A Dataset of Operator-client Dialogues Aligned with Database Queries for End-to-end Training

Ondřej Plátek and Filip Jurčíček

Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 11800 Praha 1, Czech Republic
{oplatek,jurcicek}@ufal.mff.cuni.cz

Abstract. This paper presents a novel dataset for training end-to-end task oriented conversational agents. The dataset contains conversations between an operator – a task expert, and a client who seeks information about the task. Along with the conversation transcriptions, we record database API calls performed by the operator, which capture a distilled meaning of the user query. We expect that the easy-to-get supervision of database calls will allow us to train end-to-end dialogue agents with significantly less training data. The dataset is collected using crowdsourcing and the conversations cover the well-known restaurant domain. Quality of the data is enforced by mutual control among contributors. The dataset is available for download under the Creative Commons 4.0 BY-SA license.

Keywords: task-oriented dialogue, end-to-end, dataset

1 Introduction

We present a new dataset of human-human task-oriented conversations in the restaurant information domain, suitable for supervised training of autonomous systems. We are currently collecting data for the set using crowdsourcing. We have collected 62 dialogues so far; more than 700 dialogues are planned for the final version of the dataset. [1]

We aim at representing the conversation in a way that makes it easy to train an automated system to replace a human operator. Our dataset contains easy-to-collect high-quality transcriptions of the client and operator actions during conversations. We introduce a very little overhead by collecting database calls in addition to the transcription of the conversation. In fact, we mimic very closely a real situation in call centers where operators search for answers through a database user interface. Just logging the calls to the task database provide us with information of a similar quality as a full manual dialogue state annotation. Tracking the database calls in task oriented systems together with conversation transcription should not introduce significant overhead during the data collection, but should provide a similar amount of supervision for training an end-to-end conversational agent as using fully annotated dialogue state.

[1] See the dataset at http://hdl.handle.net/11234/1756.

Current Spoken Dialogue Systems (SDS) are either handcrafted and use no training data [2,11] but require non-trivial amount of expert work, or they are gradually improved from initial policy through live user interaction [21,3]. Recently, SDS have also been trained using supervised learning in an end-to-end manner [17,20].[2] To train a system using reinforcement learning, one only needs to collect enough live user conversation containing explicit feedback. However, the feedback signal is harder to interpret than supervised annotation: Thousands of live conversations are required to improve a rather simple policy.[3] Supervised learning methods reduced the number of conversations needed for training a reasonable policy down to a few hundreds[17], but require explicit annotation for all components used in the dialogue system.[3] With the dialogue state being a simplification of the dialogue history with no broadly-accepted form, it is very expensive to collect such annotated data because the dialogue state typically differs among domains and its structure is hard to explain to annotators. In this paper, we focus on collecting a supervised training set with annotation rich enough so that supervised models similar to [17] can be trained with a few hundreds of dialogues, yet much easier to collect than previous methods.

We believe that our dataset dataset can be used for first experiments in training end-to-end systems using supervised learning without explicit dialogue state annotation. The restaurant information domain used in our dataset is well established thanks to dialogue state tracking (DST) challenges [19,4,5]. The DST challenges proved that the domain is of a reasonable complexity. In addition, the challenges showed that the DST task is solvable by architectures which may be extended to end-to-end conversational agents [10,15,6]. Note that our data-collection method is completely domain independent.

The paper is structured as follows: Section 2 motivates the annotations and collection procedure used in our dataset. In Section 3, we introduce our collection process using crowdsourcing. Section 4 presents the dataset properties, and we list related work in Section 5. We conclude the paper and suggest future work in Section 6.

2 Supervised Feedback for Task-oriented Dialogue

In this paper, we assume that we need to train a task-oriented conversational agent in a well-defined domain. We suppose that the agent has a task database[4] at hand and plays the role of the *operator*. The other interlocutor in the conversation is a human *client* that seeks information stored in the database.

The task of building a conversational agents is traditionally simplified by creating pipeline of subtasks: each solved by specialised components: automatic

[2] The work of [20] also fine-tuned the conversation with reinforcement learning after the supervised learning stage.

[3] To our knowledge, all current systems require full dialogue state annotation as a minimum [17,21].

[4] In the Cambridge restaurant domain the database is a simple table containing information about restaurants.

speech recognition (ASR), language understanding (LU) unit, dialogue tracker (DT), dialogue manager (DM), natural language generator (NLG) and text-to-speech (TTS) module [2]. The ASR [9] and TTS [13] modules can be easily trained without understanding the meaning of the input and there are typically trained domain independently. The crucial task of conversation agents is to understand the dialogue history and respond with a reply which has the greatest expected benefit for the agent. The task of understanding and reply generation is traditionally solved by the text-to-text pipeline of LU, DT, DM, and NLG components trained using just in-domain data which are annotated for each component [3,7,1]. Our datasets contains just the textual transcriptions of input from client and output of the operator from indomain conversations and in addition it includes simple annotations in form of database calls. We expect that supervised models will be able to model the meaning of the history and select a convenient reply only from the transcriptions and the annotations.

It was shown that a few hundreds of annotated domain specific conversations are needed to train a relatively well performing LU [7] as well as Dialogue State Tracking [21] components. The work of [1] and [8] showed that only several hundreds of turns with appropriate annotation is needed for training a usable NLG unit in simple domains. Even the crucial task of action selection can be trained using a reinforcement learning dialogue manager and few thousands conversations despite the fact that rewards are only provided at the end of a dialogue [3].

We believe that end-to-end systems can also be trained using a few hundreds of conversations for narrow goal-oriented domains, such as Cambridge restaurant domain [4] or its simplification [17], if additional supervision in the form of simple annotation is used. The work of [17] showed that only 680 conversations annotated just at the dialogue state level are needed for training a complete end-to-end system performing the task of summarizing the dialogue history, action selection, and response generation.

We realize that annotating the dialogue state according to an expert-handcrafted ontology is both an artificial and a very labor-intensive data collection task. Therefore, our work aims at collecting conversation with lighter and easier to get annotations. For our dataset, we collect the operator's database calls instead of dialogue state annotation after each turn A similar recording setup is known from call centers where the operator is provided with a database interface so that they quickly find relevant information and play the role of a domain expert. To obtain the dataset, we record conversations between a human operator and a human clients via a crowdsourcing platform, i.e., collecting cheap high quality training data.

The calls to the database represent very accurately the client's intention, which is the original purpose of dialogue state annotation. On the other hand, the database calls are not present in every turn and thus by logging only the calls, one loses the possibility to track partially expressed client's intention. However, we argue that in high quality human-human conversation, there are few turns which are related to the domain and do not contain calls to the database, as

demonstrated in Section 4. It is worth noting that logging the operator's database calls in operator-client dialogues is already a well-established practice in call centers, a convenient field for commercial application of dialogue systems.

3 Dataset Collection Process

Our data collection process aims at obtaining natural in-domain conversations. We selected the restaurant domain because it is convenient for collecting shorter, several turns-long conversations. Also, it was proven that the domain is on one hand difficult enough that even the state-of-the art systems do not reach human-level performance in the role of the operator [4] and on the other hand, numerous systems [21,3,17] demonstrated their ability to provide information to the clients with a reasonable quality. We define our domain by choosing the database from the 2nd Dialogue State Tracking Challenge (DSTC2) [4]. We use the CrowdFlower (CF) crowdsourcing platform to collect the conversations.

We first describe the database which defines our domain in Section 3.1. Then we describe how the hired workers use our interface to play either the client or the operator role in Sections 3.2 and 3.3, respectively. Finally, in Section 3.4, we describe how we collect conversations one response at a time without connecting the interlocutors in real-time.

3.1 Domain Database

We use exactly the same database as provided for the DSTC2 challenge [4]. The database contains information about 107 restaurants in Cambridge, and their properties are stored in several columns. We include in brackets the number of different values for each column:

- name (107),
- price range (4),
- area (7),
- food (25),
- address (101),
- phone number (97),
- postcode (62).

The address and the phone number are unique for each restaurant but are sometimes missing; the restaurant name is always known. There are typically multiple restaurants with the same postcode, price range, area, or food type properties; these properties might also be unknown for a restaurant.

We use this database to prepare client goals for CF users (see Section 3.2). The database is also available through a simple interface to CF users playing the role of the operator (see Section 3.3).

3.2 Client Interface

The CF worker playing the role of the client is asked to request specific information from the operator – a goal. They may choose not to seek the goal immediately but submit more appropriate response such as greeting instead. The goals are

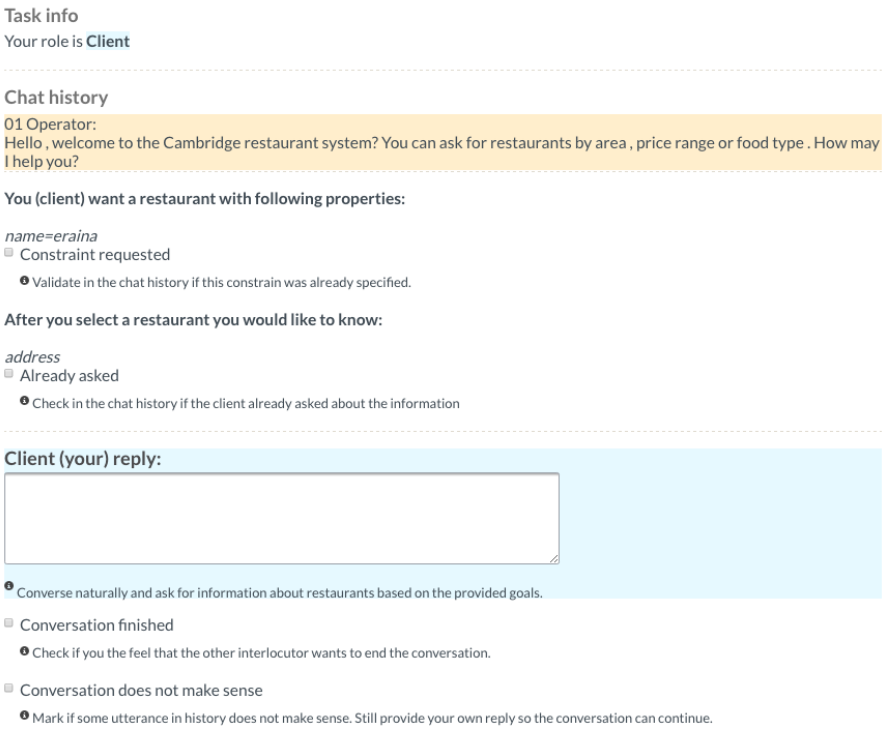

Fig. 1. Client annotation interface

displayed to the CF worker in the client role gradually turn-by-turn and may contradict each other. For example, the client may at first attempts to find a restaurant serving Chinese food, but later change their mind and seeks British food.

Before submitting a reply, the client is instructed to read the dialogue history and mark the goals which were already presented to the operator, as demonstrated in Figure 1. By marking which goals have already been presented to the operator, we oblige the CF workers to pay attention to the dialogue history before submitting their reply.

3.3 Operator Interface

The operator CF worker is asked to politely address the clients needs according to the information stored in the database. In order to provide the information to clients, the operators use the simple database interface depicted in Figure 2: They filter the restaurants using a full text search over the database. The search interface is implemented in Javascript as a simple substring matching over values in the database. We used such a simple interface because it is easy use for untrained CF workers. Multiple constrains can be expressed by two substrings separated by a comma. The content of the database is intentionally hidden. If no filter value is specified, the operator does not see content of the database so

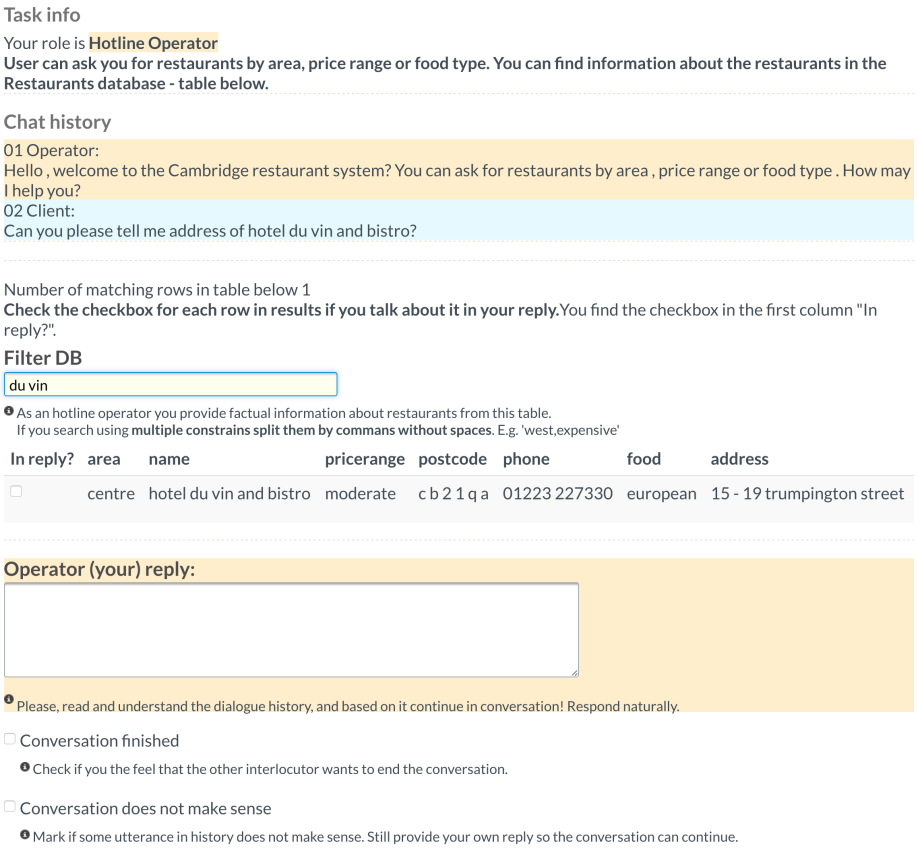

Fig. 2. Operator annotation interface

the filter has to be used to answer clients questions. The CF workers are also instructed to explicitly mark the database rows to which they refer in their answer. Similarly to the client role, the operator does not have to provide all requested information in a single turn. It is left up to the worker which information is provided first.

3.4 Asynchronous collection without real-time responses

In this section, we describe how we combine the workers' replies to form a conversation. We design the crowdsourcing jobs so that the CF contributors are able to work independently of each other. The roles of client and operator take turns as partial conversations are submitted to CF for collecting a single new response. As a consequence, a single conversation is collected from multiple contributors over several job submissions.

Each utterance in the conversation may be submitted by a different contributor. Therefore, the contributors need to read carefully the dialogue history to understand the conversation before submitting their response. The workers are

able to mark dialogues which do not make sense and thus they self-assess the quality of the conversations. The contributors mark the conversations as finished if they have no more goals to fulfill and all the goodbyes have been said.

We bootstrap the conversation either with an operator's introduction *"Hello , welcome to the Cambridge restaurant system..."*, or we bootstrap the conversation with a full turn where we add a client's greeting, e.g., *"Hi"* to the operator's introduction. If we initialize the dialogue with a full turn, the first contributor plays role of the system. If we bootstrap the conversation only with the operator's greeting, the first contributor plays the client's role.

In a single CF task submission, we let the workers append a single response to multiple different dialogues, taking on a random role in each of them. The conversations are typically rather short circa four turns and therefore, full dialogues are collected after few rounds of submissions. The workers are paid the same amount of money for every submitted response regardless of dialogue history length and whether they play the client or the operator role.

To avoid poor language level of English, we restricted the CF job to English-speaking countries. We further perform heuristic checks for too trivial replies, and we enable the CF workers to rate each other's responses. This help us filter low-quality conversations, but more importantly, it motivates the CF workers not to cheat.

4 Dataset Properties

We just started to collect our dataset and so far, we have only collected 62 conversations. The average length of a conversation is 3.8 turns, meaning that the contributors in both roles reply in average circa three times before the conversation is finished[5]. During an average conversation, 2.4 goals are requested and 1.9 answered. There are two reason why there is more goals requested than answered. We saw a rare case when the goal was actually answered, but was left marked as unanswered. However, more common case is that the client asks for several goals e.g. phone and address, but he is satisfied if the operator provides just one of the goals e.g. phone.

During the data collection process we discarded around 35 % of replies so far because the utterances were of low quality. Note that we discard only the low-quality utterance; the prefix of the conversation is submitted for finishing the conversation in a next round of CF tasks. Once we collect enough conversations, we intend to split them into training, development and test set.

5 Related Work

Our work is closely related to work of [19,4,5] in the sense that the conversation are held in the same domain. On the other hand, the collection process differs substantially because we do not use any artificial system in the operator's role; we

[5] The first system utterance is automatically generated and for some conversations, the first client reply is also generated automatically, as described in Section 3.4

rather focus on collecting high-quality human-human conversations. In addition, we do not collect complex LU and DST annotation in our work.

The relevant work of [17] includes a collection of human-human dialogues in a domain similar to ours but more restricted. Their dataset is not yet publicly available; therefore, we only compare to their statistics. The dataset is of similar size as we intend to collect, and a very similar collection scheme was used where Amazon Mechanical Turk workers submitted one reply at a time without being connected to each other in real-time. On the other hand, they collected explicit dialogue state annotation, which takes a significant annotation effort.

Another line of research [14] used CF to collect human-human conversations for interactive learning dataset. However, the collected dialogues were later annotated by expert annotators, which goes directly against our intentions to avoid any expensive annotation.

There is a significant amount of work which used Wizard-of-Oz experiments [18,16,12] for studying linguistic properties of dialogues, evaluating proof-of-concept of a dialogue system or even collecting bootstrap training data. However, to our knowledge, no other work than [17] have used cheap crowdsourcing workers to collect a full training dataset for supervised learning of dialogue management or end-to-end conversations.

6 Conclusion and Future work

We present a new human-human dialogue dataset with annotation of user intention in the form of operator's database calls. We also introduced a novel data collection setting which resembles work of operators in call centers and introduces minimum overhead to crowdsourcing workers. The collected dialogues are published under Creative Commons 4.0 BY-SA license.

We plan to increase the number of collected dialogues to 700 or more before the camera-ready deadline. We intend to use the dataset to train and evaluate an end-to-end conversational model which uses only operator's database calls as additional supervision to the recorded responses.

Acknowledgments We would like to thank Ondřej Dušek for useful comments. This research was partly funded by the Ministry of Education, Youth and Sports of the Czech Republic under the grant agreement LK11221, core research funding, grant GAUK 1915/2015, and also partially supported by SVV project number 260 333. It used language resources stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40c GPU used for this research. Cloud computational resources were provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Center CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. no. CZ.1.05/3.2.00/08.0144.

References

1. Dušek, O., Jurčíček, F.: Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. arXiv preprint arXiv:1606.05491 (2016)
2. Dušek, O., Plátek, O., Žilka, L., Jurcícek, F.: Alex: Bootstrapping a spoken dialogue system for a new domain by real users. In: 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue. p. 79 (2014)
3. Gasic, M., Jurcicek, F., Thomson, B., Yu, K., Young, S.: On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In: Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on. pp. 312–317. IEEE (2011)
4. Henderson, M., Thomson, B., Williams, J.: The second dialog state tracking challenge. In: 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue. vol. 263 (2014)
5. Henderson, M., Thomson, B., Williams, J.D.: The third dialog state tracking challenge. In: Spoken Language Technology Workshop (SLT), 2014 IEEE. pp. 324–329. IEEE (2014)
6. Henderson, M., Thomson, B., Young, S.: Word-based dialog state tracking with recurrent neural networks. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL). pp. 292–299 (2014)
7. Jurčíček, F., Dušek, O., Plátek, O.: A factored discriminative spoken language understanding for spoken dialogue systems. In: International Conference on Text, Speech, and Dialogue. pp. 579–586. Springer International Publishing (2014)
8. Mairesse, F., Gašić, M., Jurčíček, F., Keizer, S., Thomson, B., Yu, K., Young, S.: Phrase-based statistical language generation using graphical models and active learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 1552–1561. Association for Computational Linguistics (2010)
9. Plátek, O., Jurcıcek, F.: Free on-line speech recogniser based on kaldi asr toolkit producing word posterior lattices. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL). pp. 108–112 (2014)
10. Plátek, O., Bělohlávek, P., Hudeček, V., Jurčíček, F.: Recurrent neural networks for dialogue state tracking. arXiv preprint arXiv:1606.08733 (2016)
11. Raux, A., Langner, B., Bohus, D., Black, A.W., Eskenazi, M.: Let's go public! Taking a spoken dialog system to the real world. In: in Proc. of Interspeech 2005. Citeseer (2005)
12. Rieser, V., Lemon, O.: Learning effective multimodal dialogue strategies from wizard-of-oz data: Bootstrapping and evaluation. In: ACL. pp. 638–646 (2008)
13. Taylor, P.: Text-to-speech synthesis. Cambridge university press (2009)
14. Vodolán, M., Jurčíček, F.: Data collection for interactive learning through the dialog. arXiv preprint arXiv:1603.09631 (2016)
15. Vodolán, M., Kadlec, R., Kleindienst, J.: Hybrid Dialog State Tracker. CoRR abs/1510.03710 (2015), http://arxiv.org/abs/1510.03710
16. Walker, M., Hindle, D., Fromer, J., Di Fabbrizio, G., Mestel, C.: Evaluating competing agent strategies for a voice email agent. arXiv preprint cmp-lg/9706019 (1997)
17. Wen, T.H., Gasic, M., Mrksic, N., Rojas-Barahona, L.M., Su, P.H., Ultes, S., Vandyke, D., Young, S.: A network-based end-to-end trainable task-oriented dialogue system. arXiv preprint arXiv:1604.04562 (2016)

18. Whittaker, S., Walker, M.A., Moore, J.D.: Fish or fowl: A wizard of oz evaluation of dialogue strategies in the restaurant domain. In: LREC (2002)
19. Williams, J., Raux, A., Ramachandran, D., Black, A.: The dialog state tracking challenge. In: Proceedings of the SIGDIAL 2013 Conference. pp. 404–413 (2013)
20. Williams, J.D., Zweig, G.: End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. arXiv preprint arXiv:1606.01269 (2016)
21. Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., Yu, K.: The hidden information state model: A practical framework for pomdp-based spoken dialogue management. Computer Speech & Language 24(2), 150–174 (2010)