Simulating Domain Changes in Conversational Agents Through Dialogue Adaptation

Tiziano Labruna[1,2], Bernardo Magnini[1]

[1]Fondazione Bruno Kessler, Via Sommarive 18, Povo - Trento, Italy
[2]Free University of Bozen-Bolzano, Piazza Università 1, Bozen-Bolzano, Italy

Abstract
A major bottleneck for the large diffusion of data-driven conversational agents is that conversational domains are subject to continuous changes, which soon make initial dialogue models inadequate to manage new situations. In the current context, updating training data is usually carried on manually, and, in addition, there are no tools for simulating the impact of a certain domain change on the performance of the dialogue system. This position paper advocates that substantial progress in the capacity to simulate domain changes is based on the ability to automatically adapt training and test dialogues to those changes. We discuss the potential of a simulation framework for task-oriented dialogues, as well as the research challenges that need to be addressed.

Keywords
Dialogue Systems, Conversational Agents, Domain Adaptation

1. Introduction

Task-oriented dialogue systems [2, 3, 4] allow users to achieve specific tasks (e.g., booking a restaurant, buying a train ticket, ordering some food) through dialogues in natural language. While in recent years there has been a large diffusion of such conversational systems, a major bottleneck for their development, even for more complex tasks, is that conversational domains are very dynamic and are subject to continuous changes, which soon make initial dialogue models inadequate to manage new situations. As an example, a chatbot for giving information about covid-19 needs to be frequently updated, as new regulations are introduced and others are changed. A similar issue happens in the case of booking restaurants in a region, where new restaurants open and others introduce new food. In such situations initial dialogue models (e.g., intent and slot-filling) soon become obsolete and the system performance rapidly decreases. The current practice in case of domain changes consists of manually updating the training dialogues, typically adding sentences with new intents and entities that reflect the changes. However, this practice is extremely expensive and requires specialized competencies. In addition, there are no tools for simulating the impact that a certain domain change might have on the performance of the dialogue system and its components. Being able to approximate the impact of, for instance, adding or removing a certain slot in the system knowledge base, would allow a

NL4AI 2022: Sixth Workshop on Natural Language for Artificial Intelligence, November 30, 2022, Udine, Italy [1]

tlabruna@fbk.eu (T. Labruna); magnini@fbk.eu (B. Magnini)
0000-0001-7713-7679 (T. Labruna); 0000-0002-0740-5778 (B. Magnini)

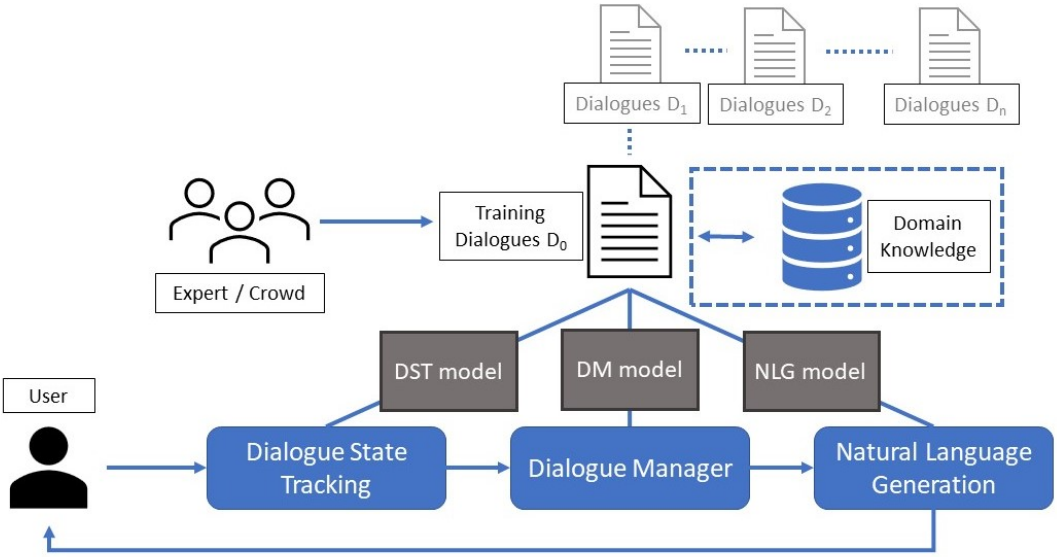

Figure 1: Representation of a typical data-driven conversational system flow. The user sends a message, the message is parsed by a dialogue state tracking component, the output is passed to a dialogue policy component, which decides the best next action of the system, and finally, a natural language generation component generates the utterance to be returned to the user. Each component is based on a model, which is, in turn, trained on some dialogues, typically created by hand and linked to a knowledge base. The dialogues will vary when there are domain changes.

more precise estimation of the re-training costs, with a significant saving of time and money. Although dialogue simulators have been proposed (e.g., Simdial [5]), to the best of our knowledge, none of them is designed to simulate domain changes.

In this position paper, we rise a number of research challenges that need to be considered when designing a dialogue simulator able to account for domain changes. First, we need to fix a reference architecture for the dialogue system, including the main dialogue components (e.g., intent detection and slot filling, dialogue manager, response generation). Second, define the experimental parameters of the simulator, i.e., which data can be manipulated, such as the kind and amount of changes and the models for the dialogue components. Finally, a relevant challenge for a dialogue simulator able to manage domain changes concerns performance evaluation. More specifically, in order to evaluate a certain change (e.g., a new type of food for a restaurant is introduced, which was not present before), we need a gold standard (i.e., test dialogues) reflecting the changes we intend to simulate. In this paper, we suggest that recent dialogue adaptation techniques [6, 7] can be applied for the automatic creation of test dialogues to be used in a dialogue simulator. We also suggest that the generative power of recent pre-trained language models may offer encouraging opportunities in the direction of automatic dialogue adaptation.

2. Dialogue System's Architecture

Figure 1 depicts a general architecture of a data-driven conversational agent, showing three main components: Natural Language Understanding (NLU), Dialogue Manager (DM) and Natural Language Generation (NLG). The user sends the message to the agent, the NLU component is responsible for extracting relevant information from the message and passing it to the DM component, which, based on that information, decides which action to take; finally, the NLG component takes the action as input and returns a natural language message to be sent back to the user.

- **Natural Language Understanding**. The goal of the NLU component [8] is to extract relevant information from the user message. This information typically consists of an *intent* (the communicative goal of the user's utterance) and a certain number of *entities* that can be contained in the message. The prediction of the former is known as Intent Recognition, while the prediction of the latter is called Entity Extraction or Slot Filling. The prediction of intents and entities is usually evaluated in terms of accuracy and f1-score.
- **Dialogue Manager**. The DM component takes an intent and a certain number of entities (possibly empty) as input and returns the best next action to take, as the output, which typically consists of an intent and some slot-values. While taking this decision, the dialogue manager also considers some state variables, such as the conversation history up to a certain point in the past. The selection of the best action is usually evaluated in terms of accuracy.
- **Natural Language Generation**. As the last component of this process, NLG is responsible for converting the output of DM into words. This means that it needs to take a structured representation of information and produce a natural language utterance that will be returned to the user. The correct generation of the utterance is evaluated using string comparison metrics (a common one is BLEU [9]).

3. A Simulator for Domain Changes

We propose a methodology to investigate the impact of domain changes in dialogue systems based on a *Domain Changes Simulator* (DCS), an architecture that simulates different types and different amounts of domain changes, chooses a model for every dialogue component and produce a report on the performances of the models given a certain configuration of the simulator.

Domain changes. Domain knowledge in a task-oriented dialogue is typically represented in a Knowledge Base (KB), where instances of concepts (e.g., RESTAURANT) are described through slots that can assume a range of values. We consider the following domain changes:

- **Concept changes**. Concepts (also referred to as domains in the literature) delimit the topics that can be discussed in a conversation. A concept can be removed (e.g., an agent does not cover information on restaurants anymore), or added (e.g., an agent starts giving information about hotels).

- **Slot changes**. Slots are associated with concepts and describe the characteristics of the concept instances. Slots can be added (e.g., there is new interest whether a restaurant has PARKING), or can be removed (e.g. we are no longer interested whether a hotel has an INTERNET_CONNECTION).
- **Instance changes**. Instances are individual entities in the conversational domain (e.g., a specific restaurant). Instances can be added (e.g., a new restaurant opens), or removed (e.g. a restaurant closes down).
- **Slot-value changes**. Slot-values are used to describe properties of instances (e.g., the MARIO's restaurant offers ITALIAN food). A new slot-value can be introduced in the KB (e.g. CARIBBEAN food starts to be served by some restaurants), a slot-value can disappear from the Knowledge Base (e.g. no more restaurants serve INDIAN food), or an instance can change its slot-value (e.g., when a restaurant changes its menu).

We are interested in simulating and assessing the impact, of all the changes described above, through configurations of the DCS dialogue simulator. As a first attempt to simulate domain changes in a task-oriented dialogue, we are experimenting on the RASA platform [10] simulating changes over the MultiWOZ dataset [11].

Dialogue Models. In addition to domain changes, the DCS simulator should be able to consider different models for the dialogue components described in section 2. The NLU component requires an annotated collection of user utterances in natural language, in order to be able to recognise and extract intents and entities from a message; the DM component requires a set of dialogues with annotations of intents and entities from the user and related intents of the answers from the system; the NLG component requires a list of natural language utterances for every system's intent. Each model can be more or less robust to domain changes, thus having different degrees of generalization and requiring more or less exhaustiveness of the training data. Some models, for example, are able to perform few-shot or even zero-shot learning, by leveraging techniques such as schema-guided algorithms [12, 13].

4. Evaluating the Impact of Domain Changes

The main purpose of the DCS simulator is to access how the performances of the dialogue components evolve when domain changes occur so that it is possible to estimate their impact on the system. A crucial issue here is to develop test data for each component and for each configuration of domain change we are interested to evaluate. Test data vary according to the dialogue component: we need dialogue annotated with intents and slot-value pairs for NLU, actions to be performed by the system at each dialogue turn for the Dialogue Manager and reference system responses for the NLG component. While in principle such test data should be collected through human intervention (e.g., Wizard of Oz), this is practically impossible given the high number of potential configurations we want to simulate.

To overcome this issue, we are proposing a *dialogue adaptation* strategy for the automatic

creation of the test data to be used by the DCS simulator. The idea behind dialogue adaptation is that domain knowledge described in the KB is somehow reflected in training and test dialogues, and that, when a domain change occurs, it is possible to adapt the initial dialogues so that the change is adequately reflected. More formally, we define the problem of Dialogue Adaptation as follows: starting from a dialogue D_0 collected for a certain knowledge base KB_0, the goal is to modify D_0 such that it reflects a knowledge base KB_1, where KB_0 and KB_1 share the same domain ontology O (i.e., they share domain entities and slots).
Dialogue adaptation has different complexity depending on the changes introduced in Section 3. Concept changes require that all dialogues referring to a certain concept (e.g., RESTAURANT) are removed, or substituted with dialogues with a different concept. This is highly complex, as it implies the ability to automatically regenerate a full dialogue. All dialogue components are affected by concept changes. Slot changes require that full portions of a dialogue, e.g., a turn referring to a certain slot, are changed to reflect a newly introduced slot. Instance changes mainly affect DM decisions about the next action. Finally, slot-value changes have reduced complexity and can be addressed through local substitutions within single turns. As regards possible approaches to domain adaptation, different strategies can be employed, spanning from rule-based to generative approaches. We have experimented with both of them, on a subset of domain changes and for the NLU component only, in our previous works [14, 15, 7], showing that the use of a pre-trained language model, fine-tuned on the target domain KB, achieves promising results. However, a simulation environment imposes more strict constraints on dialogue adaptation, not only regarding the capacity to manage different types of domain changes, but also the capacity to simulate fine-grained amounts of such change (e.g., add 20% of new RESTAURANT instances serving POKE food, and remove all dialogues mentioning PARKING).

5. Conclusion

This position paper suggests a long-term research direction aimed at simulating the impact that domain changes might have on the performance of a conversational system. Such a simulator allows to manipulate and set a number of experimental parameters, including several types and different amounts of changes, and various algorithms for training dialogue components, making it easier and less expensive to develop and maintain a conversational system. A major research challenge for a domain change simulator is the capacity to automatically generate test dialogues that approximate the domain changes. This capacity is crucial for evaluation purposes, and can be achieved through incremental substitutions in the initial training dialogues, exploiting the generative power (e.g., masked tokens, prompting) of pre-trained language models.

References

[1] D. Nozza, L. Passaro, M. Polignano, Preface to the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI), in: D. Nozza, L. C. Passaro, M. Polignano (Eds.), Proceedings of the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI 2022) co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2022), November 30, 2022, CEUR-WS.org, 2022.

[2] M. McTear, Conversational ai: Dialogue systems, conversational agents, and chatbots, Synthesis Lectures on Human Language Technologies 13 (2020) 1–251.

[3] S. Young, M. Gašić, B. Thomson, J. D. Williams, Pomdp-based statistical spoken dialog systems: A review, Proceedings of the IEEE 101 (2013) 1160–1179. doi:10.1109/JPROC.2012.2225812.

[4] M. Henderson, B. Thomson, J. D. Williams, The second dialog state tracking challenge, in: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), Association for Computational Linguistics, Philadelphia, PA, U.S.A., 2014, pp. 263–272. URL: https://www.aclweb.org/anthology/W14-4337. doi:10.3115/v1/W14-4337.

[5] T. Zhao, M. Eskenazi, Zero-shot dialog generation with cross-domain latent actions, in: Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1–10. URL: https://www.aclweb.org/anthology/W18-5001. doi:10.18653/v1/W18-5001.

[6] T. Labruna, B. Magnini, Addressing slot-value changes in task-oriented dialogue systems through dialogue domain adaptation, in: Proceedings of RANLP 2021, 2021. URL: https://aclanthology.org/2021.ranlp-1.89.pdf.

[7] T. Labruna, B. Magnini, Fine-tuning bert for generative dialogue domain adaptation, in: Text, Speech, and Dialogue, 2022, pp. 490–501.

[8] S. Louvan, B. Magnini, Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 480–496. URL: https://www.aclweb.org/anthology/2020.coling-main.42. doi:10.18653/v1/2020.coling-main.42.

[9] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.

[10] T. Bocklisch, J. Faulkner, N. Pawlowski, A. Nichol, Rasa: Open source language understanding and dialogue management, arXiv preprint arXiv:1712.05181 (2017).

[11] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, M. Gašić, Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling, arXiv preprint arXiv:1810.00278 (2018).

[12] J. Cao, Y. Zhang, A comparative study on schema-guided dialogue state tracking, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 782–796.

[13] Z. Lin, B. Liu, S. Moon, P. Crook, Z. Zhou, Z. Wang, Z. Yu, A. Madotto, E. Cho, R. Subba, Leveraging slot descriptions for zero-shot cross-domain dialogue state tracking, arXiv preprint arXiv:2105.04222 (2021).

[14] T. Labruna, B. Magnini, From cambridge to pisa: A journey into cross-lingual dialogue domain adaptation for conversational agents (2021).

[15] T. Labruna, B. Magnini, Addressing slot-value changes in task-oriented dialogue systems through dialogue domain adaptation, International Conference Recent Advances In Natural Language Processing (2021).