Chatbots - keeping track of context

Carl Balmer
University of Bern
Matr. Nr.: 13-120-431
Email: carl.balmer@students.unibe.ch

Mathias Fuchs
University of Bern
Matr. Nr.: 09-923-764
Email: fuchsmat@students.unibe.ch

***Abstract*—Nowadays chatbots become more and more sophisticated conversationalists, due to recent advances in the field. Chatbots are especially popular in handling customer service tasks. However it is crucial for a bot to be able to keep the context of a conversation. In this paper we give an overview over the different ways of contexts, the current state of the art in context tracking and we test a nerual network approach in an experiment, using the ubuntu dataset [1].**

I. Introduction

As the popularity of chatbots increases it becomes more important to increase their quality. This is why it is crucial for a chatbot to be able to keep track of the context. For example should the bot be able to know the nationality of the person using it or whether a person means his mother when saying ”she” or his wife.

There are different types of context: The world knowledge(time, location, weather) , the user knowledge(relationships, preferences) and the dialogue context(Knowledge learned during the conversation), which is also called dialogue state. (need citation here?). In the following sections we give a brief overview over all those types. However our main focus will be on the dialogue context and the most common ways used to track it.

There are different approaches used for dialogue context:

- Rule-based approaches
- Probabilistic approaches
- Data driven approaches

Depending on which domain we are in, different approaches are more suited. There are two domains:

- Open domain
- Closed domain

II. Types of context

A. World knowledge

B. User knowledge

C. Dialogue context

Dialogue context, also called dialogue state, represents the challenge of keeping track of the intent of the user and the knowledge learned during the conversation. For example:

U1: Person: I would like to see the best italian restaurant
U2: Conversational Agent: Hey, it is ”luigis pizza”. Would you like to make a reservation?
U3: Person: No, please show me Chinese restaurants
U4: Conversational Agent: We have ”Restaurant A”, ”Restaurant B” and ”Restaurant C” which are close. Which one do you like?
U5: Person: Ok I’ve changed my mind. Make a reservation at the Italian place

In the above example the conversational agent needs to know what the user means in U5 by ”the Italian place”. It has to realize that the user is referring to the Italian restaurant mentioned in U2. Depending on how far back the context goes, it can be hard to track. Often the chatbot needs to ask again what the user means, because it did not understand.

III. The context tracking Problem

We can define the general case of the problem as follows: $[u_1, a_2, u_3, a_4, u_5, ..., u_t] \rightarrow a_{t+1}$
Where

- u_i are the user actions which consist of stating an intent, give information, provide feedback to agent answers or change the intent or the given information.
- a_j are the agent actions which consist of requesting more information, give appropriate responses and request feedback.

In this case the agent decides for the appropriate response based on **all** previous utterances.

If we look at the Problem of Dialogue State Tracking in a general case, we can formulate it like this: The agent takes as input all previous utterances by both parties and then decides based on this, what the most appropriate response is. At any given point in the conversation the user and the agent both can take multiple actions. The user may state his Intent and/or give additional information (e.g. I want to ride the bus from $\rightarrow$ to). The agent can then reply by requesting additional information or presenting the user with a response and may ask for feedback

The hard thing is to track all of the information and notice changes in the users intent, the information and to incorporate feedback.

A. Closed vs. open domain

There are two domains [1] in which a chatbot can operate in.

- **Closed domain**: The bot has one specific task it has to operate in e.g. music player, restaurant finder. The

[1] http://dataset.cs.mcgill.ca/ubuntu-corpus-1.0/

advantage here is, that all the possible actions are finite and known in advance [2] (e.g. there is a finite number of Italian restaurants in a certain area).

- **Open domain**: The bot has no specific task. In the most extreme case it is completely open, which means that it has to converse with anyone about anything. This makes it impossible to model a set of possible answers beforehand.

Fig. 1. Arrow representation of closed vs.open domain

In 1 The We have made the observation that for closed domain rule based methods work and the harder(more open) it gets, the more deep learning is used. Before we talk about different approaches and methods to tackle this problem, we quickly have to mention the importance of domain size because this greatly influences the feasibility and effectiveness of different methods.

In a closed domain our agents goal is to perform a specific, well defined task (eg. control music playback). This for this task all possible interactions between the user and the agent are finite and known in advance. This means that it would be possible to model the agent as a set of states with transitions.

On the other side of the spectrum we have a completely open domain, where the conversation can go anywhere. There we have an infinite ammount of possible states / user interaction and it is impossible to model/plan bevorehand.

These are no hard rules/borders most chatbots fill fall somwhere along this axys. (eg. Helpdesk) But in general we can say that the task gets more difficult, the more open a domain is. For the closed domain we can user rule based approaches and open domain deep learning makes more sense - but we will tak about that in the next slide.

IV. State of the art approaches

A. Rule-based approaches

B. Probabilistic approaches

C. Data driven approaches

V. Ablation Study

A. Goal

B. Dataset

C. Model architecture

D. Methology

E. Results

VI. Conclusion

The conclusion goes here. [3]

References

[1] R. Yan, Y. Song, X. Zhou, and H. Wu, "Shall i be your chat companion?: Towards an online human-computer conversation system," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 649–658.

[2] F. Radlinski and N. Craswell, "A theoretical framework for conversational search," in *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*. ACM, 2017, pp. 117–126.

[3] S. Chopra and S. Chivukula, "My phone assistant should know i am an indian: influencing factors for adoption of assistive agents," in *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 2017, p. 94.