Towards an Open-Domain Social Dialog System

Maria Schmidt, Jan Niehues and Alex Waibel

Abstract This demo paper describes a text-based, open-domain dialog system developed especially for social, small talk-like conversations. While much research is focused on goal-oriented dialog currently, in human-to-human communication many dialogs do not have a predefined goal. In order to achieve similar communication with a computer, we propose a framework which is easily extensible by combining different response patterns. The individual components are trained on web-crawled data. Using a data-driven approach, we are able to generate a large variety of answers to diverse user inputs.

1 Introduction

Currently, goal-oriented dialog systems are still in the primary focus of dialog system research. In these systems, a clear goal of the dialog is defined and the user should be primarily provided with their needed information in an efficient way.

In contrast to this, in small talk-like dialogs as modeled in the approach presented in this work, no clear dialog goal is given. The main target is to enable diversified dialogs that keep the user interested in the conversation.

A similar situation to the one described in this paper was targeted in [3]. They developed a chat-like conversation system, but in contrast to this work, their work focuses on the development of a dialog strategy based on a Markov Decision Processes (MDP) framework.

Maria Schmidt, Jan Niehues and Alex Waibel
Karlsruhe Institute of Technology, Institute for Anthropomatics and Robotics, Karlsruhe, Germany
e-mail: `firstname.lastname@kit.edu`

2 System Description

The presented dialog system consists of different modules m_i, which analyze the input and generate a response r annotated with a confidence $c_i(r)$. These modules are developed to generate different types of answers. For example, we develop a module for questions and one to present additional information to the user. Furthermore, these answers belong to different topics, such as entertainment, sports, latest news or business.

Afterwards, a response combination is used to evaluate the different responses and select the most appropriate one. Figure 1 depicts the system and its modules.

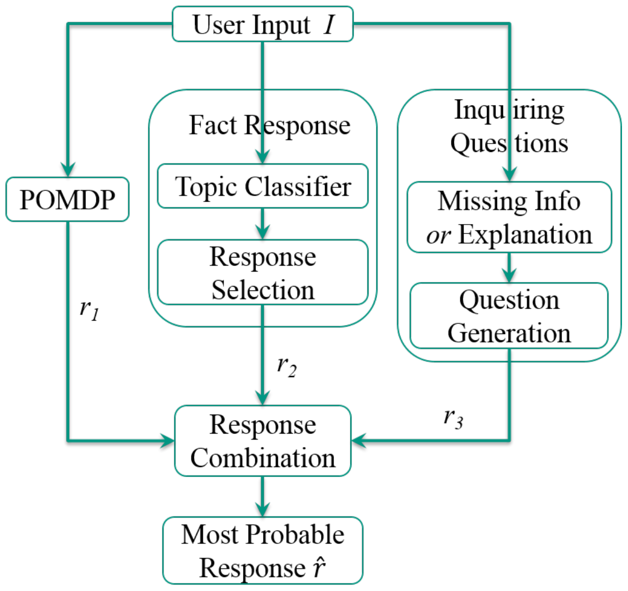

Fig. 1 System Overview

Furthermore, we keep track of the dialog flow and its continuity with session objects, in which we store dialog states, current answers and a history. Thereby, we can for example check whether the now proposed answer has already been mentioned in the last few dialog turns.

2.1 Response Combination

The different modules m_i of the system generate possible responses r and their confidence $c_i(r)$. The response combination selects the most probable answer and presents it to the user. In conventional settings in order to train a classifier, we need user input I from many users along with the best response $\hat{r}$ given a set of possible responses R and their respective confidences $\{(r, c_i(r))\}$.

2.2 Fact Response

Our Fact Response component is responsible for giving facts back to the user. One way of responding to the user is to provide additional information about the current conversation topic. The challenges here are to find answers that are interesting to the user and simultaneously matching the current topic.

2.2.1 Topic Classifier

The Topic Classifier serves as an NLU component which determines the topic the user wants to talk about.

Given the user's input $I = w_0, \ldots, w_N$ with an arbitrary number of words, the topic classifier assigns a probability to the triple (t, a, k) where t is a topic, a is an article and $k \subseteq I$ is a set of keywords in I.

The article probability is defined by

$$P(a|t,k) = \frac{1_{a \in t} \sum_{k_i \in k} e^{tfidf(a,k)}}{Z(t,k)} \tag{1}$$

For training the language models (LMs), we applied the KenLM toolkit[1] [2].

2.2.2 Response Selection

After assigning probabilities to the different articles in the database, the response selection picks an answer and assigns a confidence to it. In order to generate different answers for a classified topic t, the responses are drawn randomly given a probability distribution. The probability of a response r is given by:

$$P(r|a,t,k) = \frac{1_{|k \cap r| > 0}}{Z(a,t,k)} \tag{2}$$

[1] `http://kheafield.com/code/kenlm/`

2.3 Inquiring Questions

In this system, we use two different templates for inquiring questions. First, we search for information that are not given in the input and ask for these information. A second possibility is to ask for an explanation, since the user has given us information we do not understand.

2.3.1 Missing Information

We want to analyze the sentence in order to find information the user did not give to the computer. An example input would be a sentence like *I had many meetings today*. If we have this sentence, we could ask things like *What were the meetings about?* or *With whom did you meet?*.

2.3.2 Explanation

A different word template is to ask for explanation about unknown terms. Like in a human-human conversation, if there appears a word or phrase the system does not know, it should ask the user for an explanation of the term. Since we use training data covering many topics, we assume that a word, which is unknown to the system, should be a quite specific term, e.g., a name only known by some people. Therefore, it is reasonable to assume that a person in the conversation might also not know this term and would ask who or what it is.

Therefore, we use the training vocabulary to find unknown words in the input sentence and mark them as possible question items. Similar to the missing information, the actual questions for explanation are generated by the Question Generation module which is described in the upcoming section below.

This module is similar to the function of the OOV component in [1].

2.3.3 Question Generation

Given the two previously mentioned modules, we have annotated the input with possible missing information like *"meetings with"* and out-of-vocabulary words. In order to ask for this information, we need an NLG module to create questions based on the given input words.

2.4 POMDP Goal-Oriented Dialog

Our system contains a POMDP component in order to be able to handle goal-oriented dialogs in addition. Our implementation is based on the state-of-the-art technique shown in Young [4]. Currently, this subsystem is trained to work on a restaurant information scenario.

3 Results

For training the fact response, we used Wikipedia articles from the simple English edition and we crawled news articles from RSS feeds. We used the simple English Wikipedia edition, since in dialogs usually simpler and shorter sentences are used than in written text. For the news, we use articles published within the last three weeks. We use the articles from 5 different topics of Wikipedia and 12 RSS feeds. The resulting number of articles and facts are displayed in Table 1.

	Wikipedia	**News**
Topics	5	12
Articles	90,514	2,896
Facts	317,732	32,993

Table 1 Fact Response data

We trained the possible questions for missing information from the OpenSubtitle corpus and three additional English corpora. In total, these corpora contain 73M words.

Figure 2 illustrates example dialog turns of our social DS for RSS feed information, missing information and an explanation dialog.

U: FC Bayern lost this week. Barca was too strong for them.
S: Pep Guardiola will return to former club Barcelona for the first time with Bayern Munich after the two clubs were drawn against each other in the European Champions League semifinal.

U: I will go now
S: Go to?
U: I will walk to the hotel
S: Walk with?
⋮
U: Nobody. I am alone at the IWSDS
S: Iwsds?
U: It is a conference on spoken dialog systems

Fig. 2 Example of a dialog: RSS feed info, missing information, explanation

4 Demo Setup

The demo is accessible via a web interface. It consists of two text boxes, where the upper, bigger one is capturing the previous dialog with user and system turns indicated by `U` and `S`, respectively. The lower, smaller text box displays the user input as it is typed in. By hitting `Enter` the user submits their input which will be displayed in the upper text box followed by the corresponding system response.

5 Conclusion

In this system we have implemented a data-driven approach towards modeling open-domain social dialogs which is accessible through a website. We have shown a system architecture that is easily extensible and can generate different types of responses. Motivated by human small talk, we develop different modules to generate diverse responses to the user input.

References

[1] Banchs RE, Li H (2012) Iris: A chat-oriented dialogue system based on the vector space model. In: Proceedings of the ACL 2012 System Demonstrations, Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '12, pp 37–42, URL `http://dl.acm.org/citation.cfm?id=2390470.2390477`

[2] Heafield K, Pouzyrevsky I, Clark JH, Koehn P (2013) Scalable modified Kneser-Ney language model estimation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, pp 690–696, URL `http://kheafield.com/professional/edinburgh/estimate\_paper.pdf`

[3] Shibata T, Egashira Y, Kurohashi S (2014) Chat-like conversational system based on selection of reply generating module with reinforcement learning. In: Proceedings of the 5th International Workshop on Spoken Dialog Systems

[4] 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. Computer Speech & Language 24(2):150–174