simple, but actions a_i could also be templates which would match with various kinds of dialogue moves. Patterns cannot be empty, and we assume a minimal length of $n \geq 2$ to ensure a pattern records at least one action and a follow-up (response). For example, the pattern $\langle recipe, A{:}confirm, A{:}repeat(step), A{:}last \rangle$ instantiates an extended telling pattern used for recipe instruction where $step$ is a sub-pattern that is repeatedly deployed until all instruction steps of a recipe have been completed. Multiple *variants* of a pattern with the same name but with different actions and lengths can be specified, to provide for more flexible user interaction. As an example, a simple opening pattern would consist of greetings only while a longer variant might include welfare checks (wfc), e.g., $\langle opening, U{:}greet, A{:}greet., U{:}wfc, A{:}report \rangle$. Patterns are stored in an agent's knowledge base.

3.2 Session History

A *session history*, or simply *session*, represents the history of the dialogue up to and including the last dialogue act that was performed. A session is not simply a list of such acts but represents the hierarchical structure of the dialogue generated by the patterns that have been used in the conversation.

To formally define a session, we introduce the notion of a sequence. A *sequence* $\langle \ell, t_1, \ldots, t_n \rangle$ is a named list where the *name* ℓ is followed by a (possibly empty) *act list* $\langle t_1, \ldots, t_n \rangle$ where each element t_i is either a dialogue move, other type of act, or a sequence. A sub-sequence s' that is an element of another sequence s is said to be an *expansion* of s and is also called a sub-dialogue. A *plain* sequence is a named list where all elements t_i are plain dialogue moves (without slot-value pairs) or other types of acts; the sequence that is the result of removing all sub-dialogues and all slot-value pairs from dialogue moves in the sequence is called its *plain* sub-sequence. Note that all patterns are plain sequences but not all sequences are plain. A session is a special type of sequence named *root* as it can be viewed as the root of a dialogue tree that is dynamically constructed by a dialogue system. The session consists of the (in)complete set of (sub)sequences which represent the progress made in the conversation thus far.

A session history like a dialogue tree [15] or an HTN [12] can be viewed as a tree structure which represents the relation between sequences and their sub-sequences: when an active sequence is expanded with a sub-sequence, a new sub-dialogue is initiated which can also be viewed as a sub-tree of the overall session. The main difference, however, is that a pattern-based approach does not require the upfront specification of a tree-like structure, as the tree structure is dynamically generated while the dialogue is evolving between user and agent. Also note that typical dialogue states that consist of slot-value pairs can be extracted from a session history by extracting the slot-value pairs from all the dialogue moves that are part of the session. In this paper, we do not commit to any particular structure for representing such a dialogue state (e.g., sequence of slot-value pairs, frames).

We introduce several concepts that are important for defining the sequence update mechanisms in Sect. 3.4. First, *extending* a sequence $s = \langle \ell_s, t_1, \ldots, t_n \rangle$ with an act a means appending a to the end of s yielding the extended sequence $\langle \ell_s, t_1, \ldots, t_n, a \rangle$. We call appending a sub-dialogue s' to the end of a sequence *expanding* that sequence. Second, we say that a sequence is *active* or *ongoing* if it is the main session or the last element of a sequence that is active. Active sequences may still be extended or expanded; all other sub-dialogues are *closed*. The active sequence that does not have any active sub-sequences is said to be *in focus*. The sequence that is in focus is the sub-dialogue that was initiated last. Third, a sequence *matches* with a pattern if its plain sub-sequence s' is a prefix of or equal to the act list of that pattern. Matching always is performed with respect to a given knowledge base of patterns. A sequence is *complete* if it matches with the entire act list of a pattern. A sequence can be complete but also be active, i.e., it is still possible to extend it, which is important to be able to handle longer variants of a pattern.

The session history provides a mechanism for keeping track of discourse obligations [19]. It keeps track of the dialogue moves that have been performed which allows the agent to infer if it needs to respond to these moves. Sub-sequences, moreover, act as a stack and are used to keep track of older obligations such as the need to still provide an answer after a clarification expansion of the sequence. Note that discourse obligations need not be explicitly represented but are rather inferred from the session history by matching the dialogue moves that are performed against patterns in a knowledge base. We assume an agent will always comply with the expectations induced by the sequence that is in focus.

Finally, we introduce two functions ε and ι for controlling the key mechanisms of expanding and interrupting a dialogue sequence. Both functions map a pattern name to a set of pattern names. First, we say that a pattern *p can be expanded by p'* if $\ell_{p'} \in \varepsilon(\ell_p)$. A clarification sub-dialogue asking how to perform a step is a typical example of expanding, for example, a recipe instruction step. Second, we say that p *can be interrupted by* p' if $\ell_{p'} \in \iota(\ell_p)$. If a pattern p can interrupt another pattern p', a sub-dialogue based on p' is closed when pattern p is used to initiate a new (sub-)dialogue. An opening pattern, for example, can be interrupted by initiating a new (sub-)dialogue about available recipes (see Index 3 in Fig. 3).

3.3 Agenda and Conversation Management

An agenda is a special type of non-empty sequence $\langle agenda, \ell_1, \ldots, \ell_n \rangle$ with $n \geq 1$ where each of the elements ℓ_i are pattern names. An agenda specifies the overall expectations or 'plan' of an agent for carrying out a task, and thus has a goal-setting function. A conversational agent uses its agenda as a top-level schedule to structure the conversation and generate the next moves in a dialogue (see the Gen-3 rule in Sect. 3.4). This does not mean that an agenda necessarily dictates the order in a dialogue. User initiative can also steer the direction of the dialogue.

Another reason is that the agenda itself can be manipulated by performing various operations on it. So-called agenda management actions can be included in patterns for inserting or removing items from the agenda. Other actions for managing the conversation such as an action to repeat a pattern can also be included. For reasons of space, we do not provide detailed specifications of these actions. These specifications would, moreover, typically also depend on additional knowledge sources for determining, for example, how often a pattern should be repeated (e.g., how often a recipe instruction step pattern needs to repeated would be based on the length of the recipe that is being instructed).

3.4 Dialogue State Tracking

In the SUPPLE dialogue approach, the main task of the dialogue state tracking component is updating the agenda and session, which keeps track of all acts performed. As slot-value pairs can be extracted from a session's dialogue moves, tracking slot-value pairs is implicit in the task of session updating. Below, a denotes the last dialogue (or any other) act received from either the NLU (move performed by user) or the NLG (act performed by agent) component (see also Fig. 2). To support the explanation of the rules below, we display an example dialogue (the start of a recipe instruction dialogue) along with pattern updates in Fig. 3, and the resulting dialogue tree in Fig. 4.

The first rule deals with the case that a sequence can be extended to match a pattern and the act can be said to *contribute* to a conversation pattern. In the example dialogue this rule applies to index 5, where the user answers the agent's inquiry, index 6, where the agent advises on the user request, and index 9, where the user

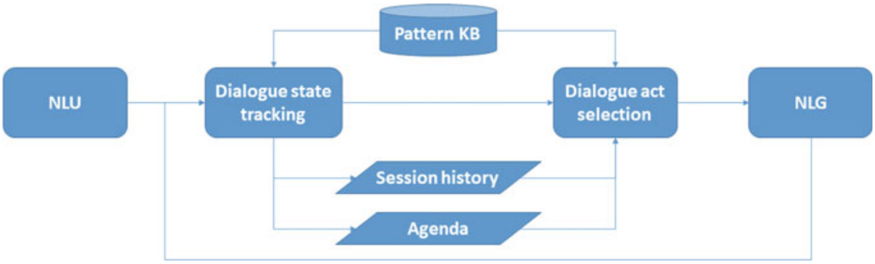


Fig. 2 A SUPPLE dialogue system

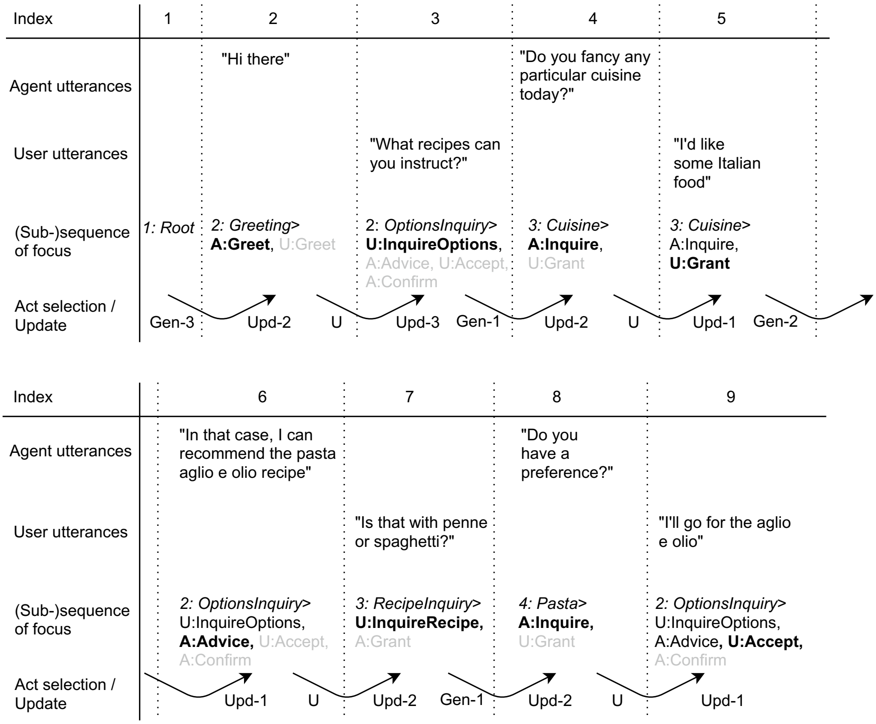


Fig. 3 Example dialogue to explain the mechanisms of SUPPLE. Notation (sub-)sequence of focus: the first number gives the level of the sequence in the tree, the name in italics is the name of the pattern, the moves in bold correspond with the utterance in the same column and the moves in gray mark the expected subsequent moves of the pattern. For presentation purposes, only plain patterns are shown (without any slot-value pairs). The Act Selection/Update gives the selection rule for the next agent act to the left-hand side of the arrow (U for user turns), and update rules to the right-hand side

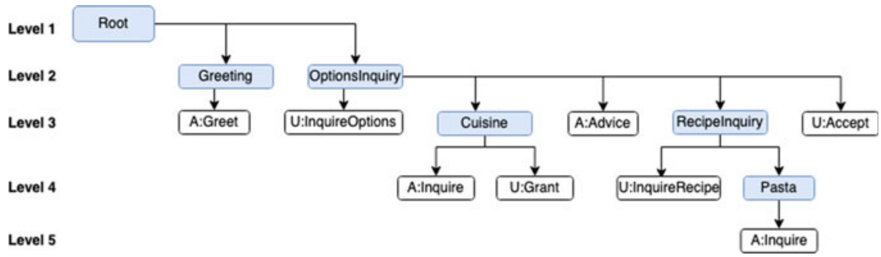


Fig. 4 The dialogue tree that emerges as the example dialogue in Fig. 3 (index 1-9) progresses. Nodes in blue are names of sequences, white nodes starting with 'A:' are agent utterances, and white nodes starting with 'U:' are user utterances

accepts the advised recipe. Note that the instance in index 6 contributes to the pattern in level 2 of the dialogue tree, which was expanded on by the 'cuisine' pattern.

Upd-1: Contribution *If an active sequence s extended with act a matches a pattern p, and all active sub-dialogues of s can be interrupted by p, then extend s with a.*

If act a contributes to the sub-dialogue in focus, which does not have any sub-dialogues of its own (by definition), the contribution rule allows to always extend this sub-dialogue. Another active dialogue sequence s can only be extended by this rule if its sub-dialogues can be interrupted by a pattern that s matches with; by extending s its sub-dialogues are interrupted and closed and the dialogue continues with sub-dialogue s. This kind of update holds for the utterance in index 9, which contributes to the 'OptionsInquiry' pattern that is active in level 2, while the uncompleted pattern of focus is a pattern deeper in the tree (the 'Pasta' pattern as an expansion of the 'RecipeInquiry' pattern). This is a valid contribution to the dialogue, and the sub-sequences are consequently closed. As mentioned at the end of Sect. 3.1, even when an active sequence is completed it may still be the case that an act can contribute to it if there is a longer variant of a matching pattern.

If the first rule does not apply, the next thing to check is whether a can initiate a sub-dialogue and expand an active sequence. An act a may initiate a sub-dialogue if it matches the first act of a pattern p. It can only expand a sequence and initiate a new sub-dialogue if the active sequence allows such an expansion with p, i.e., we have $\ell_p \in \varepsilon(\ell_s)$. Examples of these expansions are index 2, 4 and 8 (agent-initiated) and 7 and 9 (user-initiated) in Fig. 3, represented in Fig. 4 as patterns (blue nodes) that are extending the sequence of focus and form a new sub-sequence one level lower in the tree.

Upd-2: Sequence Expansion *If act a initiates a pattern p named ℓ_p, active sequence s can be expanded with p, and all active sub-dialogues of s can be interrupted by p, then expand s with $\langle \ell_p, a \rangle$.*

Similar to the first update rule, in this case by expanding an active sequence, any other active sub-dialogues of the expanded sequence are interrupted and thereby closed. Also note that this rule allows for expanding the main session sequence, which is needed to get a dialogue going in the first place when the session is initially still empty.

If the dialogue act cannot contribute to nor be used to expand an active sequence, it may still interrupt an ongoing sub-dialogue if the act matches the start of a pattern and can initiate a new sub-dialogue. Interrupting a greeting pattern by initiating an inquiry sub-dialogue is a good example. As before, the sequence that is interrupted is closed (it is no longer active as it is no longer the last element of an active sequence). In the example dialogue, the user interrupts the greeting pattern in index 3 by directly asking about the recipe options instead of greeting back. This is no valid expansion of the greeting pattern, and starts a new sequence at the same level of the tree (see the 'OptionsInquiry' node in Fig. 4). The 'Greeting' pattern is closed and the agent will not expect the user to greet back.

Upd-3: Sequence Interruption *If act a initiates a pattern p named ℓ_p, active sequence s can be interrupted by p, and s is a sub-sequence of active sequence s', then extend s' with $\langle \ell_p, a \rangle$.*

For the case where none of the previous rules apply, a fourth rule is introduced which introduces a (very) generic mechanism for repair of dialogue move types that are out of context. This rule can also handle, for example, so-called 'default fallback' intents or similar classifications of 'unknown' dialogue move types. A move type is out of context if it does not contribute to a sequence nor matches the start of a pattern and thus cannot be used either to initiate a new sequence. For handling this case, a special sequence type named *repair* is introduced. Out-of-context moves trigger an expansion with this special sequence towards the repair of the currently active sequences.

Upd-4: Out-of-context Repair *If there is an active repair sequence $s = \langle repair, \ldots \rangle$, then extend s with a; otherwise, if s is the sequence in focus, expand s with $\langle repair, a \rangle$.*

Any out-of-context dialogue act 'contributes' to a repair sequence. By expanding the sequence that is in focus we make sure that all sequences that are active remain so and can be resumed after the repair sequence is closed. To make sure this happens as soon as a dialogue act is performed that triggers one of the update rules 1–3 (contributes to or can be used to expand or interrupt one of the other active sequences), the repair sequence is interruptible by any other pattern, i.e., $repair \in \iota(\ell_p)$ for all patterns p. As we do not want any sub-dialogues being initiated as part of a repair sequence and only want to interrupt a repair sequence by application of rules 1–3, we assume that $\varepsilon(repair) = \iota(repair) = \emptyset$.

More detailed fine-tuning of (other forms of) repair can also be achieved by patterns and, for example, sequence expansion rule 2. Such fine-tuning would, however, require the specification of patterns for unexpected move types at a particular turn in a dialogue. Rule 4 aims to provide a more generic mechanism.

3.5 Dialogue Act Selection

The dialogue act selection (or policy) component handles turn-taking and determines whether it is the agent's turn. In general, the agent should only take the turn if the user is not expected to perform a dialogue move next or failed to be as informative as the agent needed the user to be. The generation rules introduced below are based on this principle. Although we recognize that there are exceptions to principles like these (as with almost any dialogue principle), we think it provides an adequate starting point for a generic dialogue policy. Moreover, this does not mean that an agent cannot perform multiple dialogue acts in a row, but such moves should be based on patterns in the agent's knowledge base.

The first generation rule allows the agent to take a turn if some values for slot types are still missing. This rule is prioritized because typically task-based agents will need this information to make progress. Missing values can be identified from the session history by checking whether a user made a move without supplying a required value for a slot type or the agent requested such a value thereafter the user made a move but these values are still missing. In that case, an agent will engage in a sub-dialogue to request the user to provide these values. In order to do so, so-called *slot-filling* patterns $\langle x, \ldots \rangle$ named by the slot type x are used.

Gen-1: Slot Filling *If a slot type x is missing, and the agent can initiate a slot-filling pattern named x by performing a, then select a.*

In the example dialogue at index 4, this rule is applied because of the missing value for the 'cuisine' slot type introduced by the recipe query at index 3.

If a sequence is in focus and the agent can contribute to it by performing an act (cf. Upd-1), the agent can select that act. An agent is only allowed to continue the sub-dialogue that is in focus. The main reason for this is that the agent should not prevent the user from making any moves in sub-dialogues that the agent would close by contributing to a sequence that is not in focus.

Gen-2: Pattern-Based Selection *If by performing act a, the agent would contribute to the sequence in focus, then select a.*

In the example dialogue, the agent uses this rule to continue the 'OptionsInquiry' pattern by performing the 'Advice' step at Index 6.

If an agent cannot contribute to the sub-dialogue that is in focus, it should wait for the user to contribute to an incomplete sub-dialogue. However, if all sequences are complete, there are no expectations on user moves and the next best thing the agent can do to ensure progress is by taking the initiative if the next item on the agenda can be initiated by the agent. Recall that an agenda is a list of pattern names. For each of the pattern names in this list, we can check if a (variant) of the pattern has been completed in the session and by doing so find the first pattern that still needs to be completed. If the agent can initiate this pattern, it should do so if all sub-dialogues have been completed.

Gen-3: Agenda-Based Selection *If all sub-sequences of the session are complete, p is the first pattern on the agenda that has not been completed, and act a is the first act of pattern p, then select a.*

In the example dialogue, this rule is applied to initiate the dialogue at Index 1 using the 'Greeting' pattern which is the first pattern on the agenda.

In order to provide a proper repair response to out-of-context moves made by the user, the context in which the user has made such a move may be relevant. By asking for an amount of an ingredient needed while not having completed the recipe selection task in the agenda yet, for example, may trigger a different response than when a recipe has been selected but does not make use of the ingredient. To handle such context issues, a repair response function *response* that maps the session to a response is assumed.

Gen-4: Repair Response *If s is a session with a repair sequence in focus and $response(s) = a$, then select a.*

A response function is a generic solution for providing a repair response mechanism. Concrete solutions can be specified by, for example, mapping all sessions to generic rephrase requests $response(s) = rephrase$, leaving the details of generating text for such a request to the NLG component. We consider the specification of repair responses out of scope of this paper.

3.6 Dialogue Systems

A *dialogue system* is a tuple $\langle s, pkb, agenda, rules, \varepsilon, \iota, response\rangle$, with s a sequence called the session, pkb a knowledge base with a selected (non-empty) set of patterns, $agenda$ an agenda where all patterns mentioned in the agenda also are part of pkb, $rules$ a (sub)set of the update and generation rules, ε and ι the expansion and interruption functions defined over the pattern names present in pkb, and $response$ is a repair response function. Only the session and agenda can change while the other components are static. An *initial* dialogue system is a system with the empty session sequence $s = \langle root\rangle$.

Given our formal definitions, we are now able to show some properties of dialogue systems in general and for specific systems in particular.

Proposition 1 *Agents never initiate repair sequences.*

Proof Observe that agents will only perform acts that contribute to or initiate a pattern by rules Gen-1-3 if there is no repair sequence. Although they may generate acts that trigger the out-of-context repair rule Upd-4, they will only do so to 'contribute' to an already existing repair sequence.

Proposition 2 *Systems without Upd-4 ignore acts that do not contribute to nor initiate patterns.*

Proof Note that without repair rule Upd-4 such acts will not be added to the session. In contrast, systems with Upd-4, Gen-4, and a well-defined response function will respond to any user input.

We define a *minimal slot filling dialogue system* as any system with a pattern knowledge base that consists of the pattern $\langle inquiry, A{:}ask(\mathbf{x} = \mathbf{?}), U{:}answer\rangle$, which also is the only pattern on the agenda, and for each slot type $x_i \in \mathbf{x}$ one slot filling pattern $\langle x_i, A{:}ask(x_i), U{:}answer\rangle$, the rules Upd-1-2 and Gen-1-3, $\varepsilon(root) = \{inquiry, x_1, \ldots, x_n\}$ where ε, ι are otherwise undefined. Because repair rules are not part of these systems, the $response$ function is redundant.

Theorem 3 *In a minimal slot filling system, if no sequences are initiated anymore and all sequences are complete, all slots have been filled.*

Proof (Sketch) In a minimal slot filling system, sequences can only be initiated by an agent who first initiates the pattern on the agenda by Gen-3 and Upd-2 and then initiates slot filling patterns (only) if a slot type is still missing using Gen-1 and Upd-2. If that does not happen anymore and all sequences are complete, by design of the slot-filling patterns all slots must have been filled.

4 Discussion and Conclusion

Our aim has been to present a new class of dialogue systems based on conversation patterns and to show that patterns can provide a powerful approach to dialogue management. To this end, we have proposed the Sequence-Update Pattern-based Processing with Logical Expansions (SUPPLE) dialogue approach by introducing update and generation rules that use patterns to manage and generate a dialogue. We have shown that with only a limited number of these rules and a few additional resources for regulating expansion and interruption of dialogue sequences it is already possible to define a rich class of dialogue systems with capabilities for conversational management, clarification, and slot filling. More research is needed to further explore (variants and various subsets of) these rules and the (properties of) dialogue systems they give rise to. We would also like to investigate the prioritization induced by the order of the rules and more concrete ways of specifying resources for regulating rule selection.

Our work shares some of the same motivations as that of [16, 17]. Conversational patterns are not only a simple and natural tool for specifying dialogue structure but also provide useful building blocks for generating such structures. By providing generic mechanisms for composing patterns into sub-dialogues, we showed that patterns provide a tractable solution to address the combinatorial problem of dialogue management that cannot be addressed, for example, by dialogue trees as a means for specifying dialogue structure. The regulated composition of patterns instead allows for a concise way of specifying large numbers of dialogues that are only restricted by structural rules for managing a dialogue.

Future Work

The cooking assistant dialogue that we used as a working example is only a proof of concept, and empirical validation of our approach is vital to gain insight into how SUPPLE compares to existing DM approaches on aspects like the effort required to set up a conversational agent, as well as the flexibility of the dialogue with the end user. We are currently designing usability studies to this end.

For the purposes of this paper, we focused on how conversational patterns can provide a conceptual framework for dialogue management and did not discuss any learning or probabilistic methods. There are obvious opportunities for applying such methods, however. Patterns can be learned from conversational data as well as specified by conversational designers, and statistical methods can be applied to keep track of the dialogue state (session). Moreover, it would be useful to statistically learn when a move type can be used to expand or interrupt a pattern instead of manually specifying the expansion and interruption functions. Finally, as suggested in the paper, we believe it will be beneficial to explore combining patterns with ontology-based frameworks to enrich, e.g., slot filling capabilities.

References

1. Wen T-H, Vandyke D, Mrksic N, Gasic M, Rojas-Barahona LM, Su P-H, Ultes S, Young S (2016) A network-based end-to-end trainable task-oriented dialogue system. arXiv:1604.04562
2. Young S, Gašić M, Thomson B, Williams JD (2013) Pomdp-based statistical spoken dialog systems: a review. Proc IEEE 101(5):1160–1179
3. Williams JD, Henderson M, Raux A, Thomson B, Black A, Ramachandran D (2014) The dialog state tracking challenge series. AI Mag 35(4):121–124
4. Budzianowski P, Wen T-H, Tseng B-H, Casanueva I, Ultes S, Ramadan O, Gašić M (2018) Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. arXiv:1810.00278
5. Rastogi A, Zang X, Sunkara S, Gupta R, Khaitan P (2020) Towards scalable multi-domain conversational agents: the schema-guided dialogue dataset. Proc AAAI Conf Artif Intell 34:8689–8696
6. Sacks H (1984) Notes on methodology. Struct Soc Action Stud Conversat Anal 21:27
7. Levinson SC (1983) Pragmatics. Cambridge University Press
8. Rosset S, Bennacef S, Lamel L (1999) Design strategies for spoken language dialog systems. In: Sixth European conference on speech communication and technology
9. Moore RJ, Arar R (2019) Conversational UX design: a practitioner's guide to the natural conversation framework. Morgan & Claypool
10. Traum DR, Larsson S (2003) The information state approach to dialogue management, pp 325–353. Springer Netherlands, Dordrecht

11. Harms J-G, Kucherbaev P, Bozzon A, Houben G-J (2018) Approaches for dialog management in conversational agents. IEEE Internet Comput 23(2):13–22
12. Bohus D, Rudnicky AI (2009) The ravenclaw dialog management framework: architecture and systems. Comput Speech Lang 23(3):332–361
13. Poesio M, Cooper R, Larsson S, Matheson C, Traum D (1999) Annotating conversations for information state update. In: Proceedings of Amstelogue 99, 3rd workshop on the semantics and pragmatics of dialogues
14. Morbini F, DeVault D, Sagae K, Gerten J, Nazarian A, Traum D (2014) Flores: a forward looking, reward seeking, dialogue manager. In: Mariani J, Rosset S, Garnier-Rizet M, Devillers L (eds) Natural interaction with robots, knowbots and smartphones. Springer New York, New York, NY, pp 313–325
15. Rich C, Sidner CL (2012) Using collaborative discourse theory to partially automate dialogue tree authoring. In: International conference on intelligent virtual agents, pp 327–340. Springer
16. Perugini S, Buck JW (2016) A language-based model for specifying and staging mixed-initiative dialogs. In: Proceedings of the 8th ACM SIGCHI symposium on engineering interactive computing systems, EICS '16, pp 204–216, New York, NY, USA. Association for Computing Machinery
17. Wessel M, Kalns E, Acharya G, Kathol A (2020) Widening the dialogue workflow modeling bottleneck in ontology-based personal assistants. arXiv:2011.08334
18. Young S, Gašić M, Keizer S, Mairesse F, Schatzmann J, Thomson B, Yu K (2010) The hidden information state model: a practical framework for pomdp-based spoken dialogue management. Comput Speech Lang 24(2):150–174
19. Allen J, Ferguson G, Stent A (2001) An architecture for more realistic conversational systems. In: Proceedings of the 6th international conference on intelligent user interfaces, IUI '01, pp 1–8, New York, NY, USA. Association for Computing Machinery

Dialogue Management as Graph Transformations

Nicholas Thomas Walker, Torbjørn Dahl, and Pierre Lison

Abstract We present ongoing work on a new dialogue management framework using *graphs* as core representation for the current dialogue state. Dialogue management tasks such as state tracking and action selection are framed as sequences of *graph transformations* that repeatedly update this graph based on incoming observations. Those graph transformations are expressed using a graph query language, making it possible to specify all dialogue management operations through a unified, declarative syntax. We argue that graphs are particularly well suited to model the dialogue state of complex, open-ended domains. In contrast to traditional dialogue state representations that are limited to fixed, predefined slots, graphs can naturally express dialogue domains with rich relational structures and variable numbers of entities to track. We describe how dialogue state tracking and action selection can be modelled in such graph-centric view of dialogue management, using either hand-crafted rules or data-driven models. We also briefly discuss how to account for some aspects of dialogue management such as uncertainties, incremental inputs and contextual knowledge. Finally, we describe a proof-of-concept study of this dialogue management framework in a human–robot interaction scenario.

Keywords Dialogue management · Dialogue state tracking · Action selection · Graph databases · Dialogue architectures · Human–robot interaction

N. T. Walker (✉) · P. Lison
Norwegian Computing Center, Postboks 114, Blindern, 0314 Oslo, Norway
e-mail: walker@nr.no

P. Lison
e-mail: plison@nr.no

T. Dahl
Department of Informatics, University of Oslo, Oslo, Norway
e-mail: torbjd@ifi.uio.no

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_15

1 Introduction

Representing, updating and acting upon the current dialogue state is at the core of dialogue management. For task-oriented systems, this dialogue state is often represented by a fixed, predefined list of *slots* to fill [21, 24]. However, this representation in terms of slot–value pairs is difficult to apply to open-ended domains with varying numbers of entities to track. For example, human–robot interaction tasks must often keep track of entities such as locations, persons or tasks to perform. Those entities may vary over time, for instance, the number of persons located in a given room is not known in advance and may change over the course of the interaction. Those entities are also connected with one another through various relations, such as a person being *in* a room or a response being an *answer* to a preceding utterance.

Graphs are well suited to represent such rich relational structures between (abstract or concrete) entities. Graphs and machine learning models operating on those graphs (in particular, *graph neural networks*) are also increasingly popular for dialogue modelling [6, 8, 12] and generally in NLP [25]. However, such approaches generally focus on specific dialogue modelling aspects and eschew the more general question of how to design a full-fledged dialogue manager operating on a dialogue state expressed as a graph. This paper is a first attempt at answering this question. We describe how to (1) encode the dialogue state as a *property graph* and (2) frame dialogue management as sequences of *graph transformations* that iteratively refine this state (and select actions to perform) based on incoming observations.

We also show how these manipulations can be expressed in a graph query language called OpenCypher [10] and executed on a graph database. Using a unified, declarative language for all dialogue operations allows us to clearly separate the domain-specific logic (which graph operations to execute and on the basis of which inputs) from implementation issues related to concurrency and query optimisation (which are handled by the back-end graph database). It also makes it possible to query knowledge graphs (expressing background knowledge) using the same syntax.

The proposed approach aims to accommodate both handcrafted rules and machine learning models. This ability to combine rule-based and data-driven modules is important when operating on rich, graph-based state representations, as the complexity of the resulting state-action space makes it difficult to learn end-to-end models, at least for domains without large amounts of training data readily available.

We start by briefly reviewing related work on graph-based dialogue management (Sect. 2), then sketch our dialogue management approach, with a particular focus on the representation of the dialogue state and formalisation of dialogue management tasks with graph transformations (Sect. 3). Section 4 illustrates this framework with a case study in human–robot interaction, and Sect. 5 concludes.

2 Related Work

The use of graphs—or more generally, relational representations—in dialogue management has been explored in several previous works. Earlier rule-based approaches to dialogue management often relied on rich formalisations of the dialogue state encoding the beliefs, desires and intentions of each conversational partner through logical forms [3, 14, 15]. However, such formalisations are typically limited to high-level symbolic knowledge, thereby leaving out non-symbolic information such as spatio-temporal features from the dialogue state. Incremental approaches to dialogue processing [20, 22] also rely on relational representations to connect incremental units with one another through temporal or semantic links.

Graphs have also been used as part of statistical and neural conversational models. One important instance is the use of probabilistic graphical models such as Bayesian Networks [17, 23]. However, the "relations" defined in such models are limited to conditional probability distributions between random variables, and cannot as such express other, more semantic relationships. Graph neural networks have also been employed for dialogue policy learning of slot-based systems, such as in [6, 7].

A number of papers have also focused on the use of knowledge graphs to improve the quality of dialogue responses. A sequence-to-sequence conversational model relying on graph embeddings derived from a knowledge graph is presented in [13]. In [11], the authors present a knowledge-grounded conversational model that exploits a large knowledge graph to derive more content-rich responses to user queries. The authors of [16, 26] make use of a graph-encoded knowledge base to inform a dialogue system along with dependency parses of sentences. Finally, [19] show how to integrate a graph database (expressed as RDF triples) to a social chatbot.

Graph representations are also a core element of conversational semantic parsing [1, 9], although the graphs are here limited to relations within a given utterance and do not typically cross utterance boundaries—although see [4] for an exception that allows for some semantic relations (references, repairs) across utterances.

The main novelty of the proposed approach is its reliance on a unified graph to track all variables relevant to dialogue management (including, e.g. utterances, speakers, entity mentions, conversational intents, external knowledge, etc.). Dialogue state tracking and action selection are then framed as operations that continuously manipulate this graph to incorporate incoming observations and select new system actions. Those updates are expressed in a declarative language and run on a graph database, making it possible to handle concurrent read-write operations and allow for arbitrarily complex manipulations of the state graph.

3 Approach

As in previous dialogue architectures such as OpenDial [18], the system architecture (illustrated in Fig. 1) revolves around a blackboard design where various software modules continuously listen for changes (insertion, deletion or modification of nodes

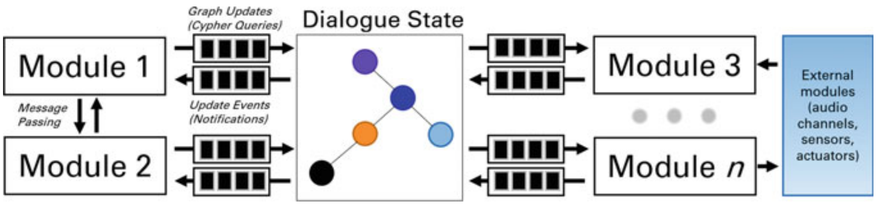


Fig. 1 General system architecture, with the graph database storing the current dialogue state at its centre. Each dialogue management module is notified of new updates to the dialogue state, and may itself submit further updates (expressed as graph queries). Modules may also receive/submit data to one another through message-passing. All modules run in parallel and can both receive new update events to process as well as produce new updates (in the form of graph queries) thereby allowing for asynchronous processing

or edges in the graph) to the central dialogue graph and generates further updates to this graph whenever necessary. All updates to the dialogue state are specified through queries encoded in a graph query language called OpenCypher [10]. The dialogue state itself is stored using an in-memory graph database.[1] The architecture also supports message-passing with ZeroMQ [2] to allow modules to exchange data that are not relevant for dialogue management and do not need to be inserted into the dialogue state.

Modules can be easily plugged in and out of the architecture, and may correspond to handcrafted rules or data-driven models. Note that there is no explicit distinction between dialogue state tracking and action selection—both operations are expressed using the same graph manipulation mechanisms.

3.1 Dialogue State

We model the dialogue state as a *property graph* [10], which is graph structure allowing both nodes and edges to be associated with properties and labels. Property graphs are often contrasted with *triple stores* such as RDF, which cannot directly attach properties to nodes or edges without having to explicitly create new entities.

We require each node and edge in the dialogue state graph to be associated with a semantic label such as `Utterance`, `Speaker`, `Intent` or `Location`. Those nodes may represent observable entities but may also express abstract objects such as a task to execute by the system. The labels attached to each node allow modules to directly filter state updates (for instance, the insertion of a new `Intent` will trigger subsequent updates related to action selection).

The architecture does not impose the use of a particular set of labels. However, the proposed framework does rely on a number of conventions to account for important dialogue modelling aspects:

[1] See http://www.memgraph.com.

Uncertainty: Accounting for uncertainties and partial observability is an important consideration in dialogue systems, especially in domains such as human–robot interaction where observations are often noisy or error-prone. To this end, both nodes and edges may be associated to probability values. In addition, a special `EXCLUSIVE` node is employed to express the fact that some nodes may be mutually exclusive (for instance, between ASR hypotheses associated with a given utterance). Discrete conditional distributions between two random variables can be similarly expressed through special `COND_PROB` edges indicating the conditional probability between two values. This representation can only express a limited form of probabilistic knowledge—in particular, it does not capture continuous distributions or conditional distributions with more than one independent variable. Nevertheless, this representation can express most common forms of uncertainties in dialogue management, such as N-Best lists and the probabilistic outputs of machine learning models.

Temporality: Time is a crucial aspect of dialogue management, in particular, to implement flexible turn-taking strategies. To this end, we treat time as a core component of the graph and associate each node with timestamps expressing its time of creation and its last update. Entities with a duration (such as `Utterances`) also include start and end timestamps. This temporal information makes it possible to (1) explicitly reason over temporal aspects of the interaction and (2) analyse how the dialogue state evolves over time.

Incrementality: As argued in [22], human speakers process dialogues *incrementally*, by gradually refining their interpretation of what is being said (and producing appropriate responses) on the basis of small units of content. To emulate such a behaviour in a dialogue system, one needs the ability to chain together such small units and revise/revoke some of these units whenever necessary. Incremental content can be expressed in our framework through a special `PREVIOUS` relation connecting together consecutive units, and be revoked by deleting the content along with all nodes derived from it.[2]

Contextual knowledge: Finally, dialogue systems often need to access background knowledge to fulfil their tasks. One important benefit of graph-centric dialogue management is the fact that such background knowledge can often be conveniently encoded as a knowledge graph and be queried using the same syntax as other dialogue management operations (as shown in our case study), without requiring ad hoc mechanisms for database access.

[2] This functionality is, however, limited to incremental units with a relatively modest throughput, and is not appropriate for handling high-frequency events, as is the case for streams of audio data.

3.2 Graph Operations

Each dialogue management module listens for notifications of changes in the dialogue state and (when necessary) outputs further updates in the form of graph queries (see Fig. 1). Those updates can be implemented in several ways.

The simplest method is to write a graph query associating a given condition to a state update. For instance, the rule below specifies that, if an utterance mentions an entity named x and our knowledge graph includes a person whose full name starts with x, a `REFERS_TO` edge can be created between the two[3]:

```
MATCH (mention:EntityMention), (person_in_kb:Person)
WHERE person_in_kb.name STARTS WITH mention.name
CREATE (mention)-[:REFERS_TO]->(person_in_kb);
```

Expressing state updates through graph queries allows us to leverage the expressive power of the Cypher language to encode complex graph patterns in an intuitive, human-readable syntax.

Alternatively, one can produce graph updates directly through Python code. Each module has read access to the dialogue state (again through graph queries executed onto the current dialogue state) to extract the inputs necessary for inference, and outputs a list of update queries in the form of `CREATE`, `MERGE`, `SET` or `DELETE` commands. In particular, modules can run machine learning models and integrate their outputs in the dialogue state.

4 Case Study

We used a simple human–robot interaction scenario to showcase how the proposed dialogue management framework can be applied in practice. The robot objective was to function as an automated receptionist, and more specifically (1) answer questions related to the availability of various researchers as well as (2) accompany visitors to a few selected places on the current office floor.

We relied on a knowledge graph storing the calendar data of all researchers to answer questions related to the whereabouts of each person. This knowledge graph includes a range of entities, notably calendar events, employees and meeting locations. Those entities are linked through multiple relations, for instance, a meeting will have a (non-empty) set of participants, a location, as well as a date and a start and end time.

We use a Pepper robot as platform, along with Google Speech for speech recognition and the TTS engine embedded in Pepper. For NLU, we used a neural intent classifier and entity extractor with a pretrained model from Rasa [5] fine-tuned with

[3] For simplicity, we ignore here how to handle ambiguous references with multiple potential targets.

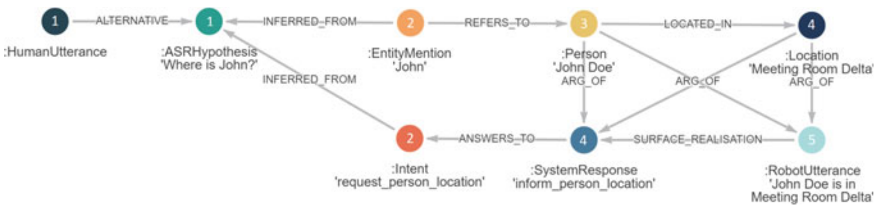


Fig. 2 Illustration of the sequence of graph operations applied upon receiving a new user utterance "Where is John?"
Step 1: The speech recogniser recognises a new utterance and inserts in the dialogue graph one new `HumanUtterance` node (with various information such as timestamps, etc.), along with one `ASRHypothesis` node attached to it.
Step 2: The NLU module is notified of this utterance and classifies it as a new request_person_location intent, along with the person mention "John".
Step 3: The mention "John" is connected to a person entity "John Doe" in the knowledge graph through a simple reference resolution rule (see Sect. 3.2).
Step 4: Another rule detects the presence of a request_person_location intent connected to a person with a known location, and produces inform_person_location as possible response with high utility. This response is then selected by another rule selecting the response with highest utility among possible candidates.
Step 5: This response triggers the creation of a `RobotUtterance` with the `Person` and `Location` as arguments. This utterance is picked up by speech synthesis.

a small list of domain-specific examples. Once a new `Intent` is added to the graph, the rest of the dialogue management process is implemented through graph queries.

A step-by-step example of such process is illustrated in Fig. 2. Due to space constraints, we only provide a high-level description of each update, but the detailed list of graph queries employed to perform each operation is available at https://github.com/NorskRegnesentral/GraphDial.

5 Conclusion

This short paper presented ongoing work on a novel, graph-centric approach to dialogue management in which dialogue state tracking and action selection are viewed as *graph manipulation problems*. The dialogue state is represented as a property graph. Our case study explored the utility and feasibility of a graph-centric dialogue management system in a human–robot interaction setting.

Along with the further development of the system architecture (and its release as an open-source toolkit), future work will concentrate on scaling up the case study with a larger set of intents and system responses, and on conducting a proper evaluation of the resulting platform. We also aim to investigate how to integrate graph neural networks into the architecture, as those types of neural models are ideally suited to exploit the relational structure expressed in the graph of the dialogue state.

References

1. Aghajanyan A, Maillard J, Shrivastava A, Diedrick K, Haeger M, Li H, Mehdad Y, Stoyanov V, Kumar A, Lewis M, Gupta S (2020) Conversational semantic parsing. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Online, pp 5026–5035
2. Akgul F (2013) ZeroMQ. Packt Publishing
3. Allen J, Byron D, Dzikovska M, Ferguson G, Galescu L, Stent A (2000) An architecture for a generic dialogue shell. Nat Lang Eng 6(3–4):213–228
4. Andreas J, Bufe J, Burkett D, Chen C, Clausman J, Crawford J, Crim K, DeLoach J, Dorner L, Eisner J et al (2020) Task-oriented dialogue as dataflow synthesis. Trans Assoc Comput Linguist 8:556–571
5. Bocklisch T, Faulkner J, Pawlowski N, Nichol A (2017) Rasa: open source language understanding and dialogue management. CoRR. arXiv:1712.05181
6. Chen L, Tan B, Long S, Yu K (2018) Structured dialogue policy with graph neural networks. In: Proceedings of 27th international conference on computational linguistics, pp 1257–1268
7. Chen L, Chen Z, Tan B, Long S, Gašić M, Yu K (2019) Agentgraph: Toward universal dialogue management with structured deep reinforcement learning. IEEE/ACM Trans Audio, Speech, Lang Process 27(9):1378–1391
8. Chen L, Lv B, Wang C, Zhu S, Tan B, Yu K (2020) Schema-guided multi-domain dialogue state tracking with graph attention neural networks. Proc AAAI 34:7521–7528
9. Cheng J, Agrawal D, Martínez Alonso H, Bhargava S, Driesen J, Flego F, Kaplan D, Kartsaklis D, Li L, Piraviperumal D, Williams JD, Yu H, Ó Séaghdha D, Johannsen A (2020) Conversational semantic parsing for dialog state tracking. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Online, pp 8107–8117
10. Francis N, Green A, Guagliardo P, Libkin L, Lindaaker T, Marsault V, Plantikow S, Rydberg M, Selmer P, Taylor A (2018) Cypher: an evolving query language for property graphs. In: Proceedings of 2018 international conference on management of data. Association for Computing Machinery, SIGMOD '18, pp 1433–1445
11. Ghazvininejad M, Brockett C, Chang M, Dolan B, Gao J, Yih W, Galley M (2018) A knowledge-grounded neural conversation model. In: Proceedings of 32nd AAAI conference on artificial intelligence, pp 5110–5117
12. Ghosal D, Majumder N, Poria S, Chhaya N, Gelbukh A (2019) DialogueGCN: a graph convolutional neural network for emotion recognition in conversation. In: Proceedings of EMNLP-IJCNLP, pp 154–164
13. He H, Balakrishnan A, Eric M, Liang P (2017) Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In: Proceedings of 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp 1766–1776
14. Jokinen K (2009) Constructive dialogue modelling: speech interaction and rational agents. Wiley-Interscience
15. Larsson S, Traum DR (2000) Information state and dialogue management in the TRINDI dialogue move engine toolkit. Nat Lang Eng 6(3–4):323–340
16. Liao L, Long LH, Ma Y, Lei W, Chua TS (2021) Dialogue state tracking with incremental reasoning. Trans Assoc Comput Linguist 9:557–569
17. Lison P (2015) A hybrid approach to dialogue management based on probabilistic rules. Comput Speech Lang 34(1):232–255
18. Lison P, Kennington C (2016) OpenDial: a toolkit for developing spoken dialogue systems with probabilistic rules. In: Proceedings of 54th annual meeting of the association for computational linguistics (demonstrations), Berlin, Germany, pp 67–72
19. Pichl J, FEE C, Marek P, Konrád J, Matulík M, Šedivỳ J, CIIRC C (2018) Alquist 2.0: Alexa prize socialbot based on sub-dialogue models. Alexa Prize Proceedings
20. Purver M, Eshghi A, Hough J (2011) Incremental semantic construction in a dialogue system. In: Proceedings of the ninth international conference on computational semantics (IWCS 2011)

21. Ren L, Xie K, Chen L, Yu K (2018) Towards universal dialogue state tracking. In: Proceedings of 2018 conference on empirical methods in natural language processing, pp 2780–2786
22. Schlangen D, Skantze G (2011) A general, abstract model of incremental dialogue processing. Dialogue Discourse 2(1):83–111
23. Thomson B, Young S (2010) Bayesian update of dialogue state: a POMDP framework for spoken dialogue systems. Comput Speech Lang 24(4):562
24. Williams JD, Raux A, Henderson M (2016) The dialog state tracking challenge series: a review. Dialogue Discourse 7(3):4–33
25. Wu L, Chen Y, Shen K, Guo X, Gao H, Li S, Pei J, Long B (2021) Graph neural networks for natural language processing: a survey. arXiv:2106.06090
26. Yang S, Zhang R, Erfani S (2020) Graphdialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems. In: Proceedings of 2020 conference on empirical methods in natural language processing, pp 1878–1888

Chatbots and Conversational Agent Technologies

Data Collection for Detecting Unwillingness to Answer Questions in Dialogue

Kazumi Nagao, Ryuichiro Higashinaka, and Kazuto Ataka

Abstract To achieve dialogue agents that are social and can exhibit hospitality, it is important to detect the unwillingness of users to answer questions. However, it is difficult to detect such unwillingness because of the lack of data in which users are having difficulty answering questions, which hinders the creation of machine-learning based classifiers. This paper aims to collect dialogue data in which people are unwilling to answer questions. For this purpose, we created a list of difficult-to-answer questions on the basis of the cross-cultural communication literature and used such a list to collect dialogues in which such questions appear. By this procedure, we successfully collected dialogues in which users feel the difficulty to answer questions. Using the collected data, we also investigated if it is possible to train a machine-learning based classifier of unwillingness to answer questions.

1 Introduction

In order to achieve dialogue systems that are social and can exhibit hospitality, it is important not to offend people by making inappropriate utterances. This situation can occur if, for example, the system asks the users questions that they feel are difficult to answer. If a system can detect with high accuracy when a person is unwilling to answer questions, it will greatly improve the impression of the dialogue system.

K. Nagao
Faculty of Policy Management, Keio University, Tokyo, Japan
e-mail: kazumingo418@keio.jp

R. Higashinaka (✉)
Graduate School of Informatics, Nagoya University, Nagoya, Japan
e-mail: higashinaka@i.nagoya-u.ac.jp

K. Ataka
Faculty of Environment and Information Studies, Keio University, Tokyo, Japan
e-mail: ataka@sfc.keio.ac.jp

Yahoo Japan Corporation, Tokyo, Japan

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_16

Extensive research on recognizing people's impressions in dialogue has been conducted in recent years and various dialogue data have been collected for machine learning purposes. Busso et al. [6] collected facial expressions of basic emotions, and Yamazaki et al. [18] annotated closeness between speakers to help construct a dialogue system for establishing a relationship with the user. A dialogue corpus named "Hazumi" [12] was annotated with labels indicating the extent to which people are willing to talk. However, these studies did not focus on situations in which people are unwilling to answer questions.

Since there has been insufficient research regarding the unwillingness to answer questions, there is not enough data to create a machine learning-based classifier of unwillingness to answer questions. The reason for this is mainly the difficulty of collecting dialogue data in situations where people are unwilling to answer questions, as such situations do not occur spontaneously; indeed, humans are typically cooperative [8] and tend to value politeness [4, 17].

To address the above issues, in this study, we collected dialogue data that include situations in which people are unwilling to answer questions. To do so, we first created a list of difficult-to-answer questions on the basis of cross-cultural communication literature [11, 13] and then used the list to collect dialogues in which such questions appear. We also investigated whether it is possible to train a machine learning-based classifier to recognize unwillingness to answer questions by using multi-modal information contained in the collected data.

2 Dialogue Data Collection

We collected dialogue data that contain situations in which people are unwilling to answer questions. To collect such data, we first created a list of questions that people are likely to be unwilling to answer. We then collected dialogues in which the speakers used this list to pose questions. In the following subsections, we describe the processes of selecting the questions and collecting the dialogue data.

2.1 Selection of Topics

We created our list of questions that people feel unwilling to answer by referring to studies in the field of cross-cultural communication in which the development of interpersonal relationships in different cultures is discussed. For example, studies have been done on how long it takes to converse on certain topics [3, 15] and which topics are preferred when people meet for the first time [11, 13].

We referred to studies by Jeon [11] and Kumagai and Ishii [13] to select questions that people are unwilling to answer because their research targeted Japanese speakers and we want to conduct the data collection in Japanese. The study by Jeon investigated questions that university students in Japan and Korea are unwilling to answer; we used questions that over 30% of the students felt they would be unwilling to answer. The study by Kumagai and Ishii investigated topics that people of a wide range of ages in Japan and Korea are unwilling to talk about; we used topics that over 30% of the people felt they would be unwilling to talk about.

We used the questions from Jeon's study as they are. For the topics from Kumagai and Ishii's study, we created question texts based on the topics. For example, for the topic "Education", we created the question "Which university are you from?".

We merged the questions from Jeon's study with those we created from the topics in Kumagai and Ishii's study to create the final list of questions, which is provided below. Words in parentheses indicate questions that were used only for people who fit the relevant profile.

- What is your yearly income?
- Have you ever thought about getting a divorce? (married)
- What is your household income?
- What is your weight?
- What religion do you believe in?
- Which part of your body do you dislike?
- How long have you been dating your boyfriend/girlfriend? (unmarried)
- Which political party do you support?
- How much do you make at your part-time job? (student)
- When do you plan to get married?
- Do you have a boyfriend/girlfriend?
- How much is your allowance? (student)
- Which university are you from?
- Did you fail to enter university and have to wait for another chance?
- Do you plan to find a job after graduation? (student)
- What is your ideal type of woman/man?
- What do your family members do for a living?
- Which high school did you go to?
- Do you plan to keep your current job? (employee)
- How would you describe your character?
- What is your dream?
- What kind of work do you want to do in the future? (student)

2.2 Procedure of Dialogue Data Collection

To collect dialogues in which people are unwilling to answer questions, we recruited participants, put them in pairs, and gave one the role of asking questions (hereinafter, "topic provider") and the other the role of answering that question (hereinafter, "responder"). We then instructed each pair to ask and answer the questions in our difficult-to-answer question list.

We presented the list of questions (Sect. 2.1) to the topic providers in advance, and they were instructed to obtain the answer for each question within a certain time limit. The responders were instructed to answer the questions asked by the topic provider. If the responders were asked something they did not want to answer, we made it explicit that answering such questions was not mandatory. The questions asked by each topic provider were randomly selected from the list of questions.

At the end of each dialogue, we surveyed the responders about their degree of unwillingness to answer questions and surveyed the topic providers about their degree of difficulty to ask questions. We administered the questionnaire to both participants because we felt the scores might be related and the information of the topic providers might be useful in analyzing the responders' unwillingness to answer questions.

2.3 Implementation of Dialogue Data Collection

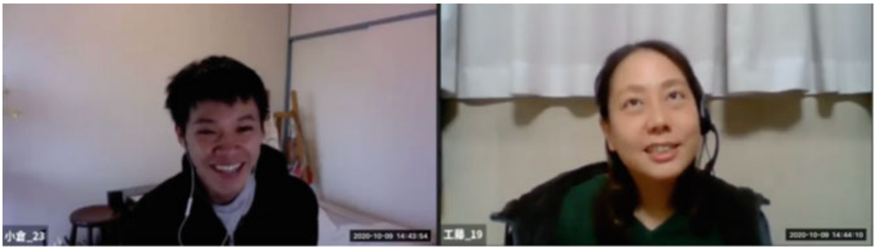
All dialogues were conducted using Zoom, a videoconferencing tool, and we recorded the video and audio.

We recruited 24 participants (12 male, 12 female) ranging in age from 20s to 60s and divided them into three groups of eight participants each. Four of the eights participants per group were assigned the role of topic provider and four were assigned the role of responder. Pairs were formed by selecting one person from each role. Each pair introduced themselves to each other at the beginning of the dialogue and then performed seven short dialogues for about two minutes each. Each participant performed a series of dialogues four times by changing the pairs. A total of 336 dialogues (only one question asked per dialogue) was collected, and the average duration was 2 min and 8 s. We transcribed each dialogue and the average number of utterances per dialogue was 73.2. Figure 1 shows a dialogue scene and an excerpt of the dialogue. The transcripts are also provided at the bottom.

Responders were asked to rate the degree of unwillingness to answer questions on a 7-point scale (1 = very willing to answer questions, 7 = very unwilling to answer questions) at the end of each dialogue. The topic providers were asked to rate the degree of difficulty to ask questions on the same scale (1 = not difficult to ask at all, 7 = very difficult to ask).

This data collection procedure was approved by the research ethics committee of Nagoya University. The participants were recruited through a recruiting agency, and only those who agreed to a consent form participated in this data collection. Participants agreed that the data could be used for academic research, and they were informed that they could quit the experiment at any time if they felt uncomfortable.

Figure 2 shows the distribution of the degree of unwillingness to answer questions for each responder. The horizontal axis shows the user ID (responder), and the vertical axis shows the frequency of the degrees answered by that user. Although some of the data indicated that the responders were willing to answer certain questions, we also successfully collected a number of dialogues in which people were unwilling to answer questions.



Topic provider:	You said that you wanted to be an actress. You're working a regular job right? Do you have any goals or dreams now?
Responder:	Um...
Topic provider:	It doesn't need to be so dramatic. Anything you want to do?
Responder:	Well, I don't have any kind of dream right now. Sorry to kill the conversation. There are things I enjoy doing, but I don't feel like I have anything worthwhile to do or anything to look forward to beyond entertainment, so I don't have any goals now. It's like I'm living my afterlife.

Fig. 1 Scene and excerpts of the dialogue (left: topic provider, right: responder)

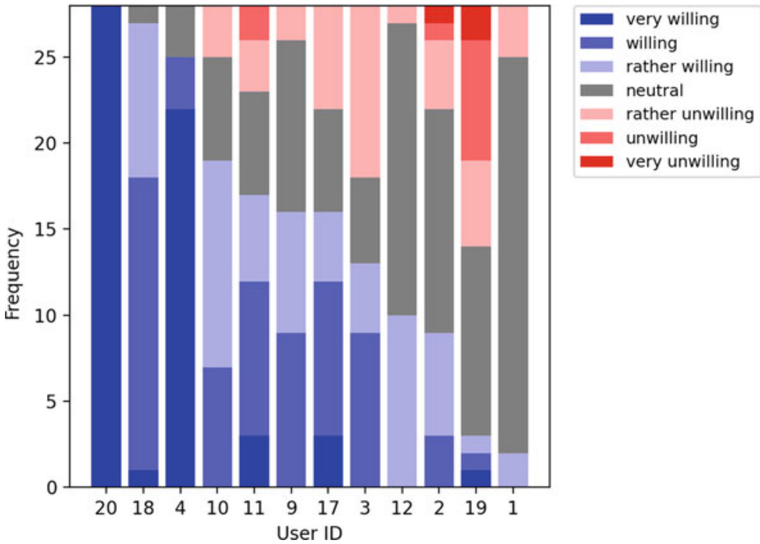


Fig. 2 Frequency of degree of unwillingness to answer question for each responder

Figure 3 shows a scatter plot of the average unwillingness rated by responders and the average difficulty of questions rated by topic providers for each question. The Pearson's correlation coefficient between the degrees of topic providers and responders was 0.67, which means that the more difficult the question was to ask, the more difficult it was to answer. Questions regarding yearly income and household income were those that people felt unwilling to answer and difficult to ask, and the question about politics was felt difficult to ask. Some questions, such as those

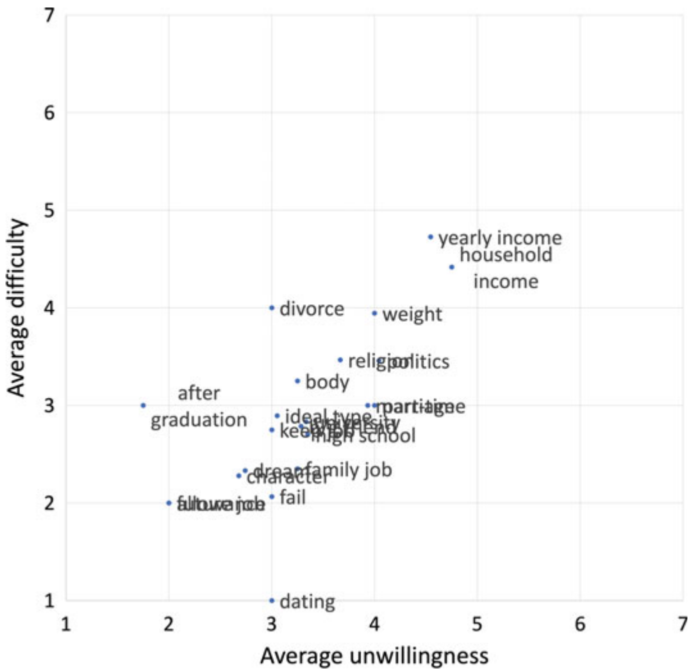


Fig. 3 Scatter plot of average unwillingness rated by responders and average difficulty rated by topic providers for questions

about divorce, dating, and work after graduation, had a discrepancy between the unwillingness to answer and the difficulty to ask.

3 Collecting Ratings from External Annotators

Since we collected ratings from the participants themselves, we felt these ratings might be quite subjective and therefore difficult for machine learning-based methods to guess. In addition, the ratings we obtained were for whole dialogues (containing a number of questions that may or may not have been related to the target question), not just the span of dialogue in which the question was posed. Therefore, the questionnaire scores might not accurately reflect the unwillingness to answer questions or the difficulty to ask questions.

To obtain more objective ratings for specific parts of dialogue, we first identified the span in each dialogue where participants were performing the questioning and answering. We then had multiple annotators rate the unwillingness to answer questions and the difficulty to ask questions. We used the average scores as the ground truth that was later utilized as a reference for training the machine learning-based classifier.

3.1 Identifying the Span of Questions and Answers

We used the ELAN toolkit [5] to identify where in the dialogue the question (from our difficult-to-answer question list) was being asked and where the response to that question was made. We extracted the span from the beginning of the question to the end of the answer for that question. The average duration of each span was 25.9 s with the standard deviation of 21.3 s. Some spans were longer than others because people tended to hesitate for some questions while others answered without any hesitation.

3.2 Annotation of Ratings

We recruited ten external annotators to collect ratings for the identified spans. These annotators were given a list of video files containing the QA spans as described in the previous section and asked to watch the videos and provide their ratings. We removed 28 videos that had technical problems in the recording and 17 videos in which questions could not be asked due to the time limit. In the end, each annotator was given 291 video files totaling 170.6 min.

Each annotator rated the degree of unwillingness to answer questions on a 7-point scale (1 = very willing to answer questions, 7 = very unwilling to answer questions) and the difficulty to ask questions on a 7-point scale (1 = not difficult to ask at all, 7 = very difficult to ask) for all videos. We asked the annotators to imagine they were the responders and the topic providers when providing their ratings. We also asked them not to be biased toward any particular ratings.

Figure 4 shows the histogram of collected ratings. We can see that 28.5% of the "unwilling to answer questions" data was rated over 4, and 36.8% of the "difficult to ask questions" was rated over 4. These results demonstrate that we successfully collected dialogue spans in which people were unwilling to answer questions and found it difficult to ask questions.

For each of the 291 dialogues, we calculated the average value of ratings. The Spearman's correlation coefficient between the average unwillingness to answer questions and the self-rated value by responders was 0.56, and that between the average difficulty to ask questions and the self-rated value by topic providers was 0.61. We therefore consider there to be a reasonable correlation between the self-declared values and the independently rated ones.

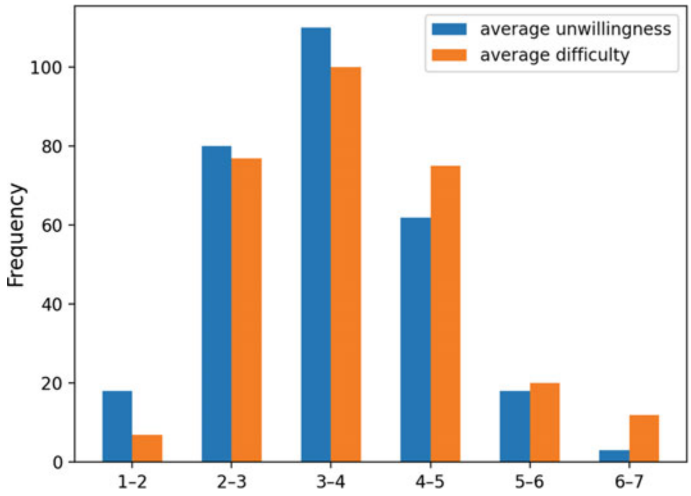


Fig. 4 Frequency of degree of average unwillingness and average difficulty

4 Estimating Unwillingness to Answer Questions and Difficulty to Ask Questions

To determine the current accuracy of models for estimating the unwillingness to answer questions and the difficulty to ask questions, we created machine learning-based models using the collected dialogue data. Since we did not have enough data to utilize deep neural networks, we turned to an existing method, namely, support vector regression [1].

4.1 Procedure for Creating a Model

For feature extraction, we used multi-modal features because the unwillingness to answer questions may appear as a facial expression, textual content, tone of voice, etc. Specifically, we extracted text, audio, visual, and facial features from the dialogue span identified in the previous section. We also used the transcriptions for that span to obtain utterance texts.

For extracting the text features, we used BERT [7], a masked language model, to convert each utterance in the span into 768-dimensional vectors. This was done by extracting the vector corresponding to the CLS token. We used an off-the-shelf BERT model[1] and did not perform any fine-tuning. The vectors for the utterances were averaged to create the text features. For the audio features, we used VGGish [10], a pre-trained model for audio classification, to convert each frame of audio in

[1] https://github.com/cl-tohoku/bert-japanese.

the span into 128-dimensional vectors. The vectors for the frames were then averaged to create the audio features. For the visual features, in the same manner as the audio features, we used ResNet50 [9], a pre-trained model for image classification, to convert each image of a video frame into 2048-dimensional vectors. The vectors for the frames were then averaged to create the visual features. Finally, the facial features were extracted by using OpenFace [2]. We extracted action unit-related features consisting of 35-dimensional vectors for each frame. The vectors for the frames were then averaged to create the facial features. In total, when all features were used, we had 2979 (768+128+2048+35)-dimensional vectors for each data span.

Since we use numeric values as scores, we regard the problem as a regression problem. Support vector regression (SVR) was utilized to train and predict the unwillingness to answer questions as well as the difficulty to ask questions. The data were randomly split into four folds to perform cross-validation. We used the scikit-learn package[2] for this process. The default parameter settings were used for the SVR implementation.

We performed two experiments: one to estimate the unwillingness to answer questions and the other to estimate the difficulty of asking questions. For the former, we used the audio, visual, and facial features extracted only from the responder's audio, video, and facial expressions, while the text features were extracted by using the utterances of both the responder and the topic provider so as to include the content of the question. The features for the latter were extracted in the same manner: the audio, visual, and facial features were extracted only from the topic provider's audio, video, and facial expressions, and the text features were extracted from the utterances of both the responder and the topic provider.

To examine the usefulness of the features, we tested the performance of all features as well as various combinations of the features. The evaluation metric we used was mean squared error. For comparison, we prepared two baselines: a random baseline (Random), which randomly returns an integer between 1 and 7, and one that always estimates the score by using the average value of training data (Average).

4.2 Results of Regression

The results of regression (mean squared error) for unwillingness to answer questions and the difficulty to answer questions are shown in Table 1. We can see that not all features contributed equally to the prediction performance.

Regarding the unwillingness to answer, the text features were the most salient. When used with the face features (face+text), the performance became the best, suggesting that these features are complementary to each other. The same performance was obtained by audio+face+text, but no improvement was made by the audio features. A statistical test (Wilcoxon signed-rank test) revealed that face+text was

[2] https://scikit-learn.org/stable/.

Table 1 Results of regression (mean squared error). Top-5 values in each column are shown in bold. The best values are underlined

	Unwillingness to answer	Difficulty to ask
Random	5.218	5.190
Average	0.989	1.154
text	**0.900**	1.177
audio	1.029	1.181
visual	0.964	1.127
face	1.030	1.247
audio+text	**0.901**	1.11
text+visual	0.924	**1.085**
face+text	<u>**0.892**</u>	1.154
audio+visual	0.946	1.107
audio+face	1.009	1.133
face+visual	0.962	1.127
audio+text+visual	0.915	<u>**1.069**</u>
audio+face+text	<u>**0.892**</u>	**1.100**
face+text+visual	0.922	**1.085**
audio+face+visual	0.945	1.108
All	**0.913**	<u>**1.069**</u>

significantly better than Average ($p < 0.01$). As for the difficulty to ask, we see a different trend: the best performance was obtained by audio+text+visual (no gain from the face features, as indicated by the result of All), outperforming Average significantly ($p < 0.05$). Considering that the performance of the audio, text, and visual features when used independently was similar to that of Average, we can conclude that these features are complementary to each other. Generally, it seems it was more difficult to detect the difficulty to ask questions. Investigating why this is so will be the focus of future work. Overall, we have demonstrated that we can detect the unwillingness to answer questions and the difficulty to ask questions to some extent.

5 Summary and Future Work

In this work, to achieve dialogue agents that are social and can exhibit hospitality, we collected dialogue data in which people are unwilling to answer questions. We then had annotators rate the unwillingness to answer questions and the difficulty to ask questions on dialogue spans where difficult-to-answer questions were being asked. Our findings demonstrate that we successfully collected dialogues and dialogue spans in which the users felt unwilling to answer and that it is possible to estimate such unwillingness by support vector regression to some extent.

There are a few issues that need to be addressed in future work. First, we need to improve the prediction performance, which is still a little low; other features and methods for regression should be investigated. Although we did not turn to deep learning-based methods due to a shortage of data, it may be possible to apply such methods by using pre-trained models trained with a large amount of multi-modal data [14, 16]. We also want to test which of the features were salient by performing various ablation studies. In addition, we want to utilize the trained regression model to develop a dialogue system that can detect when the user is unwilling to answer questions so as to improve the hospitality of the system. We focused on Japanese in this research, but we also want to use the data of other languages (e.g., English), as emotional expressions may differ depending on the culture. The data we collected are unique and can be used for other purposes, such as the generation of facial expressions under uncomfortable situations.

Acknowledgements We thank the anonymous reviewers for their helpful comments and suggestions. Funding was provided by a Grant-in-Aid for Scientific Research (Grant no. JP19H05692).

References

1. Awad M, Khanna R (2015) Support vector regression. In: Efficient learning machines. Springer, Berlin, pp 67–80
2. Baltrusaitis T, Zadeh A, Lim YC, Morency LP (2018) OpenFace 2.0: facial behavior analysis toolkit. In: Proceedings of the 13th IEEE international conference on automatic face & gesture recognition, pp 59–66
3. Berger CR, Gardner RR, Clatterbuck GW, Schulman LS (2006) Perceptions of information sequencing in relationship development. Hum Commun Res 3(1):29–46
4. Brown P, Levinson SC (1987) Politeness: some universals in language usage
5. Brugman H, Russel A, Nijmegen X (2004) Annotating multi-media/multi-modal resources with ELAN. In: Proceedings of the fourth international conference on language resources and evaluation, pp 2065–2068
6. Busso C, Bulut M, Lee CC, Kazemzadeh A, Mower E, Kim S, Chang J, Lee S, Narayanan SS (2008) Iemocap: interactive emotional dyadic motion capture database. J Lang Resour Eval 42(4):335–359
7. Devlin J, Chang MW, Lee K, Toutanova K (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, vol 1 (Long and short papers), pp 4171–4186
8. Grice HP (1975) Logic and conversation. In: Syntax and semantics: vol 3: Speech Acts. Academic, New York, pp 41–58
9. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
10. Hershey S, Chaudhuri S, Ellis DP, Gemmeke JF, Jansen A, Moore RC, Plakal M, Platt D, Saurous RA, Seybold B, et al (2017) CNN architectures for large-scale audio classification. In: Proceedings of the 2017 IEEE international conference on acoustics, speech and signal processing, pp 131–135
11. Jeon JM (2009) Shotaimen bamen ni okeru wadai kaihi ni kansuru situmonshi chousa (Survey on topic avoidance in first meetings). Language and Culture, pp 95–111. (In Japanese)

12. Komatani K, Okada S, Katade S (2020) Release of multimodal dialogue corpus Hazumi and new data collection containing biosensor signals. SIG-SLUD **B5**(02), 170–177 (2020). (In Japanese)
13. Kumagai T, Ishii E (2005) Preferences in conversational topic: a survey of Japanese and Koreans focusing on the younger generation. Jpn J Lang Soc 8(1):93–105 (In Japanese)
14. Lu J, Batra D, Parikh D, Lee S (2019) ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. arXiv:1908.02265
15. Nishida T (1998) Ibunka no ningen kankei. Taga Press. (In Japanese)
16. Sun C, Myers A, Vondrick C, Murphy K, Schmid C (2019) VideoBERT: a joint model for video and language representation learning. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7464–7473
17. Usami M (2006) Discourse politeness theory and cross-cultural pragmatics. In: Yoshitomi A, Umino T, Negishi M (eds) Readings in second language pedagogy and second language acquisition: In Japanese context. John Benjamins, pp 19–42
18. Yamazaki Y, Chiba Y, Nose T, Ito A (2020) Construction and analysis of a multimodal chat-talk corpus for dialog systems considering interpersonal closeness. In: Proceedings of the 12th language resources and evaluation conference, pp 443–448

Enhancing Self-disclosure In Open-Domain Dialogue By Candidate Re-ranking

Mayank Soni, Benjamin R. Cowan, and Vincent Wade

Abstract Neural language modelling has progressed the state-of-the-art in different downstream Natural Language Processing (NLP) tasks. One such area is of open-domain dialog modelling, neural dialog models based on GPT-2 such as DialoGPT have shown promising performance in single-turn conversation. However, such (neural) dialog models have been criticised for generating responses which although may have relevance to the previous human response, tend to quickly dissipate human interest and descend into trivial conversation. One reason for such performance is the lack of explicit conversation strategy being employed in human-machine conversation. Humans employ a range of conversation strategies while engaging in a conversation, one such key social strategies is *Self-disclosure* (SD). A phenomenon of revealing information about one-self to others. In this work, Self-disclosure enhancement architecture (SDEA) is introduced utilizing Self-disclosure Topic Model (SDTM) during inference stage of a neural dialog model to re-rank response candidates to enhance self-disclosure in single-turn responses from the model.

1 Introduction

Neural Language models based on neural language pre-training such as GPT [13], GPT-2 [14] have advanced the state-of-the-art in various NLP Tasks. Open-domain dialogue models such as DialoGPT [9], Blender-bot [10], and Meena [20] have shown promising performance in single-turn conversation. However, when focused

M. Soni (✉)
ADAPT Centre, Trinity College Dublin, Dublin, Ireland
e-mail: sonim@tcd.ie

B. R. Cowan
ADAPT Centre, University College Dublin, Dublin, Ireland
e-mail: benjamin.cowan@ucd.ie

V. Wade
ADAPT Centre, Trinity College Dublin, Dublin, Ireland
e-mail: vincent.wade@adaptcentre.ie

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_17

on social talk, such (neural) deep learning solutions have been strongly criticised for generating conversational utterances which although may have relevance to the previous human utterance, tend to quickly dissipate human interest, and descend into trivial conversation and disengage with the human user [21, 23, 30]. Some of the problems in their usage result in generating bland and repetitive responses like "Thank you", "ok", "I don't know" [12, 21, 22].

One of the reasons for such responses is the lack of explicit conversation strategy employed by a neural dialog system. Hence, in this paper we seek to enhance existing neural conversational agents based on novel model adaptation which enables the neural dialog model to activate *self-disclosure*. Self-disclosure has been well researched as higher order conversational skill in psychology literature where human centric evaluation of such strategies has proven to enhance the relationship between interlocutors and positively affect their engagement in the conversation [4, 5, 19]. However, to the authors knowledge, attempting to explicitly empower a neural dialog model to utilize self-disclosure with a corpus-neutral approach has not been attempted before. In this paper we focus on integrating Self-disclosure and evaluate the degree to which it is seen enhanced in the responses by implementing a novel architecture. Below, we briefly describe self-disclosure level, Self-disclosure Topic Model (SDTM), Self-disclosure enhancement architecture (SDEA), experiment and results.

2 Self-disclosure

Humans employ a range of conversation strategies to fulfil multiple goals in a conversation [18]. Conversation strategies are discourse units which could span multiple utterance turns and are typically larger than speech acts [5]. Such strategies contribute to building and maintaining human relationships during the dialogue. Humans employ a range of different strategies to build rapport and increase interpersonal cohesiveness [5]. It is argued that over time humans behave in ways to fulfill mutual behavior expectations [17]. The most important conversation strategy is Self-disclosure. Researchers [2] have expounded on the idea of social penetration, Social Penetration Theory (SPT) proposes that communication between two people moves from shallow to deeper levels as the relationship progresses. SPT primarily proposes the idea that relationships progress through self-disclosure, a phenomenon of revealing information about one-self to others. This information about oneself could consist of one's thoughts, aspirations, past events, and future plans. Self-disclosure is a key social strategy identified by information being revealed about oneself. This helps in creating rapport and a feeling of mutual trust among the participants engaging in dialogue. People disclose information about themselves to improve and maintain relationships and form deeper connections. Employing appropriate self-disclosure can lead to a feeling of mutual trust, friendliness and overall satisfaction in a conversation [1]. Researchers showed that self-disclosure is a key strategy in building rapport in a peer-to-peer tutoring experiment [5]. It has also been shown that self-disclosure in a human-machine conversation can lead to higher satisfaction in conversation [4, 19]. Motivated by research in psycho-linguistics, socio-linguistics and SPT, self-disclosure enhancement architecture (SDEA) in neural dialog systems is implemented and evaluated.

2.1 Self-disclosure (SD) Level

Self-disclosure can be divided into multiple levels. We follow the three level self-disclosure recognition as highlighted in [3]. The study highlights three levels of self-disclosure from social science and psychology literature: G(general) for no-disclosure, M (medium disclosure) and H (high disclosure) [6, 7]. These three levels are organized in progressing order of sensitive information being revealed by an agent.

(G) levels of self-disclosure includes no self-disclosure. Responses that are about a third-person, event or thing are labelled as general disclosure. (M) level of self-disclosure comprise of information about oneself. Examples are statement that increase information about a user such as birthday and events. Personal pronouns such as 'My', 'I' are identifiers of medium level disclosure in an utterance. Medium disclosure contains information that is non-sensitive in nature. H levels of self-disclosure contains personal and sensitive information. Mentioning concerns and insecurities are a cue to identify high levels of self-disclosure. Responses such as *'I am overweight and trying to lose some weight'* is an example of high self-disclosure. We refer to [3] for list of the keywords that help identifying H levels of self-disclosure.

2.2 Self-disclosure Recognition Model

The first step towards re-ranking response candidates, described in Sect. 3, is to be able to recognize self-disclosure levels computationally. There are various researches which have implemented a supervised classifier [4, 8], however these require disclosure annotated dataset to train a classifier. Hence, we utilize SDTM [3] as it is a semi-supervised model to recognize levels of self-disclosure as discussed in Sect. 2.1. This was developed for recognizing levels of self-disclosure in longitudinal Twitter conversations. The model recognizes three levels of self-disclosure as mentioned in Sect. 2.1 and relies on seed-keywords and n-grams to recognize the level of self-disclosure. The model classifies a given sentence into G versus M/H of disclosure. G means no disclosure and M/H are medium and high level disclosures. M level (non-sensitive) disclosures are defined by referring to personal pronouns and sharing information about oneself. While high level self-disclosure is classified by revealing secret or vulnerable information about oneself. The classification of degree of self-disclosure into high, medium, and general is based on [6, 7]. $FP + SE1$ is utilized to recognize self-disclosure levels from SDTM [3].

3 Self-disclosure Enhancement By Candidate Re-ranking

Neural Language models based on neural language pre-training such as GPT [13], GPT-2 [14] has advanced the state-of-the-art in various NLP Tasks. Pre-training has advanced the development of neural models for open-domain conversation generation. DialoGPT [9] was released as a pre-trained dialog response generation model based on GPT-2. At the core of DialoGPT [9] is language modelling, a task of estimating unsupervised language distribution from a set of examples $(t_1, t_2, t_3, \ldots, t_n)$. Since open-domain dialog follows a natural sequence, turns in dialog can be modelled as product of conditional probabilities. If source sentence S consists of a series of tokens $(t_1, t_2, t_3, \ldots, t_m)$. Then, the response sentence R can be framed as continuation of source tokens $(t_(m + 1), \ldots, t_n)$

$$p(R|S) = \prod_{m+1}^{n} P(t_n|t_1, \ldots, t_{n-1}) \tag{1}$$

DialoGPT is employed as the base neural dialog model for experimentation. Typically neural dialog models (such as DialoGPT) consist of three components namely Training Corpus, Neural Architecture, and Inference Strategy. This paper focuses on adaptation in the Inference Stage of a neural dialog system. Neural dialog models generate responses by following a probability or sampling based inference strategy. Inference strategies in open-domain dialog systems are based on probabilistic sampling (e.g. nucleus sampling [16], top-k [11], beam search, greedy sampling) from a fixed vocabulary. Rather than optimizing for semantic coherence, this work investigates if adaptation of the inference mechanism should be explored to consider semantically coherent lower probability responses that are more indicative of a self-disclosure level (G, M, H). To evaluate this hypothesis, Self-Disclosure Topic Model (SDTM) [3] is used to search response candidates for a pre-defined self-disclosure level and the said candidate is rendered as the response. Figure 1 illustrates the architecture of SDEA. The gray unit in Fig. 1 provides an overview of SDEA. Response (yellow) and SDEA Response(green) units show handpicked example of response generation from DialoGPT [15] and DialoGPT enhanced with SDEA.

Algorithm 1 Self-Disclosure Enhancement Architecture (SDEA)

1: **for** t in range $(sequencelength)$: sample tokens $(t_1, t_2, \ldots, t_n$ **do**)
2: **end for**
3: Join tokens to form one sequence S
4: Split S on $eos - id$ to obtain candidates $C_1, C_2, \ldots, C_m$
5: Compute SD level of candidates $C_1, C_2, \ldots, C_m$
6: Render 1st candidate in sequence S, C_x with specified SD level

The decoding algorithm generates tokens defined by the sequence length. For instance if the sequence length is 20, 20 tokens will be generated in accordance with

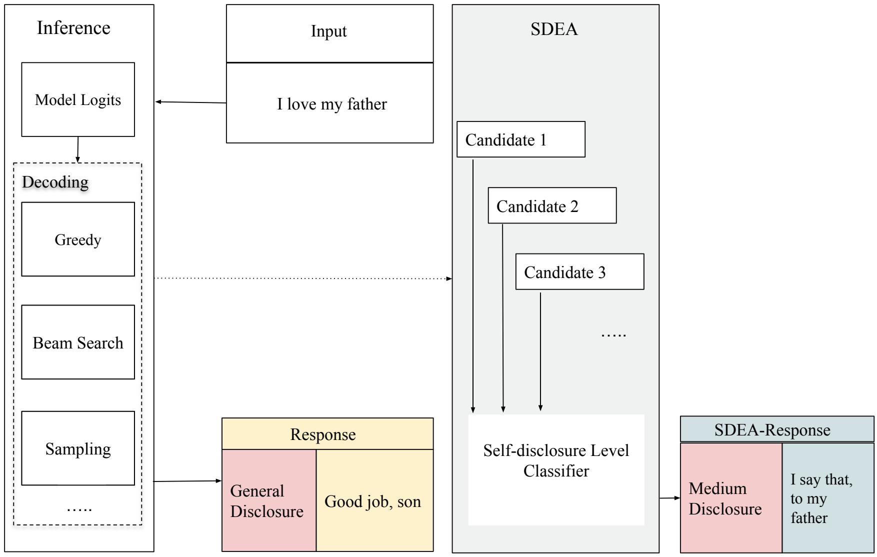


Fig. 1 Overview of SDEA (grey unit). Response in green is Self-disclosure enhanced response

the decoding strategy chosen. The tokens also contain end-of-text ($eos - id$) token which signifies the end of sentence. This leads to generation of multiple sentences separated with an end-of-text token within a sequence. These are called response candidates. Theoretically, we argue that given a large distribution of tokens and a large dataset. It should be possible to generate responses from a neural dialog system consisting of a certain SD level. Algorithm 3 elaborates steps involved in generating SD enhanced response.

4 Experiment

The central objective of the experiment was to evaluate, if the self-disclosure levels (as defined in Sect. 2.1) are higher than that of a vanilla system. Vanilla responses are defined as responses generated by a neural dialog system using a decoding strategy. To evaluate the effect of the proposed self-disclosure enhancement in neural dialog model by re-ranking candidates architecture, responses from vanilla and self-disclosure enhancement in neural dialog model by re-ranking candidates architecture are generated.

Medium sized DialoGPT model is employed as the Neural dialog system for the experiment. Changes are made to the decoding script from [15]. Vanilla responses are generated with Nucleus sampling ($top - p$) value of 0.9 and a sequence length of 100. Self-disclosure enhanced responses are generated by incorporating SDEA with aforementioned vanilla response setup. A larger sequence length is used so that

Table 1 Handpicked first prompts examples from Dailydialog and Switchboard

Dailydialog	Switchboard
Are things still going badly with your house guest	Um how's it been this week for you
What kind of food do you like	All right Amy how are you doing today
Hello is John in	So who's your uh favorite team

enough candidates could be produced to make a selection from. The experiment is performed on two dialog datasets: Dailydialog [24] and Switchboard [25]. The description of dataset preparation is in Sect. 4.1. Following the dataset preparation, responses from both (SDEA and Vanilla system), with their SD levels is obtained. Finally, Chi-Square test is conducted on the the distribution of SD levels from both the systems. Results are further discussed in Sect. 4.2. The decoding script from [15] is utilized. The history is erased after each response generation so that there is no effect of context on the new turn.

4.1 Data Configuration

Data is prepared using two dialog datasets viz. Dailydialog [24] and Switchboard [25]. Since, in the early stages of a conversation, self-disclosure leads to longer conversations [4], and the task is to test the enhancement of self-disclosure in response from a neural dialog model in a single turn. Only the first dialog turn from each conversation in the dataset is selected. This leads to creation of a dataset consisting of only first (or second if the first dialog sentence is noisy) dialog turn. Handpicked examples of such dialog turns from both, Dailydialog [24] and Switchboard [25] can be seen in Table 1.

4.2 Result and Error Analysis

The results show a clear difference in distribution of responses between general and medium disclosure in both datasets ($p \ll 0.005$), indicating that SD levels are higher from SDEA. In the DailyDialog dataset, 39.20% of responses from vanilla system have *medium* level self-disclosure, whereas 97.60% responses from the self-disclosure enhancement system have *medium* level self-disclosure. Similarly, out of 1277 prompts in the switchboard dataset, 33.90% had medium disclosure from the vanilla system, whereas 95.20% responses have *medium* disclosure levels from SDEA. Also, SDEA was unable to find *medium* disclosure responses for 4.80% switchboard prompts and 2.40% prompts because no candidate with a medium dis-

Table 2 Automatic dialog evaluation metrics scores on filtered DailyDialog dataset

Method	SD Level		NIST		BLEU		ENTROPY	DIST		Avg. length
	General (%)	Medium (%)	N-2	N-4	B-2	B-4	E-4	D1	D2	
DIALOGPT (Nucleus Sampling, p = 0.9)	60.80	39.20	**0.37**	**0.37**	**0.016**	**0.003**	8.83	**0.23**	**0.75**	09.85
DIALOGPT (Nucleus Sampling, p = 0.9) **+SDEA**	02.40	**97.60**	0.33	0.33	0.009	0.001	**8.98**	0.17	0.67	**11.03**

Table 3 Automatic dialog evaluation metrics scores on filtered Switchboard dataset

Method	SD Level		NIST		BLEU		ENTROPY	DIST		Avg. length
	General (%)	Medium (%)	N-2	N-4	B-2	B-4	E-4	D1	D2	
DIALOGPT (Nucleus Sampling, p = 0.9)	66.10	33.90	0.27	0.27	0.01	0.0005	9.10	**0.22**	**0.73**	10.17
DIALOGPT (Nucleus Sampling, p = 0.9) **+SDEA**	4.80	**95.20**	**0.34**	**0.34**	0.01	0.0005	**9.22**	0.17	0.67	**10.94**

closure was found within a sequence length of 100. Pearson's Chi-Square test is then performed to confirm that the SD level distributions from the SDEA and Vanilla systems are statistically significant. The p values for both datasets reveal that the SD levels are distributed differently, and $DialoGPT + SDEA$ responses clearly lean towards generating responses with medium disclosure. Results can be seen in Tables 2 and 3.

We further evaluate responses from vanilla system and vanilla system +SDEA on various automated dialog metrics such as BLEU [26], NIST [27], METEOR [28], Entropy [29], Dist-n [30]. The primary reason for this evaluation is to be caution against irrelevancy when enhancing self-disclosure. It is observed that the difference between the aforementioned systems is minimal. For DailyDialog dataset, Vanilla system has better NIST, BLEU and DIST scores and Vanilla System +SDEA has better Entropy and Avg.Length. Similarly, for Switchboard dataset, Vanilla system has better DIST scores and Vanilla System +SDEA has better NIST, Entropy and Avg. Length. Thus, it can be inferred that enhancing self-disclosure does not lead to irrelevant response generation. Future work will include evaluation in Multi-reference setting [31] using reddit multi-reference dataset [9].

5 Discussion, Limitation and Future Work

This study highlights that the current decoding strategies do not, yet, take into account relationship building with the user by employing any method such as using a conversation strategy. One of the limitation of the proposed method is that the processing is post-generation. Hence, we are limited by the candidates generated by a Language Model Decoding Strategy. If the candidates within a given sequence length do not have a specified level of disclosure then there would be no disclosure enhanced response generated. Hence, future work will include changes to training stage to generate enhanced disclosure responses directly. Human evaluation of the improved self-disclosure and disclosure reciprocity will be part of future research.

6 Conclusion

SDEA was presented as an architecture to enhance self-disclosure in neural dialog systems by response candidate re-ranking. The approach under consideration was corpus-neutral, meaning that no changes to the corpora were made, and the only change was made during the inference stage. This is helpful since the technique may be employed with a variety of decoding-based neural dialog models. A step is taken in the direction of making dialog systems conversation strategy aware.

Acknowledgements This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (D-REAL) under Grant No. 18/CRT/6224. We would like to thank anonymous reviewers from IWSDS 2021 for their valuable comments.

References

1. Joinson AN, Paine CB (2007) Self-disclosure, privacy and the internet. The Oxford handbook of internet psychology, 2374252
2. Altman I, Taylor DA (1973): Social penetration: the development of interpersonal relationships. Holt, New York
3. Bak A (2014) Self-disclosure topic model for classifying and analyzing Twitter conversations. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, pp 1986–1996
4. Ravichander A, Black AW (2018) An empirical study of self-disclosure in spoken dialogue systems. In: Proceedings of the 19th annual SIGdial meeting on discourse and dialogue. Association for Computational Linguistics, pp 253–263
5. Ravichander A (2018) An empirical study of self-disclosure in spoken dialogue systems. In: Proceedings of the 19th annual SIGdial meeting on discourse and dialogue. Association for Computational Linguistics, pp 253–263
6. Vondracek S, Vondracek F (1971) The manipulation and measurement of self-disclosure in preadolescents. Merrill-Palmer Q Behav Devel 17(1):51–58

7. Barak A, Gluck-Ofri O (2007) Degree and reciprocity of self-disclosure in online forums. Cyberpsychol Behav 10(3):407–417
8. Yang D, Yao Z, Kraut R (2017) Self-disclosure and channel difference in online health support groups. In: Proceedings of the international AAAI conference on web and social media, vol 11, No 1
9. Zhang Y, Sun S, Galley M, Chen YC, Brockett C, Gao X, ... Dolan B (2019). Dialogpt: large-scale generative pre-training for conversational response generation. arXiv:1911.00536
10. Roller S, Dinan E, Goyal N, Ju D, Williamson M, Liu Y, Xu J, Ott M, Shuster K, Smith EM, Boureau Y-L, Weston J (2020) Recipes for building an open-domain chatbot
11. Fan Y (2018) Hierarchical neural story generation. In: Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: long papers). Association for Computational Linguistics, pp 889–898
12. Vinyals O, Le Q (2015) A neural conversational model
13. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI blog 1(8):9
14. Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training
15. andreamad8. (n.d.) andreamad8/DialoGPT2-Interact. GitHub. https://github.com/andreamad8/DialoGPT2-Interact
16. Holtzman A, Buys J, Du L, Forbes M, Choi Y (2020) The curious case of neural text degeneration
17. Spencer-Oatey H (2008) Culturally speaking second edition: culture, communication and politeness theory. Bloomsbury Publishing
18. Tracy K, Coupland N (1990) Multiple goals in discourse: an overview of issues. J Lang Soc Psychol 9(1–2):1–13
19. Jain A, Pecune F, Matsuyama Y, Cassell J (2018) A user simulator architecture for socially-aware conversational agents. In: Proceedings of the 18th international conference on intelligent virtual agents, pp 133–140
20. Adiwardana D, Luong MT, So D, Hall J, Fiedel N, Thoppilan R, Yang Z, Kulshreshtha A, Nemade G, Lu Y, others (2020) Towards a human-like open-domain chatbot. arXiv:2001.09977
21. Huang M, Zhu X, Gao J (2020) Challenges in building intelligent open-domain dialog systems. ACM Trans Inf Syst (TOIS) 38(3):1–32
22. Gao J, Galley M, Li L (2018) Neural approaches to conversational ai. In: The 41st international ACM SIGIR conference on research and development in information retrieval, pp 1371–1374
23. Zhang S, Dinan E, Urbanek J, Szlam A, Kiela D, Weston J (2018) Personalizing dialogue agents: i have a dog, do you have pets too? arXiv:1801.07243
24. Yanran L, Su H, Xiaoyu S, Wenjie L, Ziqiang C, Niu S (2017) DailyDialog: a manually labelled multi-turn dialogue dataset
25. Calhoun S, Carletta J, Brenier JM, Mayo N, Jurafsky D, Steedman M, Beaver D (2010) The NXT-format Switchboard Corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. Lang Resour Eval 44(4):387–419
26. Papineni K, Roukos S, Ward T, Zhu WJ (2002) Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the association for computational linguistics, pp 311–318
27. Doddington G (2002) Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the second international conference on human language technology research, pp 138–145
28. Lavie A, Agarwal A (2007) METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In: Proceedings of the second workshop on statistical machine translation, pp. 228–231
29. Zhang Y, Galley M, Gao J, Gan Z, Li X, Brockett C, Dolan B (2018) Generating informative and diverse conversational responses via adversarial information maximization. arXiv:1809.05972

30. Li J, Galley M, Brockett C, Gao J, Dolan B (2015) A diversity-promoting objective function for neural conversation models. arXiv:1510.03055
31. Gupta P, Mehri S, Zhao T, Pavel A, Eskenazi M, Bigham JP (2019) Investigating evaluation of open-domain dialogue systems with human generated multiple references. arXiv:1907.10568

On the Impact of Self-efficacy on Assessment of User Experience in Customer Service Chatbot Conversations

Yuexin Cao, Vicente Ivan Sanchez Carmona, Xiaoyi Liu, Changjian Hu, Neslihan Iskender, André Beyer, Sebastian Möller, and Tim Polzehl

Abstract In this chapter, we analyse influencing factors for the assessment of user experience (UX) from a chatbot operating in the domain of technical customer support. To find out which UX factors can be assessed reliably in a crowdsourcing setup, we conduct a crowd-based UX assessment study through a set of scenario-based tasks and analyse the UX assessments in the light of influencing user characteristics, i.e., self-reported self-efficacy of individual users. By segmenting users according to self-efficacy, we find significant differences in UX assessment and expectations of users with respect to a series of UX constituents like *acceptability, task efficiency, system error, ease of use, naturalness, personality* and *promoter score*. Our results strongly suggest a potential application for essential personalization and user adap-

Y. Cao · N. Iskender · S. Möller · T. Polzehl (✉)
Technical University of Berlin, Ernst-Reuter-Platz 7, 10587 Berlin, Germany
e-mail: tim.polzehl@dfki.de

Y. Cao
e-mail: y.cao@campus.tu-berlin.de

N. Iskender
e-mail: neslihan.iskender@tu-berlin.de

S. Möller
e-mail: sebastian.moeller@dfki.de

V. I. S. Carmona · X. Liu · C. Hu
Lenovo Research AI Lab, No. 6 Shangdi West Road, Haidian District, Beijing, China
e-mail: vcarmona@lenovo.com

X. Liu
e-mail: liuxy63@lenovo.com

C. Hu
e-mail: hucj1@lenovo.com

A. Beyer
Crowdee GmbH, Zehdenicker Str. 5, 10119 Berlin, Germany
e-mail: andre@crowdee.com

S. Möller · T. Polzehl
German Research Centre for Artificial Intelligence, Alt-Moabit 91c, 10559 Berlin, Germany

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_18

tation strategies utilizing self-efficacy for the personalization of technical customer support chatbots. Therefore, we recommend considering its influence when designing chatbot adaptation strategies for maximized customer experience.

1 Introduction

Recently, conversational agents like chatbots, are widely used in the customer service domain. A good chatbot improves customer satisfaction by providing a fast, easy and satisfactory way to solve customer problems. To evaluate the performance of a chatbot, subjective evaluation from a user's point-of-view, namely the assessment of user experience (UX), is often applied. UX can be defined as "A person's perceptions and responses that result from the use and/or anticipated use of a product, system or service", according to [10]. In UX evaluation for the customer service domain, customer segmentation is widely used by dividing customers into groups that can be targeted based on information such as geographic (live place), socio-demographic (age, gender), psychographic (lifestyle, personality), and behavioural (consumption, spending) [5, 16]. By sending personalized messages or tailoring chatbot services to the needs of user segments, the satisfaction towards the customer service could be improved, contributing to customer loyalty and retention. As one of the essential factors for customer segmentation, [13] has identified self-efficacy, which describes the desire to be seen as unique and the determination to claim that. Customers with high self-efficacy urge for attention and seek out personalized, enriching, and emotionally satisfying experiences. If such expectations are violated, customers may leave more easily than customers with low self-efficacy [13].

In this paper, we investigate the robustness of UX assessment as well as UX expectations in conjunction with user's self-efficacy in a study operationalizing three task-based scenarios on Motorola Support Virtual Agent chatbot "Moli".[1] Recently, human evaluation of dialogue assessment shifted from a lab to crowd environment for reasons of scalability, speed, and costs [1, 7, 22]. Still, to the best of the authors knowledge, no study has analysed UX in the light of self-efficacy in a crowdsourcing setup to date. To fill this research gap, we analyse expectations of self-efficacy segmented users with respect to a series of UX constituents like acceptability, task efficiency, system error, ease of use, naturalness, personality, and promoter score.

2 Related Work

Regarding evaluation methods for dialogue systems, Deriu et al. [3] observed that human evaluation has shifted from lab to crowd. Especially for usability evaluation, crowdsourcing proves beneficial due to lower cost and time efforts [2, 6, 12, 15, 18].

[1] https://moli.lenovo.com/callcenter/moli.

Liu et al. [15] revealed applicability and value of crowd-based evaluation when comparing crowd and lab in a comparative study. Banches et al. [1] compared expert- and crowd-based annotations for evaluating chatting sessions at the turn level and found that simple majority vote over crowd-sourced annotations exhibits similar or even higher inter-annotator agreements compared to expert annotations. Additionally, Yu et al. [22] used crowdsourced annotations to annotate chatbot responses for likability and engagement between the crowd-workers and the chatbot. Other studies proved comparability in between crowd and lab, or crowd and expert annotation quality in related applications [8, 9].

As of UX evaluation, it is non-trivial to assess user experience since it is subjective, context-dependent, and dynamic over time [14]. Longitudinal UX evaluation methods are widely used, accessing the people's feelings about a system after using it for a while. There are a few existing UX evaluation methods suitable for the subjective evaluation, such as Usability Metric for User Experience (UMUX) [4], Chatbot Evaluation Questionnaire [19], Net Promoter Score (NPS) [20] and ITU-T Recommendation P.851 [11]. The International Telecommunication Union (ITU-T) is a specialized organ of the United Nations in the field of telecommunications, focusing on Standardization Sector. ITU-T Recommendation P.851 is a Recommendation for subjective quality evaluation of spoken dialogue systems [11] in order to gain personal, task-related and system-related information, such as availability of information obtained from the system, perceived system understanding, frequency of system errors, congruence with the user's expectations, etc. In [17] Möller et al. extracts eight components with the help of a principal component analysis, which form the basis of this work item conception.

With respect to self-efficacy, Lai et al. [13] describes self-efficacy on basis of psychologist's definitions as desire to be seen as unique and the determination to claim that. In the customer domain, the concept can be interpreted broader, including a global perception of oneself and one's self-esteem reactions to that self-perception [21]. Accordingly, users with high self-efficacy perceive themselves as unique and believe that they will get value and respect from others. In customer service, if a customer of high self-efficacy feels that the chatbot interaction is not personally enriching or satisfactory enough, this customer is more likely to stop the interaction, or switch to a competitor company.

3 Experimental Setup

Task and item design. We carefully designed three dialogue scenarios based on an expert review of a large number of real cases, all of which require a chatbot interaction to solve the scenario problem. For selection we considered multiple criteria, e.g., degree of expected dialog complexity, ambiguity, etc., trying to retain variation. Crowd-workers were provided with additional situational information, e.g., history of troubleshooting steps done, stopping criteria, indication what kind of answer is expected, and when to consider a problem as solved, etc. For example, the phone was said to have a charging problem and cannot be powered on with normal working charging system, port, charger and wall outlet. Users were to interact with Moli until Moli explains how to execute a battery diagnosis function, which finally indicates the reason of the problem. In another scenario, the phone was said to be dropped into water, and users were to inquire into warranty issues. In the third scenario, users were to inquire about a matching hardware (here for wireless charging), which was actually not available for the given phone model.

Adapting ITU-T Recommendation P.851, we selected five factors to include: (1) acceptability, (2) task efficiency, (3) system errors, (4) ease of use, and (5) naturalness. We further introduced two additional factors: personality and promoter score, the latter of which being inspired by the Net Promoter Score [20], resembling the likelihood of further recommendation to friends and others and the willingness to reuse the chatbot. Eventually we created 14 items pairs, with a pair consisting of one positively and one negatively formulated item. Exact item formulation resembles [17]. For self-efficiency, we investigated four items SE1-SE4, according to the discrete description in [13]. All items are shown in Table 1. The item order was randomized.

Table 1 Items and definitions for UX evaluation

Items	Definition
A1–A5	Five pairs assessing the factor acceptability, measuring the helpfulness (A1), satisfaction pleasure (A2), efficiency (A3), dialogue smoothness (A4), and length (A5)
TE1-TE3	Three pairs assessing the factor task efficiency, measuring the clearness and scope (TE1), accuracy of the solutions (TE2), and ease of disambiguation (TE3)
SE	One pair assessing the factor system error, measuring the perception of mistakes in understanding
E1-E2	Two pairs assessing the factor ease of use, measuring the ease of use (E1) and the expected behavior of the chatbot (E2)
N	One pair assessing the factor naturalness, measuring the naturalness of the chatbot reaction
P	One pair assessing the factor personality, measuring the politeness and friendliness of the chatbot
PS	One pair assessing the factor promoter score, measuring the likelihood of reuse and recommendation of the chatbot
SE1	I would feel inclined to switch to a competitor provider if the service experience of my current provider is not personally enriching or satisfactory enough for me
SE2	I am inclined to upgrade current contract if there was a more personally enriching or satisfactory experience for me connected to it
SE3	To my mind, the interaction with Moli cannot be taken seriously because it doesn't seem to be worthwhile to communicate with it
SE4	I stopped the interaction with Moli because I felt it would not be able to understand or help anyway

Crowdsourcing setup. We conducted all of the crowdsourcing experiments on the "Crowdee" platform.[2] The crowd workers were instructed to read the problem description first, then interact with Moli chatbot, and finally answer the UX and segmentation items. Each item was displayed on a single page using a 5-point Likert scale, ranging from *strongly agree, agree, neither agree nor disagree, disagree* to *strongly disagree*. Items on self-efficacy had an additional answer option *Cannot tell* designed for people who have difficulties in the self-assessment task.

Quality control. To control the quality of underlying crowdsourcing procedures live and directly while executing the study and excluding unmotivated users before they can introduce noise in the annotations, the Crowdee platform offers real-time scoring of participants. We chose the continuous consistency monitoring method to apply to the pairs of our inverse—not-inverse items, setting a rather conservative thresholds for the automatic participant exclusion, according to the results of an internal pre-test. Finally, answers with too short working time were also registered to be rejected in real-time. Compliant participants were provided all scenario iteratively by the automated quality control workflow.

4 Result

Overall, 313 crowd workers were recruited for the study to collect 100 repetitions of each of our three scenarios. Hence, the majority of participants passed the automatic quality control checkups described above. As a first indication, the low exclusion rate paired with very positive qualitative feedback given to us in the end of the study suggest that the design has been understood and the study could be robustly conducted in crowd environments. Eventually, very few participants chose the "cannot tell" in response to our self-efficacy items, leading to an exclusion from the analysis presented here. In total we include 299 valid crowd UX assessments for analysis.

4.1 Reliability of UX Items

The reliability of UX items within the item pairs and within the UX factors were calculated using Cronbach's Alpha. Alpha values over 0.5 can be interpreted as *moderate*, over 0.8 as *good* or *high*, and over 0.9 as *very good* or *very high* consistency, whereas values below 0.5 are commonly seen as indicating *bad* or *low* consistency.

For our first analysis, we did not differentiate between segments nor scenarios. On this level, four out of five item pairs in *acceptability* have moderate (0.73, 0.77) or high (0.87, 0.85) consistencies, leading to an overall excellent joint reliability (0.94) concerning the four acceptability item pairs A1–A4. A5 shows low consistency (0.41). Hypothetically, this may be due to A5 items not being semantically strictly

[2] https://www.crowdee.com.

biuniquely inverted, i.e., the opposite of *too long* might be the suggested *too short*, but it might as well be another positioning like *just fine* or *long enough*. Future experiments will revisit these items. As for *task efficiency*, two out of three item pairs have high (0.78, 0.89) and one pair has a moderate (0.57) consistency, leading to a high joint reliability (0.89) in the factor of task efficiency. As for *ease of use*, both two item pairs have moderate (0.69, 0.67) consistencies, leading to a high joint reliability (0.80) in the factor of ease of use. *System error* and *naturalness* could be assessed with high (0.83, 0.81) consistencies, *promoter score* achieved a moderate (0.76) consistency. Eventually, our item pair suggested measuring personality showed a low (0.41) consistency. Similar to the results on A5 reported above, these items were borrowed from other questionnaires and should be revised in future studies. Again, the concepts of *impoliteness* and *friendliness* must not necessarily be semantically understood as biuniquely opposites in our scenario.

When comparing these overall consistencies with consistencies on individual scenario levels, results show only minor deviations. Hence, the overall consistency does not seem to depend on our scenario design in the first place, but rather reflects general characteristics of the consistency.

4.2 UX and Self-efficacy

In order to analyse UX dependency on self-efficacy, we clustered the participants based on their answers to our self-efficacy items, applying a split by the median of the ratings in order to differentiate a *high* self-efficacy from a *low* self-efficacy group. Non-parametric Mann-Whitney-U tests ($p < 0.05$) show that this group membership imposes a significant difference on the expected UX assessment and the users' UX expectations towards the chatbot interaction for all our scenarios.

Most notably, participants in the high group of SE4 (*stopped the interaction*), as well as on SE3 (*interaction not worthwhile*) show significantly different assessments for all assessed UX factors, i.e. *acceptability, task efficiency, system error, ease of use, naturalness, personality* and *promoter score* when compared to the low group. Figure 1 visualizes this finding for segmentation using SE3 across our 14 UX pairs. In more detail, concerning acceptability, low-group participants significantly judged the chatbot to be more helpful, less frustrating, less unpleasant, and less cumbersome to use. These users were more satisfied with the chatbot, could interact more efficiently with it, and perceived its course as more smooth. Regarding task efficiency, participants in the low group significantly judged the answers and solutions proposed by the chatbot to be more clear, and misunderstandings could be cleared more easily. They did not expect more help from the system to solve their tasks and judged the system to perform better in providing all relevant information. Concerning system error, participants in the low group felt themselves significantly better understood by the chatbot. For ease of use, participants in the low group rated significantly higher on ease of use. They further reported to have obtained all information they needed easily while knowing and understanding the (expected) behaviour of the chatbot. For

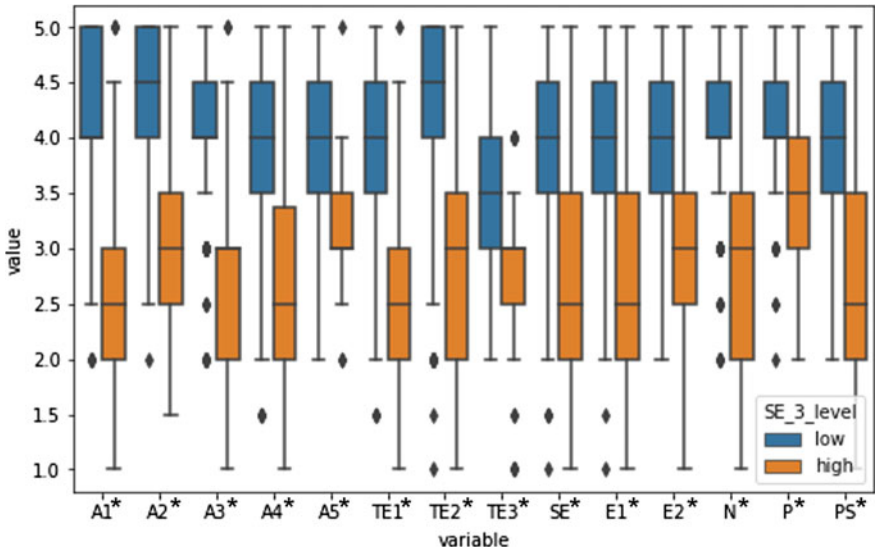


Fig. 1 Boxplot of UX pairs segmented by high and low self-efficacy with SE3. The higher the value, the more positive the UX assessment, with points illustrating outliers and "*" denoting sign. differences in between the groups

naturalness and personality, these participants significantly judged the interaction to be more natural, more friendly, and less impolitely. Finally, concerning promoter score, participants of the low group agreed significantly more to recommend the chatbot to friends and customers.

Likewise, participants in the high group of SE2 (*inclined to upgrade for more enriching or satisfactory experience*), as well as on SE1 (*inclined to switch to competitor upon bad UX*) show significantly different assessments for most of the assessed UX factors except *task efficiency and personality* when compared to the low group. Figure 2 visualizes these findings. More concretely, concerning acceptability, high-group participants significantly judged the chatbot to be less frustrating and less cumbersome to use. These users were more satisfied with the chatbot and they perceived its course as more smooth.

Eventually, a targeted adaptation aiming to increase the UX of a given chatbot for all users means to segment users by their self-efficacy and target different groups differently, e.g., through adaptation means like more help-providing functions and more course-smoothing flow options due to users' self-efficacy preferences.

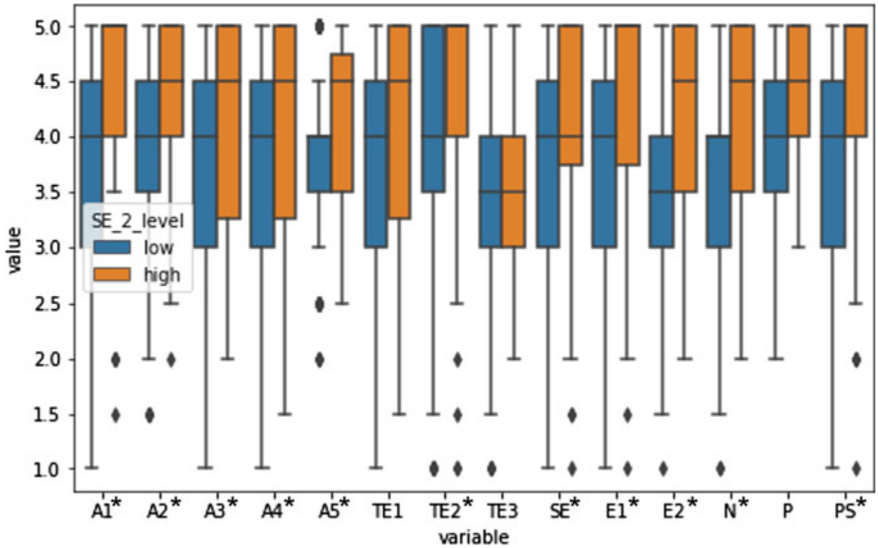


Fig. 2 Boxplot of UX pairs segmented by high and low self-efficacy with SE2. The higher the value, the more positive the UX assessment, with points illustrating outliers and "*" denoting sign. differences in between the groups

5 Discussion and Conclusion

In this paper, we demonstrated the impact of user segmentation by self-efficacy for UX expectations and preferences in a chatbot domain. Our results clearly show a significant difference in UX ratings depending on the self-efficacy of the user, strongly suggesting self-efficacy as a differentiator for chatbot personalization and user adaptation in the technical customer service domain. Reporting on several UX factors such as *acceptability, task efficiency, system error, ease of use, naturalness, personality* and *promoter score*, we found the vast majority of significant differences to be independent towards 3 chosen scenarios and individual self-efficacy item. However, as comprehensive support chatbots may easily comprise more than 100 intents, and users were free to use their own wording when conversing with the chatbot which may cause different dialog paths, a more comprehensive range of task-based scenarios (potentially sub-grouped by type of scenario) would be advisable in order to systematically further verify these findings. We aim to make available a more comprehensive data collection in the future. Also, crowd-worker users were not recruited to be actual customers, which is scheduled for future work, potentially influencing the way of conversation. Analysing UX of chatbots from other domains (e.g., medical, sales, etc.) remains future work. Finally, in this work, we focused on a personally enriching, satisfactory, or worthwhile chatbot experience for the user to relate to the user's loyalty and upgrade willingness, while more psychological definitions of self-efficacy relating to the belief in ability to succeed at a task may extent the scope towards other applications. Finally, other factors like information savviness and device usage also remain future work.

References

1. Banchs RE (2016) Expert-generated vs. crowd-sourced annotations for evaluating chatting sessions at the turn level. In: WOCHAT: second work-shop on chatbots and conversational agent technologies, IVA 2016
2. Bruun A, Stage J (2015) New approaches to usability evaluation in software development: Barefoot and crowdsourcing. J Syst Softw 105:40–53
3. Deriu J, Rodrigo A, Otegi A, Echegoyen G, Rosset S, Agirre E, Cieliebak M (2021) Survey on evaluation methods for dialogue systems. Artif Intell Rev 54(1):755–810
4. Finstad K (2010) The usability metric for user experience. Interact Comput 22(5):323–327
5. Freter H (2008) Markt-und Kundensegmentierung: kundenorientierte Markterfassung und-bearbeitung, vol 6. W. Kohlhammer Verlag
6. Gomide VH, Valle PA, Ferreira JO, Barbosa JR, Da Rocha AF, Barbosa T (2014) Affective crowdsourcing applied to usability testing. Int J Comput Sci Inf Technol 5(1):575–579
7. Hoßfeld T, Keimel C, Hirth M, Gardlo B, Habigt J, Diepold K, Tran-Gia P (2013) Best practices for qoe crowdtesting: qoe assessment with crowdsourcing. IEEE Trans Multimedia 16(2):541–558
8. Iskender N, Polzehl T, Möller S (2020) Crowdsourcing versus the laboratory: towards crowd-based linguistic text quality assessment of query-based extractive summarization. In: Proceedings of the conference on digital curation technologies (Qurator 2020). CEUR, pp 1–16
9. Iskender N, Polzehl T, Möller S (2020) Towards a reliable and robust methodology for crowd-based subjective quality assessment of query-based extractive text summarization. In: Proceedings of the 12th LREC. European Language Resources Association, pp 245–253
10. ISO (2010) Ergonomics of human system interaction-part 210: Human-centred design for interactive systems (formerly known as 13407). Standard ISO DIS 9241-210, International Organization for Standardization, Switzerland
11. ITU-T (2003) Subjective quality evaluation of telephone services based on spoken dialogue systems. ITU-T Rec. P.851, International Telecommunication Union, Geneva
12. Kittur A, Chi E, Suh B (2008) Crowdsourcing for usability: using micro-task markets for rapid, remote, and low-cost user measurements. In: Proceedings of the CHI 2008
13. Lai A (2016) The rise of the empowered customer. Technical report, Forrester Research, Inc., 60 Acorn Park Drive, Cambridge, MA 02140 USA
14. Law ELC, Roto V, Hassenzahl M, Vermeeren AP, Kort J (2009) Understanding, scoping and defining user experience: a survey approach. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 719–728
15. Liu D, Bias RG, Lease M, Kuipers R (2012) Crowdsourcing for usability testing. Proc Amer Soc Inf Sci Technol 49(1):1–10
16. Meffert H, Burmann C, Kirchgeorg M, Eisenbeiß M (2018) Marketing: Grundlagen marktorientierter Unternehmensführung Konzepte–Instrumente–Praxisbeispiele. Springer
17. Möller S, Smeele P, Boland H, Krebber J (2007) Evaluating spoken dialogue systems according to de-facto standards: a case study. Comput Speech & Lang 21(1):26–53
18. Nebeling M, Speicher M, Norrie MC (2013) Crowdstudy: general toolkit for crowdsourced evaluation of web interfaces. In: Proceedings of the 5th ACM SIGCHI symposium on engineering interactive computing systems, pp 255–264
19. Quarteroni S, Manandhar S (2009) Designing an interactive open-domain question answering system. Nat Lang Eng 15(1):73

20. Reichheld F (2011) The ultimate question 2.0 (revised and expanded edition): how net promoter companies thrive in a customer-driven world. Harvard Business Review Press
21. Rogers CR (1951) Client-centered therapy: its current practice, implications, and theory, with chapters. Houghton Mifflin Oxford, United Kingdom
22. Yu Z, Xu Z, Black AW, Rudnicky A (2016) Chatbot evaluation and database expansion via crowdsourcing. In: Proceedings of the chatbot workshop of LREC, vol 63, p 102

Learning to Ask Specific Questions Naturally in Chat-Oriented Dialogue Systems

Sota Horiuchi and Ryuichiro Higashinaka

Abstract In order for a dialogue system to provide information tailored to the user, it is important to ask users questions and obtain the necessary information. However, asking questions abruptly or in a self-centered manner would be undesirable because it may disrupt the flow of conversation and decrease the user's satisfaction. In this work, we propose a response generation model for a chat-oriented dialogue system that can ask specific questions naturally. Specifically, we train a response generation model that generates utterances on the basis of both the dialogue context and the question to be asked. The results of simulations and human evaluations demonstrate that the proposed model can make it easier for a system to ask specific questions while maintaining the naturalness of the dialogue.

1 Introduction

In both task-oriented [27, 30] and non-task-oriented dialogue systems [1, 20], it sometimes necessary to ask the user questions in order to elicit necessary information. However, the system should not ask questions in a self-centered manner or without considering the context because this would make the dialogue unnatural and the user would not be able to interact with the system comfortably. In order for the system to ask specific questions naturally, the context of the dialogue must be such that the user feels it is natural for such a question to be asked.

In this work, we propose a response generation model for a chat-oriented dialogue system that enables it to ask specific questions naturally. To construct response generation models for asking a specific question, we first create a corpus by extracting dialogue contexts that contain specific questions and then use the corpus to build response generation models that take into account both the context and the question

S. Horiuchi (✉) · R. Higashinaka
Graduate School of Informatics, Nagoya University, Nagoya, Japan
e-mail: horiuchi.sota.n4@s.mail.nagoya-u.ac.jp

R. Higashinaka
e-mail: higashinaka@i.nagoya-u.ac.jp

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_19

to be asked. We conducted dialogue simulations between the dialogue systems and human evaluations on the dialogue and utterance levels and found that the proposed models were able to ask questions more naturally than a baseline.

2 Related Work

One of the typical systems that require natural questioning is the interview system. Most of the conventional interview systems focus on eliciting the user's information accurately, and ask the user predefined questions [6, 9, 10]. For example, Kahn et al. [10] developed an interview dialogue system that aims to acquire knowledge about medical diagnosis by interviewing medical experts and asking them questions about the name and characteristics of the disease. In addition to predefined questions, generating responses adaptively through follow-up questions [7] or small talk based on the response of the participant [13] has been studied. While these studies are similar to ours in terms of asking questions, they differ in that we guide the dialogue to ask a specific question and endeavour to make that question natural within the context of the conversation.

In this study, we use a seq2seq model [3, 23] to generate responses for the dialogue systems. With the recent development of neural dialogue response generation techniques, ancillary information can be given as input to the model to enrich the responses it generates [2, 5, 8, 28]. Following these studies, we create models that guide the dialogue towards asking specific questions by inputting both the context and the question to be asked in the future.

There have been a few studies on guiding the flow of dialogue. One method aims to control the flow of dialogue by generating utterances based on a keyword that can be transitioned from the keywords in the current context [25, 29]. Another method predicts the future context by multi-turn beam search and chooses the best context that fits the system's goal [12, 14]. Since our method only manipulates the input for response generation, these methods can easily be used in conjunction with our proposed model, which we intend to investigate in future work.

3 Proposed Model

Figure 1 shows the overview of how to create the proposed model. We first create a Question-Guiding corpus to identify dialogue contexts in which it is acceptable to ask specific questions. To create this corpus, we come up with a target question list from an existing corpus and use it to extract dialogues from that corpus. Then, we train the response generation models by fine-tuning a pre-trained response generation model with the Question-Guiding corpus. In this section, we describe the details.

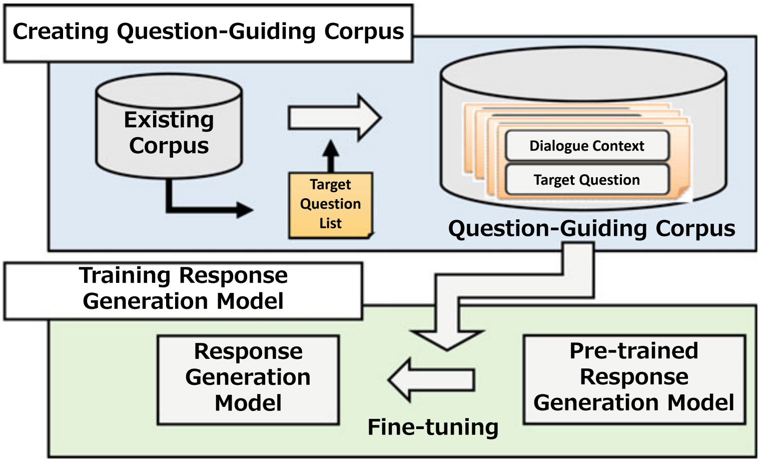


Fig. 1 Overview of how to create the proposed model

3.1 Creating Question-Guiding Corpus

To enable the models to ask the target questions, it is necessary to identify the contexts in which it is acceptable to ask such questions. To identify such contexts, we create the Question-Guiding corpus (QGC). The QGC consists of dialogue contexts and the target questions asked in those contexts. It is created from an existing dialogue corpus using the following three steps.

1. Extract questions from an existing corpus. The questions to be extracted are those that are asked several utterances after the beginning of a dialogue, as we want to extract questions that require some context to ask. In order to select utterances that are explicit questions, we extract utterances that end with "?". We do not extract questions that can be used in any context, such as "Really?" or "What do you mean?". This checking is done manually.
2. Create a target question list by extracting the top N questions with high frequency from the extracted questions. We focus on the top N because our interest is limited to the frequent ones so as to reduce noise.
3. For each of the extracted questions in the list, identify the dialogue context in which the question appears at the end of an utterance from the corpus, and extract utterances from the beginning of the dialogue to the utterance containing the question as one dialogue.

3.2 Training Response Generation Model

We adopt an encoder-decoder model based on a Transformer with an attention mechanism [26], which is a commonly used architecture for response generation models.

There are two types of question information to be put into the model: "target question" and "number of turns to ask the target question". The "target question" is a question that the system wants to ask in the future in a dialogue. The "number of turns to ask the target question" refers to the total number of utterances from the current turn until the system asks the target question; this is used to force the system to ask the target question at a certain point in the dialogue.

When the model is trained to generate a response at turn $t + 1$, we use the target question Q, the number of turns T before the target question is asked, and the dialogue context C. Conditioned on T, Q, C, our model generates a response R that maximizes the following probability:

$$P(R|T, Q, C) = \prod_{i=1}^{|R|} P(r_i|T, q_1, \ldots, q_{|Q|}, c_1, \ldots, c_{|C|}, r_1, \ldots, r_{i-1}),$$

where q_i is the ith word in Q, c_i is the ith word in C, and r_i is the ith word in R. T, Q, and C are separated and combined by a special token, [DELIM], which also separates each utterance in a dialogue context.

In order to create a model that can generate responses of sufficient quality, a large amount of data is typically required. Therefore, as a base model, we use a pre-trained language model similar to T5 [18] or BART [15] that has already been trained from a large amount of data and is capable of generating sufficiently high-quality responses from the dialogue context.

4 Experiment

4.1 Question-Guiding Corpus

We created the QGC using dialogue data from the PersonaChat dataset [31] as an existing corpus. PersonaChat was selected because it focuses on asking about the personality of the other person, and therefore contains many questions. The number of target questions, N, was set to 200, and a list of target questions was created from PersonaChat. For evaluation (see Sects. 4.4 and 4.5), we randomly divided the target questions into two sets with the ratio of 1:1 and used one for training and the other for testing. Using the target questions for training, we created the QGC by extracting the dialogue context from PersonaChat.

In total, we extracted 2,352 dialogues. The number of utterances was 38,968, the number of words was 430,942, and the number of unique words was 9,116. The

Table 1 Examples of target questions

What do you do for a living?
Do you have any hobbies?
What city are you from?
Do you have any pets?
What do you do for work?
What do you do for fun?
Do you have any kids?
What is your favourite colour?
What kind of music do you like?
What is your favourite food?

Table 2 Example of dialogue in QGC (bold text indicates the target question)

Target question: Where do you live now?

Speaker	Utterance
P1	Hello, how are you this morning?
P2	Good just back from my daily run
P1	I used to run, I am practicing my song for the talent show
P2	Nice I love running and yoga but hate vegetables
P1	I do not like veggies either. I am very shy, are you an outgoing person?
P2	Yeah my strength makes me a big help at the dog shelter
P1	I love dogs! do you have one?
P2	I've 2 dogs but I am so tall we can't all fit on couch
P1	That is too bad, I do not have one yet
P2	Aww everyone should have a dog
P1	I agree, we never had one when I was a kid. Not enough room
P2	I grew up in apartments so I understand
P1	**Where do you live now?**

average number of utterances at which the target question was asked in the corpus was 7.15. Table 1 shows part of the list of target questions, and Table 2 shows an example dialogue contained in the QGC.

4.2 *Training Response Generation Model*

We used the pre-trained model of BlenderBot [20] (Reddit2.7B) before it is fine-tuning with blending skills, that is, only pre-trained with Reddit data. For the fine-tuning, we used five datasets: PersonaChat [31], EmpatheticDialogues [19], Wizard of Wikipedia [4], BlendedSkillTalk [22] (all of which are used in BlenderBot), and QGC. For training, we used the standard Maximum Likelihood Estimation (MLE) approach. The loss weights for each task during training were the ratios used in BlenderBot for its four tasks and the ratio of the loss in the QGC was the sum of the four tasks. Specifically, as ratios, we used PersonaChat = 3, EmpatheticDialogues = 3, Wizard of Wikipedia = 3, BlendedSkillTalk = 1, and QGC = 10. These ratios were set heuristically to ensure sufficient weights for our task, that is, to ask questions naturally.

In the QGC, we randomly used 90% of the data for training and the remaining 10% for validation. The hyperparameters for the training and models were taken from BlenderBot. Sentences were tokenized using Byte Pair Encoding [21] with the tokenizer used in BlenderBot. Learning was done in mini-batches with the mini-batch size of 64. We used Adam [11] for optimizing the learning parameters. Learning rate was 7e-6 and leaning rate scheduler was ReduceOnPlateau. The training was performed by measuring the perplexity of the validation data after each round of training, and we used early stopping (with the patience of 5 epochs). The model with the lowest perplexity was used for the experiments.

To infer the response, we used beam search with a width of 10. The beam blocking of n-gram (n = 3) [17] was adopted. As in BlenderBot, the blocking was used for both the input dialogue context and the generated utterance.

4.3 Comparison Models

We compared the baseline model with two variants of the proposed model, as described below.

Baseline The base model (Reddit2.7B) is fine-tuned using only the four datasets used in BlenderBot; this model is the same as the BlenderBot2.7B shared publicly.[1] The model generates the next utterance by using only the dialogue context. In order to make the response include the target question, the utterance at turn 7, which is close to the average number of turns where the target question appears in QGC, is replaced by the target question.

TargetQ (proposed) The model is fine-tuned using the five datasets including QGC to generate the next utterance from the input with both the dialogue context and the target question.

TargetQ+Turn (proposed) The model is fine-tuned using the five datasets including QGC to generate the next utterance by adding the target question and the number of turns to ask the target question to the dialogue context. The number of turns was set to 7, the same as in the baseline. After the specified turn, question information is not added as input to the model.

[1] https://parl.ai/projects/recipes/.

In both Baseline and TargetQ+Turn, the target question is output on a fixed turn. However, there is no control over the number of turns in TargetQ. In the evaluation of TargetQ, the question may come out early in the dialogue, which might provide an unfair advantage because it allows TargetQ to generate target questions before enough context has been built. Therefore, we only allowed TargetQ to utter the target question after the seventh utterance by adjusting the beam blocking.

4.4 Evaluation by Dialogue Simulation

In order to evaluate the models for comparison, we simulated dialogues between the response generation models. Specifically, dialogues were conducted between each of the models to be compared and BlenderBot. A total of 100 dialogues was conducted. Each dialogue started with a fixed utterance of "Hello" from the comparison models. Excluding the fixed utterance at the beginning, the comparison models and BlenderBot each generate five utterances before the dialogue ends. Table 3 shows an example of a simulated dialogue.

4.4.1 Evaluation Metrics

The following two metrics were used to evaluate the simulated dialogues.

Target question implementation rate The percentage of dialogues in which the model output the target question. Note that we do not count the cases when the baseline is forced to output the target question. The higher the value, the better the evaluation result.

Table 3 Example dialogue (the target question is in bold)

Target question: What is your favourite fruit?

Speaker	Utterance
TargetQ	Hello
BlenderBot	Hi, how are you doing?
TargetQ	I'm good, thanks for asking
BlenderBot	What do you do for a living?
TargetQ	I work at a grocery store
BlenderBot	Do you like it?
TargetQ	Yeah it pays the bills
BlenderBot	Do you have to interact with people often?
TargetQ	Yes I do. **What is your favourite fruit?**

Table 4 Results of target question implementation rate and naturalness of target question. The best value for each column is in bold

Model	Target question implementation rate (%)	Naturalness of target question (perplexity)
Baseline	0	10.79
TargetQ	59	**6.822**
TargetQ+Turn	**71**	8.515

Naturalness of target question The average of the perplexity from the context to the target question, where perplexity is calculated as the inverse of the probability of generating the target question for the context. The smaller the perplexity value, the more natural the target utterance is to the context.

The "Naturalness of target question" cannot be measured for dialogues in which the model does not ask the target question. Therefore, when there was no utterance containing the target question, we calculated the perplexity by assuming that the system uttered the target question at the seventh utterance.

4.4.2 Results

Table 4 lists the evaluation results of the dialogue simulation. In terms of the target question implementation rate, TargetQ and TargetQ+Turn generated the target questions in 59% and 71% of the dialogues, respectively, whereas the baseline could not generate the target question at all. We found that both variants of the proposed model were able to generate the target questions and that we could force the target question to be output by inputting the number of turns. In terms of naturalness of the target question, we found that both proposed models could generate target questions more naturally in context compared to the baseline.

We also found that the naturalness was better when the number of turns was not input. This can be attributed to the fact that by inputting the number of turns, the model generates the target question a little too forcefully for a dialogue context that is not sufficient to accommodate the target question. In other words, although the number of turns is effective to a certain extent in generating the target question, the downside is that the model outputs the target question even if it is not contextually natural.

Table 5 Example dialogue (the target question is in bold)

Target question: What kind of books do you like to read?	
Speaker	Utterance
Human	Hi there
TargetQ	Hello, how are you?
Human	I'm well, how are you?
TargetQ	I'm good, thanks for asking
Human	It's been really hot weather we've been having, hasn't it?
TargetQ	Yes it has
Human	What do you do for a living?
TargetQ	I work at a bookstore
Human	Oh nice, I love books
TargetQ	**What kind of books do you like to read?**

4.5 Human Evaluation

4.5.1 Subjective Evaluation of Dialogue

In order to conduct the human evaluation, we first collected data on the dialogue between the model and a human and then used that data to conduct an evaluation that focused on the target question. For the dialogue data collection, as in the dialogue simulation, the target questions were randomly selected before the dialogue began. In the baseline, the seventh utterance was replaced by the target question, as in the simulation. The dialogue was started by the user. Table 5 shows an example of a dialogue.

We utilized Amazon Mechanical Turk to recruit 50 people to interact with the comparison models. Before participating in the experiment, the Turkers were given a brief screening in which they were asked questions selected from CommonsenseQA [24]. The Turkers interacted with each of the three comparison models in random order. The dialogue took seven turns per model (for a total of 14 utterances in a dialogue). The dialogue interface used was ParlAI's[2] Chat service. The dialogue experiment took about 15 minutes to complete, and we rewarded each participant $2. Note that this experiment, together with subsequent ones involving human evaluation, was approved by the research ethics committee of Nagoya University.

Table 6 lists the results of the target question implementation rate in human interaction. TargetQ and TargetQ+Turn generated the target question themselves in 64% and 84% of the interactions, respectively. TargetQ+Turn generated the target question more often than the other models, as in the simulation.

After having collected the dialogues of the comparison models, we evaluated the naturalness of the target questions. For this evaluation, we used only those dialogues

[2] https://parl.ai/.

Table 6 Results of target question implementation rate in human experiment. The best value for the column is in bold

Model	Target question implementation rate
Baseline	0
TargetQ	64%
TargetQ+Turn	**84**%

Table 7 Results of naturalness of entire dialogue and naturalness of question in context. Values represent average ratings. Rows with ∗∗ ($p < 0.01$) are statistically significant compared to Baseline (Wilcoxon signed-rank test with Bonferroni correction). The best value for each column is in bold

Model	Naturalness of entire dialogue	Naturalness of question in context
Baseline	5.37	3.64
TargetQ	5.53	**4.70****
TargetQ+Turn	**5.58**	4.35**

in which the model asked the target question. The evaluation was performed on the context from the start of the dialogue to the target question. We again used Amazon Mechanical Turk for this evaluation, and a simple screening was performed, as in the data collection. Each Turker read the dialogues and rated each dialogue and target question located at the end of the dialogue context. Nine Turkers were recruited for this evaluation, in which each dialogue was evaluated by five different Turkers. Each Turker was paid $2 for this evaluation.

4.5.2 Evaluation Metrics

The evaluation focused on the following two items (1 indicates strongly disagree, 7 indicates strongly agree).

Naturalness of entire dialogue We asked participants to judge the dialogue naturalness by indicating their level of agreement with "The overall dialogue is natural."

Naturalness of question in context We asked participants to judge the naturalness of the context flow by indicating their level of agreement with "The question posed by the system at the end of the conversation is appropriate to the context."

4.5.3 Results

Table 7 shows the results of the dialogue evaluation. In terms of the naturalness of the question in context, the proposed model was statistically superior to the Baseline.

Table 8 Results of naturalness of entire dialogue when target question is asked after user's non-question and question. The best value for each column is in bold

Model	Naturalness of entire dialogue when target question is asked after user has asked a question.	Naturalness of entire dialogue when target question is asked after user has uttered a non-question.
Baseline	4.90	5.66
TargetQ	**6.25**	5.25
TargetQ+Turn	5.42	**5.69**

Table 9 Results of naturalness of question in context when target question is asked after user's non-question utterance. The best value for each column is in bold

Model	Naturalness of question in context when target question is asked after user has asked a question.	Naturalness of question in context when target question is asked after user has uttered a non-question.
Baseline	2.33	4.47
TargetQ	**4.34**	**4.73**
TargetQ+Turn	3.81	4.56

In contrast, there was no statistically significant difference in the overall naturalness of the dialogue.

As for why the baseline was weak, we consider it was possibly due to the fact that the baseline asks the target questions at a fixed turn, often responding to a question from the user with a question, which degrades the naturalness considerably. Therefore, we checked the difference in the evaluation value in terms of whether the target question was asked after a user's question or non-question.

Tables 8 and 9 show the differences in evaluation values according to the content of the user's response before the target question. The Baseline model did not perform well when the target question was followed by the user's question. In contrast, the proposed model, especially TargetQ, showed some decrease in values in the same situation, but not as much as Baseline. The results of Baseline demonstrate that asking the target question immediately after the user asks a question has a negative impact on the naturalness of the question in the context. However, the proposed model scored relatively well on the naturalness of the question in the context even when the target question was asked immediately after the user asked the question. This suggests that the proposed model guides the dialogue flow in such a way that the system can ask the target question. On the other hand, the naturalness of the entire dialogue was not significantly different from that of the Baseline. This suggests that even if the target question is somewhat unnatural, the effect on the naturalness of the entire dialogue is relatively small as long as the other utterances are appropriate.

5 Conclusion

In this study, we developed response generation models that can naturally ask specific questions in a chat-oriented dialogue system. We first created a corpus for training and used it to train the response generation models to ask a target question. We then used these learned response generation models in dialogue simulations with another response generation model and in dialogue experiments with humans. The results showed that the proposed model can ask the target question more naturally than the model that responds without considering the target question.

Our future work will include a more detailed analysis of the data obtained from the evaluation experiments and a deeper investigation into how humans are guided and to what extent it is possible for humans to ask questions naturally. In addition, it should be possible to apply reinforcement learning [16] to our problem, as our objective relates to the behaviour in consideration of some future state. We will also investigate how to create a system that can naturally elicit information from the user and make better recommendations.

Acknowledgements We thank the anonymous reviewers for their helpful comments and suggestions. Funding was provided by a Grant-in-Aid for Scientific Research (Grant no. JP19H01125).

References

1. Adiwardana D, Luong M, So DR, Hall J, Fiedel N, Thoppilan R, Yang Z, Kulshreshtha A, Nemade G, Lu Y, Le QV (2020) Towards a human-like open-domain chatbot. arXiv:2001.09977
2. Budzianowski P, Vulić I (2019) Hello, it's GPT-2 - how can I help you? Towards the use of pretrained language models for task-oriented dialogue systems. In: Proceedings of the 3rd workshop on neural generation and translation, pp 15–22
3. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (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, pp 1724–1734
4. Dinan E, Roller S, Shuster K, Fan A, Auli M, Weston J (2019) Wizard of Wikipedia: Knowledge-powered conversational agents. In: Proceedings of the 7th international conference on learning representations
5. Ham D, Lee JG, Jang Y, Kim KE (2020) End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 583–592
6. Hoque ME, Courgeon M, Martin JC, Mutlu B, Picard RW (2013) Mach: my automated conversation coach. In: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing, pp 697–706
7. Inoue K, Hara K, Lala D, Nakamura S, Takanashi K, Kawahara T (2019) A job interview dialogue system with autonomous android ERICA. In: Marchi E, Siniscalchi SM, Cumani S, Salerno VM, Li H (eds) Proceedings of the 10th international workshop on spoken dialogue systems, vol 714, pp 291–297
8. Johnson M, Schuster M, Le QV, Krikun M, Wu Y, Chen Z, Thorat N, Viégas F, Wattenberg M, Corrado G, Hughes M, Dean J (2017) Google's multilingual neural machine translation system: enabling zero-shot translation. Trans Assoc Comput Linguist 5:339–351

9. Johnston M, Ehlen P, Conrad FG, Schober MF, Antoun C, Fail S, Hupp A, Vickers L, Yan H, Zhang C (2013) Spoken dialog systems for automated survey interviewing. In: Proceedings of the 2013 annual meeting of the special interest group on discourse and dialogue, pp 329–333
10. Kahn G, Nowlan S, Mcdermott J (1985) More: an intelligent knowledge acquisition tool. In: Proceedings of the 9th international joint conference on artificial intelligence, pp 581–584
11. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
12. Kishinami Y, Akama R, Sato S, Suzuki J, Tokuhisa R, Inui K (2021) Data-oriented approach for lookahead response generation. In Proceedings of the 2021 annual conference of JSAI, p 3J2GS6b02. (in Japanese)
13. Kobori T, Nakano M, Nakamura T (2016) Small talk improves user impressions of interview dialogue systems. In: Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue, pp 370–380
14. Kulikov I, Lee J, Cho K (2019) Multi-turn beam search for neural dialogue modeling. arXiv:1906.00141
15. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2020) BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 7871–7880
16. Li J, Monroe W, Ritter A, Jurafsky D, Galley M, Gao J (2016) Deep reinforcement learning for dialogue generation. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 1192–1202
17. Paulus R, Xiong C, Socher R (2017) A deep reinforced model for abstractive summarization. arXiv:1705.04304
18. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 21(140):1–67
19. Rashkin H, Smith EM, Li M, Boureau YL (2019) Towards empathetic open-domain conversation models: a new benchmark and dataset. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 5370–5381
20. Roller S, Dinan E, Goyal N, Ju D, Williamson M, Liu Y, Xu J, Ott M, Shuster K, Smith EM, Boureau Y, Weston J (2020) Recipes for building an open-domain chatbot. arXiv:2004.13637
21. Sennrich R, Haddow B, Birch A (2016) Neural machine translation of rare words with subword units. In: Proceedings of the 54th annual meeting of the association for computational linguistics, vol 1, pp 1715–1725
22. Smith EM, Williamson M, Shuster K, Weston J, Boureau YL (2020) Can you put it all together: evaluating conversational agents' ability to blend skills. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 2021–2030
23. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Proceedings of the 27th neural information processing systems, pp 3104–3112
24. Talmor A, Herzig J, Lourie N, Berant J (2019) CommonsenseQA: a question answering challenge targeting commonsense knowledge. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 4149–4158
25. Tang J, Zhao T, Xiong C, Liang X, Xing E, Hu Z (2019) Target-guided open-domain conversation. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 5624–5634
26. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Lu, Polosukhin I (2017) Attention is all you need. In: Proceedings of the 2017 advances in neural information processing systems, vol 30
27. Wen TH, Vandyke D, Mrkšić N, Gašić M, Rojas-Barahona LM, Su PH, Ultes S, Young S (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, vol 1, pp 438–449

28. Xing C, Wu W, Wu Y, Liu J, Huang Y, Zhou M, Ma WY (2017) Topic aware neural response generation. In: Proceedings of the 31th AAAI conference on artificial intelligence, pp 3351–3357
29. Xu J, Wang H, Niu Z, Wu H, Che W (2020) Knowledge graph grounded goal planning for open-domain conversation generation. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 9338–9345
30. Yang Y, Li Y, Quan X (2020) UBAR: towards fully end-to-end task-oriented dialog systems with GPT-2. arXiv:2012.03539
31. Zhang S, Dinan E, Urbanek J, Szlam A, Kiela D, Weston J (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, vol 1, pp 2204–2213

Fine-Tuning a Pre-trained Transformer-Based Encoder-Decoder Model with User-Generated Question-Answer Pairs to Realize Character-Like Chatbots

Koh Mitsuda, Ryuichiro Higashinaka, Hiroaki Sugiyama, Masahiro Mizukami, Tetsuya Kinebuchi, Ryuta Nakamura, Noritake Adachi, and Hidetoshi Kawabata

Abstract In order to realize character-like chatbots, it is necessary to collect dialogue data of particular characters. However, collecting such data is not an easy task. To solve this problem, we have previously proposed a method called "Role-play-based question answering" in which many users play the role of a particular character to answer questions, resulting in a large number of question-answer (QA) pairs associated with that character. In this study, we investigated how character-like dialogue could be realized by fine-tuning a pre-trained Transformer-based encoder-decoder model, which has shown its effectiveness in dialogue modelling, with the QA pairs collected via role-play-based question answering. The results of automatic

K. Mitsuda (✉) · R. Higashinaka · T. Kinebuchi
NTT Human Informatics Laboratories, NTT Corporation, Kanagawa, Japan
e-mail: koh.mitsuda.td@hco.ntt.co.jp

R. Higashinaka
e-mail: ryuichiro.higashinaka.tp@hco.ntt.co.jp

T. Kinebuchi
e-mail: tetsuya.kinebuchi.xh@hco.ntt.co.jp

H. Sugiyama · M. Mizukami
NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
e-mail: hiroaki.sugiyama.kf@hco.ntt.co.jp

M. Mizukami
e-mail: masahiro.mizukami.df@hco.ntt.co.jp

R. Nakamura · N. Adachi · H. Kawabata
DWANGO Co., Ltd., Tokyo, Japan
e-mail: ryuuta_nakamura@dwango.co.jp

N. Adachi
e-mail: noritake_adachi@dwango.co.jp

H. Kawabata
e-mail: hidetoshi_kawabata@dwango.co.jp

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_20

and manual evaluations show that, with the fine-tuned model, it is possible to significantly outperform a retrieval-based baseline and that, with 44 K QA pairs, it is possible to achieve high naturalness and characterness scores.

1 Introduction

Research on chatbots with consistent personalities has received considerable attention in recent years [12, 14, 23, 24]. Creating such chatbots is typically done by preparing a large amount of data on a specific character and then training a generation-based model to respond like that character. Unfortunately, collecting a large amount of data focused on one particular character is not easy. To solve this problem, we previously proposed a method called "Role-play-based question answering" and used it to efficiently collect question-answer (QA) pairs related to famous characters from their fans [6, 9, 10]. We also proposed a method that enables chatbots to talk like these characters by a retrieval-based method using the collected QA pairs.

In recent years, the language models used in natural language processing have become more and more sophisticated [2, 3, 15, 22]. In particular, Transformer-based encoder-decoder models trained on large-scale dialogue corpora have been used to create chatbots such as Meena [1] and BlenderBot [17] that can interact with users much more fluently than conventional chatbots. In light of the remarkable progress of the pre-trained Transformer-based encoder-decoder model, we hypothesize that fine-tuning a pre-trained model with collected QA pairs for a particular character could enable not only natural but also character-like dialogue in which users will feel as if they are actually interacting with the character.

In this study, we investigate whether a character-like chatbot can be created by fine-tuning a pre-trained Transformer-based encoder-decoder model using a large number of collected QA pairs for a particular character by role-play-based question answering. To evaluate the quality of the fine-tuned model, we conducted both automatic and manual evaluations on both the utterance-level and dialogue-level in which workers interacted with chatbots. The results indicate that the fine-tuned model significantly outperformed a retrieval-based baseline and that, with 44 K QA pairs, it was possible to achieve high naturalness and characterness scores.

2 Data Collection

To fine-tune the pre-trained encoder-decoder model, we first created a role-play-based QA corpus for a particular character following previous studies [9, 10]. We also created a character-dialogue corpus in which workers manually extended the QA pairs to dialogues for investigating the limitations of the models trained with the QA data only. The following sections describe the creation of the corpora.

Table 1 Statistics of role-play-based question-answer (QA) corpus for famous video game character Amadeus Kurisu

No. of users who participated	1,916
No. of QA pairs	44,805
No. of questions	23,466
No. of tokens per question	11.2
No. of letters per question	22.5
No. of unique tokens in all questions	18,132
No. of answers	43,752
No. of tokens per answer	15.3
No. of letters per answer	32.1
No. of unique tokens in all answers	22,977

2.1 Role-Play-Based QA Corpus

The role-play-based QA corpus was created by setting up a website where fans of a character could post questions to the character as well as answers to questions as if the users were actually the character. To stimulate interaction, the Web sites show the rankings of users by their number of posts. In addition, a "like" button is placed beside each answer so that when a user thinks the answer sounds very much "like" the character in question, this opinion can be reflected in the number of "likes". As a result, although the users were not paid for their work, we were able to collect a large number of QA pairs with the data collection speed being 1K pairs in 1.3 h and 10 K pairs in 16 h. The data collection was conducted in Japanese. The example QA pairs look as follows:

Q Could you tell me your name?
A_1 My name is Kurisu. I've been looking forward to meeting you.
A_2 This is Kurisu Makise. What's up?

Q and A correspond to a question and an answer. Each question can have more than one answer because users can post different answers to the same question.

Table 1 shows the statistics of the collected QA pairs for the target character, Amadeus Kurisu. This character is a famous character in the Japanese video game STEINS;GATE.[1] Around 44 K QA pairs were collected, which is quite large considering that they concern only one character. Since the fans are familiar with the personality of the character, the collected data consist of characteristic questions and answers.

[1] https://store.steampowered.com/app/412830/STEINSGATE/.

2.2 Character-Dialogue Corpus

Although role-play-based question answering is an efficient way to collect a large amount of QA pairs for a particular character, it may be inadequate in two respects: it does not include character-initiated utterances (e.g., questions from the character), and it does not include contextual data. To investigate the limitations of the role-play-based QA corpus, we collected character-like dialogue data and used it along with the QA data for fine-tuning. To collect various dialogues, we prepared two versions of the character-dialogue corpus: one in which the dialogue starts with the user, as in the role-play-based QA corpus, and one in which the dialogue starts with the character. For the user-initiated corpus, QA pairs in the role-play-based QA corpus were extended by manually adding four utterances. For the character-initiated corpus, the character's first utterance was created and then four utterances that would naturally follow that first utterance were created. As a stimulus to write the character-initiated utterances, we randomly selected utterances from a chat-oriented dialogue corpus [8] with a balanced number of dialogue acts [13]. We asked the workers to paraphrase the stimulus utterance to create the character's utterance.

For creating character-dialogue corpus, we used workers recruited from a Japanese crowdsourcing platform. In contrast to the creation of the QA corpus, the workers here were paid for every created utterance. We limited the number of workers to those who had knowledge of the character (i.e., they had seen or played at least one related game or movie in which the target character appeared). The total number of workers was nine. Each worker was assigned to different initial QA pairs or utterances to increase the variation of the collected dialogues.

Table 2 shows the statistics of the character-dialogue corpus, which was manually created using the collected QA pairs for Amadeus Kurisu. A total of 4.5 K dialogues consisting of 18 K utterances was collected. The number of tokens per utterance was smaller than that in the role-play-based QA corpus because short replies or simple questions to continue the dialogue were included. An example of a collected dialogue extended from the QA pair looks as follows.

U_1 (Q) Could you tell me your name?
S_1 (A_1) My name is Kurisu. I've been looking forward to meeting you.
U_2 I think Kurisu is a really cool name.
S_2 Really? Thank you.
U_3 I'll tell everyone about it!
S_3 Stop it! It's embarrassing!

U and S correspond to a user utterance and a character's utterance. U_1 and S_1 were those taken from a QA pair, and additional utterances from U_2 to S_3 were manually created by extending the pair.

Table 2 Statistics of character-dialogue corpus, where dialogues were manually created using QA pairs in role-play-based QA corpus for Amadeus Kurisu. User-init and Char-init correspond to user-initiated and character-initiated

	User-init.	Char-init.
No. of dialogues	2,250	2,250
No. of utterances per dialogue	6	5
No. of total utterances	9,000	9,000
No. of tokens per user utterance	7.98	7.04
No. of letters per user utterance	16.6	15.6
No. of unique tokens in all user utterances	9,213	6,058
No. of tokens per character's utterance	10.1	8.33
No. of letters per character's utterance	21.1	17.5
No. of unique tokens in all character's utterances	10,461	7,501

3 Response Generation Models

In this section, we describe the response generation model for constructing a character-like chatbot. We first explain a retrieval-based model as a baseline, which is an improved version of the method proposed by Higashinaka et al. [9]. We then outline the process of constructing the pre-trained Transformer encoder-decoder model, which is a Japanese version of the BlenderBot model trained by Sugiyama et al. [20, 21].

3.1 Retrieval-Based Model (Baseline)

Higashinaka et al. [9] proposed a method in which, given an input question, QA pairs in the role-play-based QA corpus are retrieved and ranked, and the answer of the best QA pair is used to output a system utterance. The score of each QA pair is calculated using various features in dialogue research, such as search-engine score, question type, topical word, semantic-similarity score, and translation score (which is the probability that the answer is generated from the question).

In this work, instead of Higashinaka's method based on some heuristic features, we train a ranker of QA pairs with negative sampling. Preliminary results show that this method improved the naturalness of responses. The improved method first uses the Lucene text search engine,[2] which was also used for the original method. The method then calculates the probabilities that the retrieved candidates obtained with Lucene are appropriate responses and uses such probabilities for re-ranking. These probabilities are calculated using a supervised classification model based on bidirectional encoder representations from Transformers (BERT) [3].

3.2 Pre-trained Transformer-Based Encoder-Decoder Model

BlenderBot [17] is a state-of-the-art chatbot based on the Transformer-based encoder-decoder model. To verify the effectiveness of the model in Japanese and the problems that may remain after applying it, a Japanese version of BlenderBot model was developed by Sugiyama et al. [20, 21], who reported that it was able to chat with users fluently. With the trained model, we fine-tuned the Japanese version of the BlenderBot model to build a character-like chatbot. The outline of the model construction conducted by Sugiyama et al. [20, 21] is described below.

Instead of Reddit posts, which were used as the pre-training data in the original BlenderBot [17], dialogues consisting of Twitter replies in Japanese were used [20]. Sugiyama et al. retrieved the tweet threads as dialogues from January 2016 to March 2018, filtered certain tweets (those with similar surface forms, those containing URLs or parentheses, those by bots and retweets, and those with a small percentage of Japanese), and removed account names and emojis from the remaining tweets after filtering. Input-output pairs were then created by splitting the threads so that each tweet became the output. Inputs were allowed to include up to four utterances. As a result, the size of the pre-training data was 2.1B (512 GB).

In fine-tuning BlenderBot, Roller et al. [17] used two types of dialogue: one for learning the skills of personality, knowledge, and empathy and one for blending these skills. Sugiyama et al. [20] created Japanese versions of these two types through crowdsourcing. For the first type, regarding personality, they translated the profile sentences of PERSONA-CHAT [23] into Japanese and manually created 200 dialogues based on the process of creating PERSONA-CHAT. Regarding knowledge, they created 1,886 dialogues from Wizard-of-Wikipedia [4] by translating and editing them. Regarding empathy, they created 400 dialogues, including 200 that were translated from Empathetic Dialogue [16] and 200 that were manually created. For the second type of dialogue, three Japanese corpora were used for learning the blended skills, instead of the Blended Skill Talk corpus [19]. The first is a corpus of listening-oriented dialogues in which the goal is to actively listen and show satisfaction to the speaker (1,260 dialogues) [13]. The second is a corpus of chat-oriented dialogues about a variety of topics among speakers who have never met before (3,600 dialogues) [8]. The third is a corpus of dialogues in which each speaker chats with dozens of others about their hobbies (3,483 dialogues).

[2] https://lucene.apache.org/.

A generative Transformer-based encoder-decoder model [17, 20] with 1.6B parameters (2-layer encoder, 24-layer decoder, and 1920 dimensions of each embedding) was used. This is the maximum model size that can fit in the memory of the GPUs used for training in their resources and is smaller than the original model (2.7B). The decoding method was a standard sample-and-rank method [1] to prevent dull responses. An utterance filter was applied to the generated utterances for re-ranking to prevent the selection of utterances similar to those in the context. If the similarity of token sets between a candidate and utterances in the context exceeded a threshold, a penalty score was added to the candidate.

The pre-trained model was trained on a corpus of Twitter dialogues by connecting each utterance with a separator and setting the number of tokens processed per step to 4.8M. Adafactor [18] was used as the optimizer. The training time was about 28 h, up to 30,000 steps on 400 GPUS of V100 (16 GB), until the validation loss became approximately flat. For fine-tuning, the mixture of the corpora for learning the above skills was used. The input to the encoder was embedded with information in accordance with the following template.

```
DialogueName:[SEP]SpeakerID[SEP][SPK1]Utterance1[SEP][SPK2]Utter
ance2[SEP][SPK1]Utterance3[SEP][SPK2]Utterance4[SEP]TurnNumber
```

DialogueName, SpeakerID, and TurnNumber correspond to the kind of corpus, the unique speaker ID in the corpus, and the number of utterances from the beginning of the dialogue, respectively. Adafactor was also used to train up to 100 steps to minimize the perplexity of the validation set on 128 GPUs of V100. Finally, using this model, we created character-like chatbots by conducting additional fine-tuning with the collected role-play-based QA corpus and character-dialogue corpus. The fine-tuning was done in the same manner as described above regarding the template of the input to the encoder and the process of training.

4 Experiments

Using the model described in the previous section with the collected role-play-based QA corpus and character-dialogue corpus, we created chatbots and performed automatic and manual evaluations to determine whether it was possible to make character-like dialogues. We conducted three manual evaluations: two utterance-level evaluations, in which the input is a question in the role-play-based QA corpus (**Question-In**) or a context in the character-dialogue corpus (**Context-In**), and one manual evaluation on the dialogue-level, in which a user interacts with a chatbot for a certain number of turns and then evaluates the entire dialogue.

4.1 Data and Model Preparation

The role-play-based QA corpus was divided into training, validation, and test sets (8:1:1). The splits were made so that unique questions would not overlap. The character-dialogue corpus was divided in the same manner. For the manual utterance-level evaluation, we used questions from the role-play-based QA corpus or contexts created by excluding the last character's utterance for testing. We used 50 questions and 50 contexts as test inputs.

We prepared four models: one retrieval-based and three generation-based.

(a) Ret-BERT-NS: The retrieval-based model using fine-tuned BERT[3] with the role-play-based QA corpus and negative samples (five per reference) created from the corpus. The model selects answers from the QA pairs contained in the training or validation set.

(b) Gen-QA-Small: A generation-based model in which the pre-trained Transformer-based encoder-decoder model is fine-tuned with a reduced role-play-based QA corpus (10 K QA pairs). We prepared this variation so as to investigate the effect of the size of role-play-based QA corpus on performance.

(c) Gen-QA: A generation-based model in which the pre-trained Transformer based encoder-decoder model is fine-tuned with the full role-play-based QA corpus (44 K QA pairs).

(d) Gen-QA+Dial: A variant of the generation-based model in terms of the training data, in which the full role-play-based QA corpus and character-dialogue corpus are used. This model was compared with Gen-QA to investigate the limitations of only using the role-play based QA corpus.

4.2 Automatic Evaluations

Table 3 shows the automatic evaluation results of system utterances for input questions (left side) and input contexts (right side). The scores are for three metrics: BLEU-1, distinct-1 [11], and length (tokens) of responses. The BLEU scores show that the generation-based models could generate more similar responses to the references than the retrieval-based model. Although the lengths were similar for both types of model, distinct scores were much higher for the retrieval-based model, indicating the limitation of generation-based models in terms of diversity.

[3] https://github.com/yoheikikuta/bert-japanese.

Table 3 Automatic evaluation results of system utterances for input questions (Question-In) and input contexts (Context-In). BLEU, Distinct, and Length correspond to BLEU-1, Distinct-1, and average number of tokens, respectively

Model	Question-In			Context-In		
	BLEU	Distinct	Length	BLEU	Distinct	Length
(a) Ret-BERT-NS	0.120	**0.646**	17.1	0.077	**0.635**	**15.4**
(b) Gen-QA-Small	0.197	0.496	17.2	0.132	0.482	14.1
(c) Gen-QA	0.197	0.487	**18.6**	0.135	0.476	14.4
(d) Gen-QA+Dial	**0.204**	0.512	17.0	**0.166**	0.480	12.2

4.3 Manual Evaluation (Utterance-Level)

In this evaluation, each output utterance generated from the models was manually evaluated with the input question or context. The following three evaluation metrics were used.

- **Naturalness**: Whether the response was natural in general (irrespective of the character).
- **Characterness**: Whether the response to the input question was appropriate for the target character (Amadeus Kurisu in this experiment).
- **Informativeness**: Whether the response provided new information related to the input question.

The evaluation was conducted using a five-point Likert scale (1: strongly disagree; 5: strongly agree). For the evaluation, ten crowdworkers were recruited through the same platform as for collecting the character-dialogue corpus. We required the applicants to have sufficient knowledge of the target character. After the workers were instructed to evaluate each metric independently, they evaluated randomly shuffled output utterances generated from the compared models for each input question or context.

Table 4 shows the results of manually evaluating system utterances for input questions (left side) and input contexts (right side). For Question-In, the naturalness scores of the generation-based models were significantly better than those of Ret-BERT-NS. We can assume that these scores saturated even in Gen-QA-Small because of the similar scores among the generation-based models. The fact that the generation-based models could create responses with the same characterness and informativeness as the manually written retrieval-based responses demonstrates the effectiveness of fine-tuning the pre-trained model with the collected character data. For Context-In, the scores of not only naturalness but also characterness with the generation models were significantly better than those with Ret-BERT-NS. Unlike Question-In, the characterness score of Ret-BERT-NS was low because it does not take into account entire contexts. Gen-QA was significantly better than the others. These results lead us to conclude that it is possible to generate character-like context-aware responses even without character dialogue data, as long as a large number of QA pairs is available.

Table 4 Results of manual evaluation (utterance-level). Input was question (Question-In) or context (Context-In). Nat, Char, and Info correspond to naturalness, characterness, and informativeness, respectively. Subscripts indicate that a score is significantly better than those of corresponding models

Model	Question-In			Context-In		
	Nat.	Char.	Info.	Nat.	Char.	Info.
(a) Ret-BERT-NS	3.05	3.33	3.30	2.53	2.96	2.89
(b) Gen-QA-Small	**3.59**$_a$	3.42	**3.42**$_a$	3.66$_a$	3.55	3.09
(c) Gen-QA	3.57$_a$	3.48	3.37	**3.81**$_a$	**3.79**$_{abd}$	**3.17**$_{ad}$
(d) Gen-QA+Dial	3.52$_a$	**3.50**	3.32	3.62$_a$	3.54	2.94

4.4 Manual Evaluation (Dialogue-Level)

In addition to evaluating the output utterances, workers interacted with the chatbots and evaluated the entire dialogue to determine how much the dialogue felt like the real character. To increase the reliability of the results, 20 workers participated in the evaluation, i.e., the original ten workers who participated in the utterance-level evaluation and ten newly recruited workers from the same crowdsourcing platform. They interacted with each of the four chatbots three times in random order. Inspired by the rules of a competition for evaluating chat-oriented dialogue systems [7], the length of each dialogue was 15 turns, where a pair of a system utterance and user utterance is considered one turn. At the end of each dialogue, the entire dialogue was evaluated on a five-point Likert scale in terms of naturalness, characterness, and informativeness. Telegram[4] was used as the tool for workers to interact with the chatbots.

Table 5 shows the results of the human evaluation at the dialogue-level. Although Gen-QA+Dial had the best scores on average, Gen-QA shows that character-like context-aware dialogues were possible without the dialogue data. As with the utterance-level evaluations, the naturalness of the generation-based models was significantly better than that of Ret-BERT-NS, and the difference was more significant in the dialogue-level evaluations. On the other hand, there was no difference in characterness. This may be because Ret-BERT-NS returned character-like utterances that were written manually and were sometimes natural in context. Informativeness also tended to be higher in the generation-based models, but no significant differences were found except for Gen-QA-Small.

[4] https://telegram.org/.

Table 5 Results of manual evaluation (dialogue-level). Workers evaluated entire dialogue after 15 turns. Nat, Char, and Info correspond to naturalness, characterness, and informativeness, respectively. Subscripts indicate that a score is significantly better than those of corresponding models

Model	Nat.	Char.	Info.
(a) Ret-BERT-NS	2.50	3.87	3.08
(b) Gen-QA-Small	3.30_a	3.50	$\mathbf{3.75}_a$
(c) Gen-QA	3.55_a	**3.95**	3.47
(d) Gen-QA+Dial	$\mathbf{3.87}_{ab}$	3.90	3.58

5 Analysis

We analysed the evaluation results reported in the previous section in greater detail to qualitatively investigate the reasons for improvement.

For the results of the utterance-level evaluation, system utterances were assumed to be inappropriate if any of the three scores were less than three on average. Therefore, we labelled such inappropriate utterances with the types of error in chat-oriented dialogue systems [5]. The errors are categorized into four main categories (utterance-level: errors in utterance, response-level: ignorance of user utterances, context-level: irrelevant topics, and society-level: violation of a social norm), and these are further divided into 17 sub-categories. For inappropriate utterances, labelling was done by an experienced in-house annotator. Results showed that the generation-based models had fewer response-level and context-level errors than Ret-BERT-NS. However, in Context-In, context-level errors were still common (40% of inputs), particularly in the "Repetition" (utterance repeats previous utterances) and "Lack of information" (utterance omits important information) sub-categories. Thus, while using the pre-trained encoder-decoder models suppressed response-level and some context-level errors, repetition or lack of information were still problems with the generation-based models.

For the dialogue-level evaluation, we checked the length, distinct, and dialogue act of system and user utterances and found that the system utterances of Gen-QA+Dial were shorter (24.1 tokens) than those of the others (about 31–32 tokens). The tendency of higher naturalness in Gen-QA+Dial might be because it returns utterances that are closer to the length of the user utterances (about 19 tokens) than the other models, which increases the continuity of dialogue. As for dialogue acts, we used a label set proposed for chat-oriented dialogue systems [13] and the same SVM-based classifier for labelling the dialogue acts used in previous research [8], and found that in Gen-QA+Dial, questions from the chatbot (11.1%) and self-disclosure from the user (39.2%) were more frequent than in Gen-QA (8.9 and 34.6%). Since the collected QA data do not include questions from the system, the additional dialogue data might have been effective to increase the naturalness of the dialogue.

Table 6 Sample of successful dialogue with proposed chatbot Gen-QA+Dial (S) and user (U)

ID	Utterance
S_1	[Fixed utterance] Hello. Nice to meet you
U_1	Hello, nice to meet you too
S_2	You're looking good. Did something nice happen?
U_2	I cleaned up my room and I feel good
S_3	A clean room makes you feel good, doesn't it?
U_3	Yes, it does. I don't like to clean, though
S_4	You don't? But it feels good to clean
U_4	Well, it feels good to be clean
S_5	That's true. It's not good to accumulate things
U_5	That's right. Do you throw everything away immediately?
S_6	I don't throw anything away. I keep things that hold memories
U_6	Is that so! That's a little surprising
S_7	What do you mean, "surprising"? What sort of person do you think I am?
U_7	I thought you wouldn't cherish past memories too much because your personality is so outgoing
S_8	That's not true. I'm surprisingly meticulous
U_8	That's what I think
S_9	Thank you. I'm happy to hear you say that
U_9	You seem meticulous
S_{10}	Thank you. I'll take that as a compliment
U_{10}	Of course it's a compliment
S_{11}	Thank you. I'll never forget this
U_{11}	Please continue being good friends with me
S_{12}	Of course. I'm looking forward to hanging out with you a lot
U_{12}	I'd like to go someplace with you in the summer
S_{13}	That would be great. I'd be happy if you could take me around different places
U_{13}	Let's go to the beach
S_{14}	That's a good idea. I like the ocean, too
U_{14}	Let's go into the sea
S_{15}	I don't want to. No one will be happy to see me in one. [Fixed utterance] Oh, I'm sorry. I think it's time to go. Thank you very much for today
U_{15}	Thank you very much

Table 6 shows a successful dialogue of the Gen-QA+Dial. We can confirm that the system was able to generate contextualized utterances with consistent characterness. On the other hand, some of the utterances, such as S_3, were repetitive and contained little information.

6 Conclusion

In this study, we investigated how character-like dialogue can be achieved by fine-tuning a pre-trained Transformer encoder-decoder model using question-answer pairs collected for a specific character. As training data, we collected QA data of a famous animation character and used it along with an additional dialogue corpus for the character. The results showed that, with about 10 K QA pairs, it is possible to create significantly more natural chatbots than with the conventional retrieval-based method. In addition, with 44 K QA pairs, it is possible to achieve high naturalness and characterness scores (close to 4 on a 5-point scale) without the dialogue data.

For future work, we intend to improve the generation-based method in various ways. Since we used at least 10 K QA pairs in this experiment, we want to develop a method that behaves appropriately even with less data. For this purpose, a method that refers to external knowledge about the character (Wikipedia text, script, knowledge graph, etc.) may be useful. In addition, we would like to enable natural dialogue by using only QA data without dialogue data on the basis of the analysis in this study. We also want to improve the system performance by having users continuously add or modify QA data, or by relearning based on user evaluations, in order to create a chatbot learns and behaves like the target character.

References

1. Adiwardana D, Luong MT, So DR, Hall J, Fiedel N, Thoppilan R, Yang Z, Kulshreshtha A, Nemade G, Lu Y, Le QV (2020) Towards a human-like open-domain chatbot. arXiv:2001.09977
2. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. In: Proceedings of NeurIPS, pp 1877–1901
3. Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL, pp 4171–4186
4. Dinan E, Roller S, Shuster K, Fan A, Auli M, Weston J (2019) Wizard of wikipedia: knowledge-powered conversational agents. In: Proceedings of ICLR, pp 1–18
5. Higashinaka R, Araki M, Tsukahara H, Mizukami M (2021) Integrated taxonomy of errors in chat-oriented dialogue systems. In: Proceeding of SIGDIAL, pp 89–98
6. Higashinaka R, Dohsaka K, Isozaki H (2013) Using role play for collecting question-answer pairs for dialogue agents. In: Proceedings of INTERSPEECH, pp 1097–1100
7. Higashinaka R, Funakoshi K, Inaba M, Tsunomori Y, Takahashi T, Akama R (2021) Dialogue system live competition: identifying problems with dialogue systems through live event. In: Proceedings of IWSDS, pp 185–199
8. Higashinaka R, Imamura K, Meguro T, Miyazaki C, Kobayashi N, Sugiyama H, Hirano T, Makino T, Matsuo Y (2014) Towards an open domain conversational system fully based on natural language processing. In: Proceedings of COLING, pp 928–939
9. Higashinaka R, Mizukami M, Kawabata H, Yamaguchi E, Adachi N, Tomita J (2018) Role play-based question-answering by real users for building chatbots with consistent personalities. In: Proceedings of SIGDIAL, pp 264–272

10. Kodama T, Higashinaka R, Mitsuda K, Masumura R, Aono Y, Nakamura R, Adachi N, Kawabata H (2020) Generating responses that reflect meta information in user-generated question answer pairs. In: Proceedings of LREC, pp 5433–5441
11. Li J, Galley M, Brockett C, Gao J, Dolan B (2016) A diversity-promoting objective function for neural conversation models. In: Proceedings of NAACL-HLT, pp 110–119
12. Li J, Galley M, Brockett C, Spithourakis G, Gao J, Dolan B (2016) A persona-based neural conversation model. In: Proceedings of ACL, pp 994–1003
13. Meguro T, Higashinaka R, Minami Y, Dohsaka K (2010) Controlling listening-oriented dialogue using partially observable Markov decision processes. In: Proceedings of COLING, pp 761–769
14. Qian Q, Huang M, Zhao H, Xu J, Zhu X (2018) Assigning personality/profile to a chatting machine for coherent conversation generation. In: Proceedings of IJCAI, pp 4279–4285
15. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2019) Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv:1910.10683
16. Rashkin H, Smith EM, Li M, Boureau YL (2019) Towards empathetic open-domain conversation models: a new benchmark and dataset. In: Proceedings of ACL, pp 5370–5381
17. Roller S, Dinan E, Goyal N, Ju D, Williamson M, Liu Y, Xu J, Ott M, Smith EM, Boureau YL, Weston J (2021) Recipes for building an open-domain chatbot. In: Proceedings of EACL, pp 300–325
18. Shazeer N, Stern M (2018) Adafactor: adaptive learning rates with sublinear memory cost. In: Proceedings of ICML, pp 4596–4604
19. Smith EM, Williamson M, Shuster K, Weston J, Boureau YL (2020) Can you put it all together: evaluating conversational agents' ability to blend skills. In: Proceedings of ACL, pp 2021–2030
20. Sugiyama H, Narimatsu H, Mizukami M, Arimoto T, Chiba Y, Meguro T, Nakajima H (2020) Development of conversational system talking about hobby using transformer-based encoder-decoder model. In: Proceedings of special interest group on spoken language understanding and dialogue processing (in Japanese), pp 104–109
21. Sugiyama H, Narimatsu H, Mizukami M, Arimoto T, Chiba Y, Meguro T, Nakajima H (2021) Analysis of subjective evaluation for fine-tuning methods of transformer encoder-decoder based conversational systems. In: Proceedings of The 35th annual conference of the Japanese society for artificial intelligence (in Japanese), pp 4E1–OS–11a–03
22. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Proceedings of NeurIPS, pp 5998–6008
23. Zhang S, Dinan E, Urbanek J, Szlam A, Kiela D, Weston J (2018) Personalizing dialogue agents: i have a dog, do you have pets too? In: Proceedings of ACL, pp 2204–2213
24. Zhou L, Gao J, Li D, Shum HY (2020) The design and implementation of XiaoIce, an empathetic social chatbot. Comput Linguist 46(1):53–93

Investigating the Impact of Pre-trained Language Models on Dialog Evaluation

Chen Zhang, Luis Fernando D'Haro, Yiming Chen, Thomas Friedrichs, and Haizhou Li

Abstract Recently, there is a surge of interest in applying pre-trained language models (Pr-LM) in automatic open-domain dialog evaluation. Pr-LMs offer a promising direction for addressing the multi-domain evaluation challenge. Yet, the impact of different Pr-LMs on the performance of automatic metrics is not well-understood. This paper examines eight different Pr-LMs and studies their impact on three typical automatic dialog evaluation metrics across three different dialog evaluation benchmarks. Specifically, we analyze how the choice of Pr-LMs affects the performance of automatic metrics. Extensive correlation analyses on each of the metrics are performed to assess the effects of different Pr-LMs along various axes, including pre-training objectives, dialog evaluation criteria, model size, and cross-dataset robustness. This study serves as the first comprehensive assessment of the effects of different Pr-LMs on automatic dialog evaluation.

C. Zhang (✉) · Y. Chen · H. Li
National University of Singapore (NUS), Singapore, Singapore
email: haizhouli@cuhk.edu.cn
e-mail: chen_zhang@u.nus.edu

Y. Chen
e-mail: yiming.chen@u.nus.edu

H. Li
e-mail: haizhou.li@u.nus.edu

L. F. D'Haro
Universidad Politécnica de Madrid (UPM), Madrid, Spain
e-mail: luisfernando.dharo@upm.es

T. Friedrichs
Robert Bosch (SEA) Pte Ltd, Singapore, Singapore
e-mail: Thomas.Friedrichs@sg.bosch.com

H. Li
Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen, China
email: haizhouli@cuhk.edu.cn

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*, Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_21

1 Introduction

Evaluation is crucial for monitoring the research progress of dialog systems [1, 2]. Even though human evaluation is the most accurate way to assess the performance of a dialog system, the required expenses and efforts restrict its application to large-scale dialog evaluation tasks. Therefore, automatic dialog evaluation (ADE) serves as an efficient alternative to human evaluation. An ideal ADE metric is expected to evaluate dialog systems of different domains efficiently and effectively. Realizing such a metric is a challenging task [1]. One promising trend is to leverage large-scale pre-trained language models (Pr-LMs) [3, 14, 16, 19], which have gained significant momentum in a wide range of NLP tasks. Several recent studies [4–8] have also demonstrated their usefulness in ADE.

However, different Pr-LMs have different pre-training schemes and they are not directly optimized for dialog evaluation. A dialog metric relies on dialog-specific features for determining the quality of dialog responses and whether the pre-training process of a Pr-LM captures such features has not been extensively studied. In addition, the choice of Pr-LM will significantly affect the ADE metrics' performance and generalizability across various evaluation tasks and evaluation dimensions (e.g., coherence, interestingness, and naturalness). Currently, there is no comprehensive analysis to guide the choice of Pr-LM for ADE.

To this end, a systematic study on how different Pr-LMs affect the evaluation effectiveness of various ADE metrics is highly sought after. In this paper, we survey eight existing state-of-the-art Pr-LM variants and analyze their impact on the performance of three typical ADE metrics, the embedding-based similarity measure, the normalized sentence-level log probability, and the context-response coherence metric. These three metrics are commonly used across multiple domains and can directly relate to three basic dimensions to evaluate a dialog system: adequacy (semantic similarity with existing references), naturalness, and context coherence. Through extensive correlation analysis on three dialog evaluation benchmarks, we try to link the properties associated with the Pr-LMs, such as model size, source of pre-training data, and learning objective, with the metrics' final performance. Our work serves as a first step into understanding the role of different Pr-LMs in automatic open-domain dialog evaluation. In addition, it will help guide future research in making more informed choices when applying pre-trained language models in various dialog evaluation tasks. Note that we are not proposing a new evaluation metric to advance the state of the art, but rather studying the impact of different Pr-LMs on existing ADE metrics.

The paper is organized as follows: Sect. 2 discusses the eight different pre-trained language models. Section 3 talks about the three ADE metrics. Section 4 is the experiment results and corresponding analysis. Finally, Sect. 5 concludes the paper and lays out the future work.

2 Pre-trained Language Models

In this section, we discuss eight state-of-the-art variants of Pr-LMs and group them based on their pre-training objectives. Specifically, there are five groups. The first four groups are token-level representation models: (a) masked language modeling (Sect. 2.1), (b) replaced token detection (Sect. 2.2), (c) causal language modeling (Sect. 2.3), and (d) permutation language modeling (Sect. 2.4). The final group consists of sentence-level representation models, which are learned with specific sentence-level objectives (Sect. 2.5). All of the Pr-LMs are based on the transformer architecture [9], which has greatly changed the NLP landscape in recent years. Contextualized embeddings derived from the Pr-LMs are beneficial to the evaluation metrics since they have been demonstrated to carry rich syntactic structure information [10] and semantic meanings [11] of sentences. Recent studies have also shown that rich world knowledge is encoded in the parameters of the Pr-LMs [12]. Such information helps the metric better determine the linguistic quality of the generated dialog responses.

2.1 Masked Language Modeling (MLM)

MLM is a self-supervised pre-training task proposed in [3] whereby the language model randomly masks out some tokens from the input sequence and the learning objective is to predict the original vocabulary ids of the masked tokens. Formally, given a input text sequence of n tokens, $T = \{t_1, \ldots, t_n\}$, we corrupt T into $\tilde{T}$ by randomly masking a portion of tokens. A language model parameterized by θ is trained to reconstruct T by predicting the set of masked tokens $\tilde{t}$ conditioned on $\tilde{T}$:

$$\max_{\theta} \log p_{\theta}(T|\tilde{T}) = \max_{\theta} \sum_{i \in C} \log p_{\theta}(\tilde{t_i} = t_i|\tilde{T}) \tag{1}$$

where C presents the set of indices of masked tokens. Typical examples in this category include BERT [3], RoBERTa [14], DeBERTa [15], and ELECTRA [16].

BERT [3] BERT is currently the most fundamental Pr-LM and a must-have baseline in a wide range of NLP tasks. The backbone of BERT is a stack of transformer encoders, which is pre-trained with two learning objectives in a multi-task setting. The first objective is the aforementioned mask language modeling. The second objective is next sentence prediction (NSP). BERT is useful in dialog evaluation in the sense that it reconstructs the original input text leveraging bidirectional context. This leads to an accurate estimation of the sentence-level probability rendering it useful for evaluating the fluency of the generated dialog responses. Several previous studies have questioned the necessity of the NSP objective [13, 14]. Hence, BERT's benefits to context coherence metric are questionable.

RoBERTa [14] is an optimized version of BERT. The backbone architecture and hyperparameters of RoBERTa are almost the same as BERT. Unlike BERT, RoBERTa is solely optimized with the MLM objective. It has been shown to demonstrate better performance across a wide range of natural language understanding tasks compared to BERT. Since RoBERTa is only pre-trained with the MLM objective, we hypothesize that it may perform worse when evaluating sentence-level dialog properties, such as context coherence. However, compared to BERT, it may provide a more accurate estimation of the sentence-level probability since RoBERTa is a highly optimized token-level language model with the MLM objective.

DeBERTa [15] DeBERTa is a relatively new member in the BERT family[1] and it improves upon BERT and RoBERTa by disentangling the attention mechanism whereby two separate vectors are used to represent each input token and they encode the token's semantic meaning and position in the text sequence accordingly. Then, disentangled matrices are adopted for computing the attention weights among all the tokens in the input sequence on their contents and relative positions, respectively. Furthermore, to account for the syntactical nuances in the input text, DeBERTa incorporates absolute position token embeddings right before the prediction layer. The model predicts the masked tokens based on aggregated information about the token contents and positions. Since DeBERTa only modifies the model architecture and there is no change in the pre-training objective. Hence, we hypothesize that applying DeBERTa for dialog evaluation will obtain similar performance w.r.t. to the use of RoBERTa or BERT.

2.2 Replaced Token Detection (RTD)

The RTD objective is introduced to pre-train the **ELECTRA** framework [16], which consists of an MLM-based generator and a discriminator. The backbone of ELECTRA is BERT with a binary classification layer on top. The difference between RTD and MLM is that MLM is a fill-in-the-blank task while RTD relies on the mask-and-infill mechanism. Specifically, some tokens in the input text sequence are replaced with alternatives sampled from a small generator. Then, a discriminator is trained to predict the identity of each of the tokens in the input sequence (whether it is the original token or a sampled one). Compared to MLM, the RTD pre-training objective is more useful for dialog evaluation because of the following reasons: (1) There is no mismatch between pre-training and testing as the model doesn't have to deal with the artificial mask tokens. (2) RTD directly optimizes the discriminator to distinguish tokens from the original distribution against adversarial samples from a generator conditioned on the bidirectional context. A major goal of dialog evaluation is to rank dialog responses sampled from different generators of varying degrees of quality conditioned on the dialog context. Hence, RTD better aligns with the goal of dia-

[1] Introduced in the Ninth International Conference on Learning Representations (ICLR 2021).

log evaluation compared to MLM. We hypothesize that applying ELECTRA as the backbone will achieve good performance for ADE.

2.3 Causal Language Modeling (CLM)

CLM objective is the traditional unidirectional autoregressive way of pre-training language models whereby the model tries to predict the next token conditioned on all the previous tokens. Formally, given a input text sequence of n tokens, $T = \{t_1, \ldots, t_n\}$, the language model parameterized by θ performs pre-training by maximizing the likelihood:

$$\max_{\theta} \log p_{\theta}(T) = \sum_{i=1}^{n} \log p_{\theta}(t_i | t_{<i}) \tag{2}$$

In our study, we examine one Pr-LM with the CLM objective, **DialoGPT** [17]. The backbone architecture of DialoGPT is the same as GPT-2 [18], which consists of a stack of masked multi-head self-attention layers. Unlike GPT-2, which is pre-trained on a massive amount of web-text data, DialoGPT is pre-trained on large-scale dialogs extracted from Reddit. The input to the DialoGPT is the context-response pairs and the model is pre-trained to maximize the probability of the response conditioned on the corresponding context. Out of all the Pr-LMs examined in this paper, DialoGPT is the only dialog-specific Pr-LM. Given its pre-training objective, we hypothesize that DialoGPT is useful for perplexity-based evaluation metrics.

2.4 Permutation Language Modeling (PLM)

In [19], the authors propose the **XLNET** framework and introduce the PLM pre-training objective, which tries to combine the best of both CLM and MLM pre-training schemes: MLM-based Pr-LMs can capture bidirectional contextual information whereas CLM-based Pr-LMs don't assume the independence of tokens within the sequence and, hence, can model the high-order and long-range dependency in natural language. With the PLM objective, the model aims to maximize the expected log likelihood of a sequence w.r.t. all its possible permutations of the factorization order. Although PLM addresses the shortcomings of MLM and CLM, [14] has demonstrated that with the same amount of pre-training data, XLNET is not superior compared to BERT. Hence, we hypothesize that its contribution to dialog evaluation may not be better than Pr-LMs optimized with the MLM objective.

2.5 Sentence-Level Representation Learning

Compared to token-level representation models, sentence-level representation models may be more pertinent to the dialog evaluation tasks for the following reasons: (1) A dialog is essentially a coherent structure consisting of multiple utterances. The dynamics of information exchange among the interlocutors are captured by examining the interaction among utterances instead of a flattened sequence of tokens, which is too fine-grained. (2) Both the embedding-based similarity measure and the context-response coherence measure operate at the sentence level. Extra adaptation may be required for Pr-LMs pre-trained with the token-level objectives.

Sentence-BERT [20] The backbone framework of Sentence-BERT (SBERT) is the BERT model. A siamese network is constructed to encode pairs of sentences. The model is then fine-tuned with the combination of SNLI [21] and MNLI [22] datasets. The standard cross-entropy classification objective is adopted to optimize the model. Two other supervised objective functions that operate at the sentence level have also been experimented with. One is the mean-squared loss function and the other is the triplet loss function. Since SBERT is a sentence-level representation model, we hypothesize its performance will be better than BERT or RoBERTa for the embedding-based similarity and the context-response coherence metrics.

SimCSE [23] SimCSE is the current state of the art in both supervised and unsupervised sentence representation learning. The unsupervised SimCSE only leverages dropout for data augmentation whereby the same sentence is passed into the encoder twice. With an independently sampled dropout mask, a positive pair can be obtained. Other sentences in the same batch serve as negative instances. A contrastive loss is applied to pull the positive pairs closer and the negative pairs apart in the vector space. The supervised SimCSE uses the same contrastive loss on the entailment and contradiction pairs from the NLI datasets for sentence representation learning. Since SimCSE demonstrates state-of-the-art performance in a wide array of semantic textual similarity and transfer tasks, we hypothesize that it will also greatly benefit the dialog evaluation tasks and obtain better performance compared to SBERT.

3 Automatic Dialog Evaluation Metrics

In this section, we introduce three simple, but widely adopted automatic dialog evaluation metrics of which the choice of Pr-LM can lead to a significant impact on performance.[2]

Embedding-based Similarity The embedding-based similarity metric (ESM) is a reference-based measure, which evaluates a generated dialog response based on its similarity w.r.t. a reference sentence written by the human annotators. Usually, cosine

[2] Implementation at https://github.com/e0397123/dstc10_metric_track.

similarity between the response embedding, $\mathbf{h}$, and the reference embedding, $\mathbf{r}$, is used as the metric score:

$$sim(\mathbf{h}, \mathbf{r}) = \frac{\mathbf{h}^T \mathbf{r}}{||\mathbf{h}|| \cdot ||\mathbf{r}||} \tag{3}$$

Compared to lexical-overlap metrics, such as BLEU [24] and ROUGE [25], ESM is more flexible by allowing variations in the lexical form and focusing on the sentence-level semantics. The choice of embeddings has a significant impact on the performance of ESM. In our study, we investigate which Pr-LM provides useful vector representations to ESM. Several prior studies [4, 26, 27] have proposed improvement versions of ESM leveraging Pr-LMs, but they didn't conduct a comprehensive analysis of the effects of different Pr-LM variants.

Normalized Sentence-level Log Probability Language models estimate the true distribution of natural language and assign probabilities to sequences of words. The sentence-level log probability (SLP) estimated by a language model can indicate the naturalness of a generated dialog response. Unlike ESM, SLP is a type of reference-free metrics, which doesn't depend on human-written references to determine the quality of generated dialog responses. In dialog evaluation, we want to examine how the naturalness of the generated dialog responses is affected by the corresponding context. Hence, we formulate SLP by taking into account the preceding contexts, p, of the generated response, h. For language models pre-trained with the CLM or PLM objectives, the normalized log probability is estimated as follows:

$$\frac{1}{M} \log p_\theta(T) = \frac{1}{M} \sum_{i=1}^{M} \log p_\theta(t_i | t_{<i}) \tag{4}$$

where T denotes the concatenation of p and h. M is the total number of tokens in T. For language models pre-trained with the MLM objective, the normalized log probability is computed with

$$\frac{1}{M} \log p_\theta(T) = \frac{1}{M} \sum_{i=1}^{M} \log p_\theta(t_i | \tilde{T}) \tag{5}$$

where $\tilde{T}$ is the corrupted T with t_i being masked.

Context-response Coherence Discourse coherence is a broad area of research, and in dialog evaluation, we try to assess coherence at different granularity. One is the coherence of the entire dialog flow [7] and the other is the local coherence at the turn level, i.e., context-response coherence [28]. In our study, we focus on the local coherence assessment and adopt a simple metric (CoSim) to evaluate the coherence between the context, p, and the response, h:

$$sim(\mathbf{p}, \mathbf{h}) = \frac{\mathbf{p}^T \mathbf{h}}{||\mathbf{p}|| \cdot ||\mathbf{h}||} \tag{6}$$

Unlike existing state-of-the-art model-based metrics, of which the evaluation capability may be jointly influenced by several different factors, such as learning strategy, training data, and model architecture, CoSim's performance heavily relies on the choice of sentence embeddings, and hence, it helps us straightforwardly examine the effects of Pr-LMs without the need to decouple impact due to other factors.

4 Experiment and Analysis

This section demonstrates our key findings and is organized as follows: Sect. 4.1 briefly describes the three dialog evaluation benchmarks we use. In Sect. 4.2, we conduct preliminary analysis on the eight Pr-LMs' performance along axes, including model size and cross-dataset robustness to select the top-ranked Pr-LMs for further analysis in the subsequent sections. Section 4.3 includes the main results of Pr-LMs based on average turn-level Spearman rank correlations over all three evaluation benchmarks for each automatic metric. Section 4.4 zooms into the USR-TopicalChat benchmark and analyzes the performance of the top-ranked Pr-LMs along each dialog evaluation dimension for each automatic metric. **Note that all the Pr-LMs are not fine-tuned with any task-specific datasets in our experiments**.

4.1 Dialog Evaluation Benchmarks

We conduct our experiments on the Dailydialog-Eval [29], USR-PersonaChat [5], and USR-TopicalChat [5] dialog evaluation benchmarks. The detailed statistics[3] are presented in Table 1. We select these three benchmarks for the following reasons:

(1) They can be used for both reference-based and reference-free evaluation due to the presence of human-written references.

(2) Annotations of multiple dialog evaluation dimensions are available. This enables a more fine-grained analysis of how different Pr-LMs affect individual dimensions. In both USR-Topical and USR-PersonaChat, each context-response pair is annotated by three dialog researchers along six evaluation dimensions based on different Likert scales: understandability (0–1), naturalness (1–3), maintaining context (1–3), interestingness (1–3), using knowledge (0–1), and overall quality (1–5). For Dailydialog-Eval, 900 dialog context-response data points are annotated and each data point is annotated by four Amazon Mechanical Turkers. The turkers rate the

[3] In our experiment, we use the original human response as the reference w.r.t. the dialog context. Hence, the number of data points in each dataset is less than the original amount.

Table 1 The statistics of Dailydialog-Eval, USR-TopicalChat, and USR-PersonaChat

Dataset name	No. of Data	Dialog len	Utterance len	No. of Annotations	Domain
Dailydialog-Eval	800	4.9	20.18	128,00	Chit-chat
USR-TopicalChat	300	11.20	23.14	5,400	Knowledge-based
USR-PersonaChat	240	4.72	12.39	4,320	Persona-based

response along four different dimensions on a five-point Likert scale: content (adequacy), grammar, relevance, and overall.

(3) The three benchmarks cover the three most common dialog domains often used in open-domain dialog system training.

4.2 *Initial Analysis*

We conduct initial analysis on the Pr-LMs and those with good performance across all the benchmarks are selected for more fine-grained analysis.

The Effects of Model Size For most Pr-LMs, the model size doesn't have a significant influence on the performance of the metrics. The only exception is RoBERTa-large versus RoBERTa-base for SLP where RoBERTa-large outperforms RoBERTa-base by 3.96 percent in terms of the average turn-level Spearman correlation over all the three evaluation benchmarks. This may be related to the fact that RoBERTa is solely optimized with the MLM objective. With more pre-training data and a larger size, the Pr-LM will provide a more accurate estimation of the sentence-level probability. Hence, in our subsequent analysis, we focus on the large version of the Pr-LMs. The detailed results of all Pr-LM variants can be found at https://bit.ly/2UFjWOH.

Cross-dataset Robustness We further analyze the cross-dataset robustness of each Pr-LM by examining their results for each of the evaluation benchmarks. Table 2 shows the per-dataset turn-level average Spearman correlation scores for all the Pr-LMs.[4] **A large variation in terms of average Spearman correlations can be observed**. The difference between the best-performing Pr-LM (SimCSE) and the worst-performing Pr-LM (XLNET) is 10.48%. This may be because SimCSE is optimized for natural language understanding tasks and, thus, provides a good semantic representation of the sentences while XLNET or DialoGPT is optimized to generate more fluent texts. Our evaluation task benefits from better semantic representations of the dialog utterances.

[4] SLP correlations are not included in the computation as sentence-level Pr-LMs cannot serve as the backbone of SLP.

Table 2 Unweighted average turn-level Spearman correlation scores (%) of Pr-LMs across two evaluation metrics (ESM and CoSim) as well as across evaluation dimensions on each benchmark. The best score for each benchmark is highlighted in bold

Dataset	XLNET	DialoGPT	RoBERTa	BERT	DeBERTa	ELECTRA	SimCSE	SBERT
Dailydialog-Eval	4.21*	11.70	8.06*	13.66	15.34	9.47*	**18.24**	13.60
USR-TopicalChat	17.61	11.11	18.13	22.17	20.74	**37.81**	22.89	26.22
USR-PersonaChat	8.47*	10.43*	8.15*	14.87	14.95	14.20	**20.61**	15.03
Average	10.10*	11.08	11.45	16.90	17.01	20.49	**20.58**	18.28

* Denotes statistically insignificance (p-value > 0.05)

Furthermore, it can be seen that **SimCSE performs the best on Dailydialog-Eval and USR-PersonaChat while ELECTRA performs the best on both USR-TopicalChat**. **SimCSE is the most robust Pr-LM** as its performance is consistently good across all three benchmarks. The consistent performance of SimCSE makes it a good choice for multi-domain dialog evaluation metrics.

In general, **almost all of the Pr-LMs perform the best on USR-TopicalChat**. This is because the pre-training data domain of the Pr-LMs is close to that of USR-TopicalChat. Most of the Pr-LMs are pre-trained with Wikipedia articles and USR-TopicalChat contains dialogs discussing topics and facts from Wikipedia.

On the contrary, XLNET and DialoGPT do not perform as well as Pr-LMs pre-trained with the masked language modeling objective, such as BERT and DeBERTa. This may be because the bidirectional language models provide a more accurate representation of sentence semantics compared to unidirectional language models. A more accurate representation of sentence meanings will greatly benefit the embedding-based metrics.

4.3 Rankings of Pre-trained Language Models

After the initial analysis in Sect. 4.2, we try to assess the impact of different Pr-LMs on the performance of each ADE metric. Table 3 shows the average Spearman correlation scores of different (Pr-LM, metric) combinations. Each entry in the table is computed by taking the unweighted average of correlation scores w.r.t each (Pr-LM, metric) pair on all the three dialog evaluation benchmarks. Based on the experiment results, we can make the following observations:

Sentence level versus Token level The results validate our hypothesis in Sect. 2.5 that **sentence-level representation models generally outperform the token-level representation models for the adequacy and coherence metrics**. For ESM (adequacy), SimCSE is the best and for CoSim (coherence), SimCSE and SBERT are among the top-3 rank models.

Table 3 Unweighted average turn-level correlation scores (%) of Pr-LMs w.r.t. ESM (adequacy metric), SLP (fluency metric), and CoSim (coherence metric), respectively. The best score for each metric is highlighted in bold. The second best is italicized and the third one is underlined

Pr-LM	ESM (Adequacy)	SLP (Fluency)	CoSim (Coherence)	AVG. Corr
BERT	17.80	*15.56*	16.01	16.46
DeBERTa	<u>18.35</u>	8.32*	15.67	14.11
DialoGPT	13.13	10.55*	9.02	10.90*
ELECTRA	*18.45*	6.58*	**22.53**	15.85
RoBERTa	12.38	**18.37**	10.52*	13.75
SBERT	17.57	–	<u>19.00</u>	–
SimCSE	**21.68**	–	*19.48*	–
XLNET	8.57*	<u>10.85*</u>	11.62*	10.35*

* Denotes statistically insignificance (p-value > 0.05)

RTD versus MLM For CoSim (coherence) and ESM (adequacy) metrics, we can observe that ELECTRA is ranked the first and the second, respectively. It outperforms BERT by a significant margin of around 6 percent for CoSim (coherence). This validates our hypothesis in Sect. 2.2 that **RTD equips the model with better discrimination power in determining responses of varying degrees of quality compared to MLM**.

Impact of MLM Generally, Pr-LMs pre-trained with mask language modeling (MLM) objective outperform the causal or permutation language models across all three metrics. Based on the results, it can be seen that **MLM-based Pr-LMs provide a more useful semantic representation compared to CLM-based or PLM-based models**. This may be attributed to MLM-based Pr-LMs' ability to capture bidirectional contextual information. In addition, RoBERTa and BERT are ranked the first and the second for the SLP metric, and this validates our hypothesis in Sect. 2.1 that **a highly optimized MLM-based Pr-LM is capable of providing an accurate estimation of sentence naturalness for dialog evaluation**. Moreover, DeBERTa's performance is not better than that of BERT. This is consistent with our hypothesis that **modifications to model architecture instead of pre-training objectives may not bring performance improvement in dialog evaluation**.

Impact of CLM/PLM Based on the average correlation scores, XLNET is the lowest-ranked model for ESM and CoSim metrics. This corroborates our hypothesis in Sect. 2.4 that the contribution of **PLM-based Pr-LM to dialog evaluation is not better than Pr-LMs optimized with the MLM objective**. Furthermore, it is surprising that DialoGPT performs poorly in terms of these three metrics. A possible reason is that even though DialoGPT is a dialog-specific language model, it is pre-trained with large-scale Reddit data, which is more casual and colloquial in style while the dialogs in the three benchmarks are written by humans to fulfill specific purposes. Hence, the language used may be more formal and the quality of the text is better compared to that of Reddit conversations. **Future work should explore adaptation techniques of Pr-LMs to perform different dialog evaluation tasks**.

Correlation Across Metrics It can be observed that the correlations scores in the ESM (adequacy) and CoSim (coherence) categories are generally higher than those in SLP (fluency). This may be due to the properties of the dialog responses whereby a coherent and adequate response is generally fluent while a fluent response may be off-topic or irrelevant to the context. ESM (adequacy) and CoSim (coherence) are designed to distinguish relevant responses from irrelevant ones. They can detect a fluent, yet off-topic response. However, it is hard for SLP (fluency), which is specifically designed to evaluate the naturalness of the generated responses, to distinguish the good from the bad in such scenarios.

4.4 Fine-Grained Analysis on Evaluation Dimension

Section 4.3 provides a holistic comparison of the Pr-LMs for automatic dialog evaluation. In this section, we analyze the performance of SimCSE, RoBERTa, and ELECTRA at a more fine-grained level on the USR-TopicalChat Benchmark. Table 4 showcases the Spearman correlation results at turn level. It can be seen that **ESM-ELECTRA combination performs the best in three out of all six evaluation dimensions. The average correlation scores of ESM-ELECTRA are even approaching that of state-of-the-art USR metric**, which is a model-based metric specifically optimized for the dialog evaluation task. Remarkably, in the ESM (adequacy) category, ELECTRA outperforms the second-best model, SimCSE, by an absolute 13 percent for the overall category. **ELECTRA's superior performance is due to its RTD pre-training objective since all three models are pre-trained on similar datasets and based on similar model architecture. Future work can consider adapting RTD to the sentence level**.

In addition, all three Pr-LMs perform quite well along the using knowledge dimension along the ESM (adequacy) and CoSim (coherence) categories even though no specific adaptation is performed to incorporate the external knowledge sources associated with the dialogs. This corroborates with findings in prior studies mentioned in Sect. 2 that **world knowledge is implicitly encoded in the parameters of the Pr-LMs. Therefore, applying Pr-LMs for multi-domain automatic dialog evaluation is a viable direction**.

Furthermore, the interestingness dimension assesses whether a dialog response is generic/dull or specific to the context and the relevance dimension determines whether a dialog response is on-topic or off-topic w.r.t. the corresponding context. **Even though RoBERTa, SimCSE, and ELECTRA are not directly pre-trained to determine the interestingness and relevance of a dialog response, they perform well when used as the backbones of CoSim (coherence) and ESM (adequacy)**. CoSim-ELECTRA even outperforms USR (49.54 versus 48.77) along the interesting dimension. The reason may be that for ESM (adequacy), there is the presence

Table 4 Turn-level Spearman correlation scores (%) of each metric on the USR-TopicalChat benchmark along individual dialog evaluation dimensions are reported. For each metric, the results of RoBERTa, SimCSE, and ELECTRA are presented. USR [5] is the state-of-the-art reference-free metric for this benchmark. The highest correlation score along each dimension is highlighted in bold. The highest correlation score along each Pr-LM is underlined. Since SimCSE is a sentence-level representation model, it cannot be used for SLP

USR-TopicalChat	ESM (Adequacy)			SLP (Fluency)		CoSim (Coherence)			SoTA
Dimension	RoBERTa	SimCSE	ELECTRA	RoBERTa	ELECTRA	RoBERTa	SimCSE	ELECTRA	USR
Understandability	12.60	17.94	**34.27**	20.50	7.09*	10.65*	13.78	25.36	32.68
Naturalness	12.97	13.26	**34.52**	22.19	5.27*	11.98	14.91	31.17	32.54
Relevance	20.22	25.43	34.78	26.79	2.47*	20.37	36.81	36.10	**37.69**
Interestingness	29.79	37.69	47.52	29.05	5.18*	21.73	21.99	**49.54**	48.77
Using Knowledge	26.79	31.66	32.52	16.82	5.11*	6.94*	5.88*	37.17	**44.68**
Overall	25.78	32.83	**45.89**	27.89	1.56*	17.76	22.52	44.83	41.92
Average	21.36	26.47	38.42	23.87	4.45*	14.91	19.32	37.36	**39.71**

* Indicates statistically insignificance (p-value $>$ 0.05)

of ground-truth references, which are not dull nor off-topic. CoSim (coherence) is explicitly designed to look into the context. ELECTRA, RoBERTa, and SimCSE are state-of-the-art semantic representation models. When the meaning of the sentences is accurately encoded, the performance of both metrics will be greatly boosted.

When evaluating responses with ESM (adequacy) and CoSim (coherence) along understandability and naturalness, RoBERTa and SimCSE don't perform as well as ELECTRA. Their performance is also worse than when applied to evaluate along other dimensions. However, **ESM-ELECTRA performs exceptionally well along these two dimensions and it even outperforms USR. This showcases that ELECTRA may be a good and robust candidate for future development of embedding-reliant ADE metrics**.

However, when used for SLP (fluency), ELECTRA's performance is far worse compared to when used for ESM (adequacy) and CoSim (coherence). The RTD pre-training objective of ELECTRA is to optimize the discriminator instead of the MLM-based generator. Using ELECTRA's generator as a language model to estimate the sentence-level log probability may not be as accurate as RoBERTa, which has been highly optimized for the MLM objective. This explains ELECTRA's poor performance for the SLP metric.

5 Conclusion and Future Work

In conclusion, this paper provides a comprehensive assessment of the impact of eight different state-of-the-art Pr-LMs on ADE metrics. We try to analyze how different pre-training objectives align with the dialog evaluation task. Through extensive correlation analysis, we find out that sentence-level representation models are more robust for multi-domain evaluation tasks. ELECTRA is good at distinguishing the relevant or specific responses from the off-topic or dull responses. Finally, MLM-based Pr-LMs work better than CLM/PLM-based Pr-LMs in evaluating the fluency aspect of the responses. In the future, we will adapt the token-level RTD objective to sentence level to better align with the dialog evaluation task. Additionally, with the insights from this work, we will try to propose new metrics that are useful for multi-domain dialog evaluation. Lastly, we will further examine the impact of task-specific fine-tuning of different Pr-LMs on automatic dialog evaluation metrics.

Acknowledgements We would like to thank all the reviewers for their constructive comments. This work is supported by Science and Engineering Research Council, Agency of Science, Technology, and Research (A*STAR), Singapore, through the National Robotics Program under Human-Robot Interaction Phase 1 (Grant No. 192 25 00054); Human-Robot Collaborative AI under its AME Programmatic Funding Scheme (Project No. A18A2b0046); Robert Bosch (SEA) Pte. Ltd. under EDB's Industrial Postgraduate Program-II (EDB-IPP), project title: Applied Natural Language Processing; The work leading to these results is also part of the project AMIC-PoC (PDC2021-120846-C42) funded by MCIN/AEI/10.13039/501100011033 and by "the European Union "NextGenerationEU/PRTR".

References

1. Yeh YT, Eskenazi M, Mehri S (2021) A comprehensive assessment of dialog evaluation metrics. arXiv:2106.03706
2. Mehri S, Choi J, D'Haro LF, Deriu J, Eskenazi M, Gasic M, ... Zhang C (2022) Report from the NSF future directions workshop on automatic evaluation of dialog: research directions and challenges. arXiv:2203.10012
3. Devlin J, Chang MW, Lee K, Toutanova K (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, Vol 1 (Long and Short Papers), pp 4171–4186
4. Zhang C, D'Haro LF, Banchs RE, Friedrichs T, Li H (2021) Deep AM-FM: toolkit for automatic dialogue evaluation. In: Conversational dialogue systems for the next decade. Springer, Singapore, pp 53–69
5. Mehri S, Eskenazi M (2020) USR: an unsupervised and reference free evaluation metric for dialog generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 681–707
6. Zhang C, Lee G, D'Haro LF, Li H (2021) D-score: holistic dialogue evaluation without reference. IEEE/ACM Trans Audio Speech Lang Proc
7. Zhang C, Chen Y, D'Haro LF, Zhang Y, Friedrichs T, Lee G, Li H (2021) DynaEval: unifying turn and dialogue level evaluation. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing, pp 5676–5689
8. Zhang C, D'Haro LF, Friedrichs T, Li H (2022) MDD-Eval: self-training on augmented data for multi-domain dialogue evaluation. arXiv:2112.07194
9. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, ... Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
10. Liu NF, Gardner M, Belinkov Y, Peters ME, Smith NA (2019) Linguistic knowledge and transferability of contextual representations. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, Vol 1 (Long and Short Papers), pp 1073–1094
11. Tenney I, Xia P, Chen B, Wang A, Poliak A, McCoy RT, ... Pavlick E (2018) What do you learn from context? Probing for sentence structure in contextualized word representations. In: International conference on learning representations
12. Zhou X, Zhang Y, Cui L, Huang D (2020) Evaluating commonsense in pre-trained language models. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, No 05, pp 9733–9740
13. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2020) ALBERT: a lite BERT for self-supervised learning of language representations
14. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, ... Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. arXiv:1907.11692
15. He P, Liu X, Gao J, Chen W (2020) Deberta: decoding-enhanced bert with disentangled attention. arXiv:2006.03654
16. Clark K, Luong MT, Le QV, Manning CD (2019) ELECTRA: pre-training text encoders as discriminators rather than generators. In: International conference on learning representations
17. Zhang Y, Sun S, Galley M, Chen YC, Brockett C, Gao X, ... Dolan WB (2020) DIALOGPT: large-scale generative pre-training for conversational response generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics: system demonstrations, pp 270–278
18. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI blog 1(8):9
19. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) Xlnet: generalized autoregressive pretraining for language understanding. Adv Neural Inf Proc Syst 32

20. Reimers N, Gurevych I (2019) Sentence-BERT: sentence embeddings using siamese BERT-Networks. 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), pp 3982–3992
21. Bowman S, Angeli G, Potts C, Manning CD (2015) A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 632–642
22. Williams A, Nangia N, Bowman S (2018) A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (Long Papers), pp 1112–1122
23. Gao T, Yao X, Chen D (2021) SimCSE: simple contrastive learning of sentence embeddings. arXiv:2104.08821
24. Papineni K, Roukos S, Ward T, Zhu WJ (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics, pp 311–318
25. Lin CY (2004) Rouge: a package for automatic evaluation of summaries. In: Text summarization branches out, pp 74–81
26. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y (2019) BERTScore: evaluating text generation with BERT. In: International conference on learning representations
27. Sellam T, Das D, Parikh A (2020) BLEURT: learning robust metrics for text generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics. pp 7881–7892
28. Cervone A, Riccardi G (2020) Is this dialogue coherent? Learning from dialogue acts and entities. In: Proceedings of the 21th annual meeting of the special interest group on discourse and dialogue, pp 162–174
29. Zhao T, Lala D, Kawahara T (2020) Designing precise and robust dialogue response evaluators. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 26–33