DOC2DIAL: a Framework for Dialogue Composition Grounded in Business Documents

Song Feng **Kshitij Fadnis** **Q. Vera Liao** **Luis A. Lastras**
IBM Research AI
sfeng,kpfadnis@us.ibm.com vera.liao@ibm.com lastrasl@us.ibm.com

Abstract

We introduce DOC2DIAL, an end-to-end framework for generating conversational data grounded in business documents via crowdsourcing. Such data can be used to train automated dialogue agents performing customer care tasks for the enterprises or organizations. In particular, the framework takes the documents as input and generates the tasks for obtaining the annotations for simulating dialog flows. The dialog flows are used to guide the collection of utterances produced by crowd workers. The outcomes include dialogue data grounded in the given documents, as well as various types of annotations that help ensure the quality of the data and the flexibility to (re)composite dialogues.

1 Introduction

There has been growing interest in using automated dialogue agents to assist customers through online chat. However, despite recent effort in training automated agents with human-human dialogues, it often faces the bottleneck that a large number of chat logs or simulated dialogues with various scenarios are required. Meanwhile, enterprises and organizations often own a large number of business documents that could address customers' requests, such as technical documentation, policy guidance and Q&A webpages. However, customers would still prefer having interactive conversations with agents instead of searching and reading through lengthy documents. Taken together, a promising solution is to build machine assisted agents that could perform task-oriented dialogues that are based on the content of the business documents.

In task-oriented dialogues for customer care, a recurrent theme is a diagnostic process – identifying the contextual conditions that apply to the customer to retrieve the most relevant solutions. Meanwhile, business documents often contain similar information, with prior conditions, in for example if-clauses or subtitles, followed by corresponding solutions. Therefore, these documents can be used to guide diagnostic dialogues–we call them *document-grounded dialogues*. For example, the sample business document in Figure 1 contains information for an agent to perform the dialogue on the right, where P-n (S-n) denotes text span n labeled a precondition (solution) and "O-D" denotes "out of domain". The preconditions are expressed in various ways such as subtitles or if-clauses, followed by corresponding solution if that precondition applies. In this work,

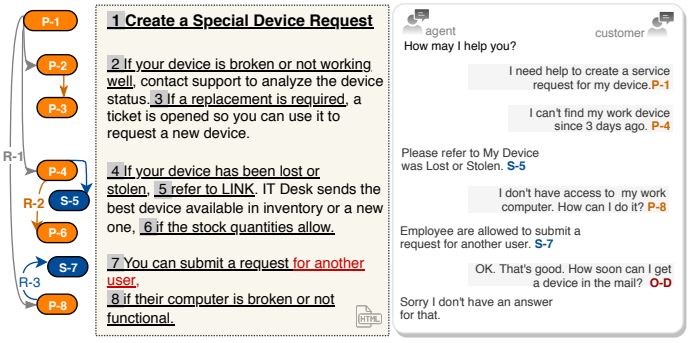

Figure 1: A sample conversational data grounded by a document.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

we hypothesize that an essential capability for a dialogue agent to perform goal-oriented information retrieval tasks should be to recognize the preconditions and their associated solutions covered by the given documents and then use them to carry out the diagnostic interactions. Towards this goal, we introduce DOC2DIAL, an end-to-end framework for generating conversational data grounded in business documents via crowdsourcing. It aims to minimize the effort for handcrafting dialog flows that is specific to the document but still introduce dynamic dialog scenes. It also provides quality control over the data collection process.

We guide our investigation with the following principles: 1) We aim to identify the document content that provides solution(s) to a user's request as well as describes the prerequisites required. 2) The generated dialog flows should be specific to the given document without relying on heavily supervised or handcrafted work. 3) The generated data tasks should be easy to scale – feasible to crowdsourcing platforms and could be updated with respect to changes in the document.

Thus, we propose a pipeline of three interconnected tasks for dialogue composition based on business documents: (1) labeling text spans as preconditions or solutions in a given documents (**TextAnno**); (2) identifying the relation(s) between these preconditions or solutions (**RelAnno**; (3) simulating dialog flows based on the linked preconditions/solutions and applying them to guide the collection of human generated utterances(**DialAnno**). For the dialogue collection, we could deploy it via both synchronized and asynchronized processes. An asychronized process allows crowd workers to work on the production and evaluation of individual turns without the constraints of timing or having a dialog partner. The outcome includes the document grounded dialogues as well as various types of implicit and explicit annotations that help ensure the quality of the data. Such data sets can be used to develop various types of dialogue agent technologies in the context of customer care.

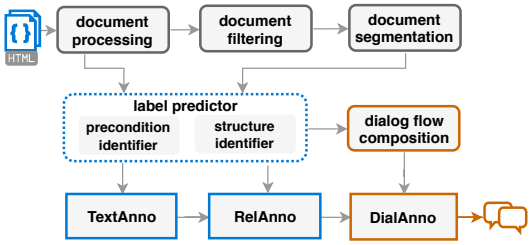

Figure 2: An overview of DOC2DIAL framework

Our primary contributions can be summarized as follows: (1) we introduce DOC2DIAL, an end-to-end framework [1] to generate task-oriented conversational data from business documents; (2) we propose a novel pipeline approach of three interconnected tasks that minimizes the dialog flow crafting manual effort and enables comprehensive cross-task validation to ensure the quality of dialogue data; (3) the system supports both synchronized and asynchronized dialogue collection processes. We demonstrate that such setting allows flexible dialogue composition by utterances, which are guided by the given document content and its annotations.

2 Related Work

Our work is mostly motivated by the success of research in training automatic agents with documents, as evidenced by the wide usage of data sets for machine comprehension tasks, e.g. [6]. Two recent data tasks, CoQA [7] and QuAC [1], aim to support conversational QA which involves understanding contexts and interconnected questions with multi-turn conversations. These data sets were created by pairing crowd workers to chat about a passage in the form of multi-turn questions and answers. Although getting closer to enabling automatic dialogue agents with documents, these data tasks do not tackle the understanding of preconditions, which is common in real-world task-oriented dialogues. In a recent paper, ShARC data task was proposed to address under-specified questions in conversational QAs by asking follow-up questions [8], which can be created by crowd workers based on supporting rules presented in a document. While it shares similar goals as our work on supporting dialogues that derive answers by understanding preconditions, it is a much simpler task focusing only on asking boolean follow-up questions. In contrast, our DOC2DIAL framework enables the end-to-end pipeline, from extracting complex precondition and solutions that present in business documents, to generating complete task-oriented dialogues that fit different scenarios.

[1] Code: `http://github.com/doc2dial/doc2dial-crowd`

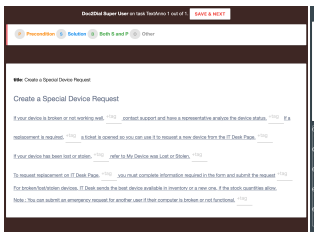

Figure 3: UI of **TextAnno**

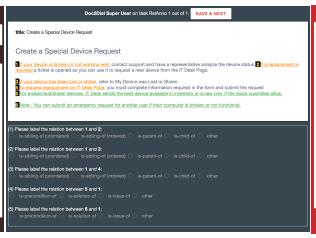

Figure 4: UI of **RelAnno**

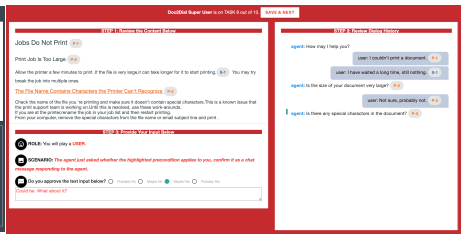

Figure 5: UI of **DialAnno**

3 DOC2DIAL Framework

Figure 2 presents the overview of our framework. Next, we first introduce the main tasks, and then describe a bit more about the core components on processing the documents for generating the tasks.

Main Tasks There are three tasks in our pipeline approach: (1) **TextAnno**, as in Figure 3, is for labeling individual text spans as `precondition`, `solution` or otherwise in a selected document. The resulted annotations can be used to automatically determine if a document is feasible for creating dynamic dialogues, thus enters the next steps in the pipeline. (2) **RelAnno** as seen in Figure 4 is to identify the relations between precondition pairs labeled in **TextAnno**, such as hierarchical relations i.e. `is-sibling`, `is-parent-of`, as R-1 and R-2 in Figure 1. The task can also include text pairs to be labeled as `is-precondition-of` or `is-solution-of` for cross validating the annotations from **TextAnno** (e.g. R-3 in Figure 1). (3)**DialAnno**, as shown in Figure 5, collects the dialog utterances based on the pre-generated dialog flow. Each turn correspond in the flow corresponds to a dialog scene, which is a composite of the interlocutor role (`agent` or `user`), a dialog action [5], a selected text span in the document based on the annotations from **TextAnno** and **RelAnno**, and dialogue history if applies. For instance, if the selected text is a `precondition`, the worker is given the role of an `agent`, the dialog action could be `request/query/open`. The worker is then given the instruction that *You need to know whether the highlighted precondition applies to the customer in order to narrow down the solution*. More examples of dialogue scenarios are given in Table 1.

Document Processing prepares the input documents for the three annotation tasks. We first obtain various syntactic-semantic analysis ranging from computational linguistic statistics to HTML-based tree structures. For instance, we apply constituency parsing results [3] for splitting long sentences to text spans for **TextAnno**. We also extract sub-clauses with certain discourse connectives (e.g. "if", "as long as") via [2] for identifying linguistic indicators of preconditions. In addition, we try to capture the outline patterns embedded in HTML tree structure in the documents [4]. Documents that are well structured and clearly written with descriptive sub-titles and the discourse connectives are considered as good candidates for generating the dialogues with dynamic flows.

Automated Labeling The system is equipped with the capabilities to automatically assign labels based on the linguistic indicators mentioned above. We also employ heuristics based on the HTML tree structures and text proximity for the relation annotations. Such fuzzy labels are mainly for allowing generating and testing the data tasks without human labels. They can also be used as pseudo-gold labels for quality control for the crowdsourced tasks.

Dialog Flow Composition generates the dialog flows for **DialAnno** from the labels of precondition/solution text and their relations obtained via **TextAnno** and **RelAnno**. For each turn, the dialog scene is determined by three factors, i.e., selected text span content, role and dialog act, which are determined sequentially. The dynamics of the dialog flows are introduced by varying the three aforementioned factors that are constrainted by the relations collected via **RelAnnp**. First, we randomly select content from a candidate pool of preconditions and solutions identified in the document, which is updated after every scenario generated. The general rule for updating the candidate pool is to eliminate preconditions/solutions that are already verified or eliminated. Then role is randomly selected between `AGENT` and `USER`. For our pilot study, we mainly consider dialog acts corresponding to the preconditions/solutions as selectively shown in Table 1. The dialogue ends when the candidate pool is empty, no solution is found, or it reaches the preassigned maximum turns.

LABEL	ROLE	DA	ACTION DESCRIPTION
precondition	user	request/query/open	Describe the highlighted text as an issue you have/haven't encountered.
solution	user	assert/provide/state	Describe the highlight solution as not working for you and seek for agent's help.
solution	user	request/query/open	Ask a question that can/can't be answered by the highlighted text.
precondition	agent	request/query/open	Verify whether the highlighted text is applied to the user.
solution	agent	respond/reply/open	Respond with the highlighted text that addresses the aforementioned issue(s).

Table 1: Sample Dialog Scenes.

4 Case Study

As a pilot study, we demonstrate how to apply DOC2DIAL to various documents and task contexts. The input were internal customer care documents on topics such as technical trouble shooting, policy guidance, etc. We experimented with 1900 documents from the candidate document pool after the automatic filtering. With these documents, we obtained 26,000 text spans labeled as either precondition or solution via **TextAnno** and **RelAnno**. Then, we generated 5 dialog flows per document and selected those with more than 5 turns. Next, we evaluated the dialog flows by asking the crowd worker if the highlighted text matched the given dialogue scenario of a turn in a dialog flow. 67% of the turns were labeled as "match". Most of the mismatches were due to disagreement on the precondition/solution labeled by crowd workers in earlier steps. Figure 5 shows the sample task on collecting the 6-th turn of a dialog flow of 7 turns by providing the dialogue scenario and the chat history. The sample dialogues show that when the crowd contributors were able to understand the selected text in the context of the document, they could properly interpret the dialog scene to produce utterance or evaluate the task.

5 Conclusion and Future Work

We introduced DOC2DIAL, an end-to-end framework and an implemented platform for simulating task-oriented conversations that are grounded in a given document content. We proposed a pipeline approach to minimize the effort of hand-crafting dialog flows and ensure quality control. Our system demonstrated the feasibility of collecting dynamic dialogues that are grounded by the documents of various tasks. Many challenges remain such as when the selected texts are not sufficiently informative for end users to improvise, which we hope to improve in the future work.

References

[1] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*, 2018.

[2] Debopam Das, Tatjana Scheffler, Peter Bourgonje, and Manfred Stede. Constructing a lexicon of english discourse connectives. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 360–365, 2018.

[3] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*, 2018.

[4] Saikat Mukherjee, Guizhen Yang, Wenfang Tan, and IV Ramakrishnan. Automatic discovery of semantic structures in html documents. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 245–249. IEEE, 2003.

[5] Silvia Pareti and Tatiana Lando. Dialog intent structure: A hierarchical schema of linked dialog acts. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[7] Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

[8] Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. Interpretation of natural language rules in conversational machine reading. *arXiv preprint arXiv:1809.01494*, 2018.