

A Context-aware Natural Language Generation Dataset for Dialogue Systems

Ondřej Dušek, Filip Jurčiček

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

Abstract

We present a novel dataset for natural language generation (NLG) in spoken dialogue systems which includes preceding context (user utterance) along with each system response to be generated, i.e., each pair of source meaning representation and target natural language paraphrase. We expect this to allow an NLG system to adapt (entrain) to the user’s way of speaking, thus creating more natural and potentially more successful responses. The dataset has been collected using crowdsourcing, with several stages to obtain natural user utterances and corresponding relevant, natural, and contextually bound system responses. The dataset is available for download under the Creative Commons 4.0 BY-SA license.

Keywords: natural language generation, entrainment, task-oriented dialogue

1. Introduction

We present a new dataset intended for fully trainable natural language generation (NLG) systems in task-oriented spoken dialogue systems (SDS). It is, to our knowledge, the first dataset of its kind to include preceding context (user utterance) with each data instance (source meaning representation and target natural language paraphrase to be generated, see Figure 1). Taking the form of the previous user utterance into account for generating the system response should presumably improve the perceived naturalness of the output, and may even lead to a higher task success rate (see Section 3.). Crowdsourcing has been used to obtain natural context user utterances as well as natural system responses to be generated. The dataset covers the domain of public transport information and is released under a permissive Creative Commons 4.0 BY-SA license.

NLG systems in current SDS are in most cases handcrafted, e.g., (Rudnicky et al., 1999; Raux et al., 2005). Such systems are efficient and maintainable for limited domains, but provide little to no variance in their outputs, which makes them repetitive. Their scalability is also limited (Mairesse and Walker, 2011). Recent fully trainable NLG systems for SDS typically use random sampling to provide variance in outputs (Mairesse et al., 2010; Wen et al., 2015a; Wen et al., 2015b). This is perceived as more natural by the users, but still lacks adaptation to previous context, which is the norm in human-human dialogues.

We believe that the present dataset can be used for proof-of-concept experiments studying context adaptation in human-computer dialogues and that the results will be applicable to other domains as well as open-domain and chat-oriented systems. The method used to collect the data is completely domain-independent.

This paper is structured as follows: Section 2. introduces the task of NLG in SDS and describes the dialogue system and domain used in data collection. We give a brief explanation of the phenomenon of dialogue alignment, or entrainment, in Section 3. Section 4. then contains a description of the data collection process. We outline the main properties of the dataset in Section 5., and we list related works in Section 6. Section 7. then concludes the paper.

```
inform(vehicle=subway,line=C,  
      from_stop=Bowery,to_stop=Central Park,  
      departure_time=10:04am)
```

OK, take the C subway from Bowery heading for Central Park at 10:04am.

Figure 1: An example of NLG input (top) and output (bottom) in a task-oriented SDS

2. Natural Language Generation in Task-oriented Spoken Dialogue Systems

We understand the task of NLG in the context of task-oriented SDS which use *dialogue acts* (DA) to represent meaning (Young et al., 2010; Jurčiček et al., 2014). A DA represents a specific system or user action, such as *hello*, *inform*, *confirm*, or *request*. It is typically accompanied by one or more *slots* (variables) which may take specific values. The job of NLG in this context is to translate an input DA into one or more sentences in a natural language. An example input-output pair is shown in Figure 1.

We use the domain of English public transport information as implemented in the Alex SDS framework (Jurčiček et al., 2014; Vejman, 2015). It is a mixed-initiative dialogue system using Google Maps API to find public transit directions among bus and subway stops on Manhattan.¹ The user is able to specify a time preference or select a means of transport; they may ask for duration or trip distance.

3. Entrainment in Dialogue

Entrainment in dialogue, also referred to as alignment or adaption, is the mutual linguistic convergence of speakers during the course of a conversation. Speakers are primed (influenced) by previous utterances (Reitter et al., 2006) and tend to reuse vocabulary, syntactic structure, and prosody (Levitan, 2014) (see Figure 3). Entrainment occurs naturally and subconsciously and facilitates successful conversations (Friedberg et al., 2012).

¹The Alex system handles a larger domain, but we limited it to prevent data sparsity when collecting our dataset.

Nenkova et al. (2008) have shown that higher entrainment in frequent words correlates with a higher success rate in task-oriented human-human dialogues. Users have been reported to entrain naturally to prompts of a SDS (Stoyanchev and Stent, 2009; Parent and Eskenazi, 2010).

There have been several attempts to introduce a two-way entrainment into SDS, i.e., let the system entrain to user utterances. Hu et al. (2014) report an increased naturalness of the system responses, while Lopes et al. (2013) and Lopes et al. (2015) also mention increased task success. All of these approaches focus on lexical entrainment and are completely or partially rule-based.

Using the present dataset, we are planning to take entrainment even further in the context of a fully trainable NLG and train a system that adapts to users' lexical as well as syntactic choices. We hope that this will further increase both perceived naturalness of the system responses and overall task success rate.

4. Dataset Collection Process

When collecting the dataset, we aimed at capturing naturally occurring entrainment between pairs of user utterances and system responses. Collecting complete natural human-human task-oriented dialogues would probably yield better conditions for entrainment and make much wider contexts available in our dataset. However, in order to avoid data sparsity, we limited the context to a single preceding user utterance, which is likely to have the largest entrainment influence.

To obtain both natural user utterances and natural system responses, we took the following approach: First, user utterances were recorded in calls to a live SDS (see Section 4.1.). The recorded utterances were then transcribed (see Section 4.2.), and the transcriptions were parsed and delexicalized (see Section 4.3.). Finally, based on the meaning of the user utterances, we generated possible response DA (see Section 4.4.) and obtained their natural language paraphrases (see Section 4.5.).

We used the CrowdFlower (CF) platform² to crowdsource call recording, transcription, and response paraphrase creation. To attract native speakers only, the tasks were only made available to CF users in English-speaking countries.

4.1. Recording Calls

Using the Alex English public transport information SDS (Vejman, 2015), we recorded calls in a setting similar to SDS user evaluation (Jurčiček et al., 2011).³ CF users were given tasks that they should attempt to achieve with the system running on a toll-free phone number. The SDS would give them a code that allows them to collect CF reward.

The task descriptions presented to the users were designed so that variable and natural utterances are obtained. Even though the task itself stayed relatively similar,⁴ we varied

the description and used different synonyms (e.g., *schedule/ride/connection*) so that the users are primed with varying expressions. To generate the task descriptions, we used the Alex template NLG system with a specially-designed set of templates where many combinations can be created at random. Furthermore, the users were not aware that the exact wording of their requests is important. According to manual cursory checks of the recordings, they mostly tried to complete the task assigned to them and often kept to wording given to them in the description.

We collected 177 calls comprising 1,636 user utterances. We decided to also include recordings collected previously by Vejman (2015) (347 calls and 2,530 utterances). The response generation step (see Section 4.4.) selected 630 relevant utterances from our calls and 384 utterances from the calls of Vejman (2015).

4.2. Transcription

To ensure that the context user utterances in our dataset are accurate, we had our recorded calls manually transcribed using the standard CF transcription task. A brief description of the domain and lists of frequent words/expressions and subway stations were provided to transcribers to minimize the number of errors.

We collected three transcriptions per utterance and used the transcription variant provided by at least two users, resolving a small number of problematic cases manually.

4.3. Re-parsing

We needed to identify the meaning of the transcribed user utterances in order to generate relevant system response DA (see Section 4.4.). While the recorded calls contain Spoken Language Understanding (SLU) parses of all user utterances, those are based on speech recognition transcriptions. We applied the rule-based Alex SLU system again to manual transcriptions in order to obtain more reliable parses.

To reduce data sparsity, we delexicalized the utterances based on their SLU parses – all stop names as well as time expressions and names of transport vehicles were replaced with placeholders. Identical delexicalized utterances are treated as a single utterance (one context instance) in the dataset, but the frequency information is retained.

4.4. Generating response DA

We have created a simple rule-based bigram policy to generate all possible system response DA.⁵ Based on the given user utterance, it can generate several types of responses:

- a confirmation that the system understands the utterance (DA type `confirm`),
- an answer, providing a transport connection or specific details (DA type `inform`),
- an apology stating that the specified connection cannot be found (DA type `inform_no_match`),
- a request for additional information to complete search (DA type `request`).

The `confirm` response may further be combined with `inform` or `request` in a single utterance. As our policy is

⁵In a real dialogue, the correct response would depend on the whole dialogue history.

²<http://crowdflower.com>

³The task design was adapted from Vejman (2015).

⁴The users were supposed to ask for directions between two stops and request several additional details, such as duration of the ride, or ask for a schedule at a different time.

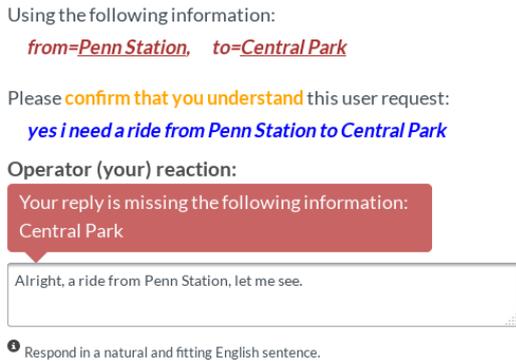


Figure 2: A response task in the CrowdFlower interface

only able to react to our limited domain (see Section 2.), it implicitly filters out all irrelevant user utterances.

4.5. Obtaining response paraphrases for NLG

The generated response DA were then used as the input to a CF task (see Figure 1) where users were asked to create appropriate natural language paraphrases. We designed the CF task interface iteratively based on several trial runs.

The CF user is asked to write a response of a certain kind (corresponding to DA types listed in Section 4.4.) and given information (slots and values) to back it up. The context user utterance is displayed directly above the text entry area to maximize entrainment influence. This simulates a natural situation where a hotline operator hears a request and responds to it immediately. To avoid priming CF users with slot names (e.g., `from_stop`, `departure_time`), we left out slot names where the meaning is unambiguous from the value (e.g., in time expressions) and used very short descriptions (e.g., `from`, `to`) elsewhere.⁶ The task instructions are relatively short and do not include any response examples so that CF users are not influenced by them.⁷

We use a JavaScript checker directly within the CF task to ensure that the paraphrase contains all required information (the exact value for stop names or time, or one of several synonyms in other slots). We also check for presence of irrelevant information, such as stop names, time expressions, or transport vehicles not included in the assignment.⁸ To check the created responses for fluency, we use AJAX calls to our spell-checking server based on Hunspell.⁹

Since about 20% of the responses collected in the testing runs contained errors (irrelevant information or non-fluent responses not discovered by our checks), we performed a manual quality control of all collected responses and requested additional paraphrases on CF where needed. This is quite straightforward and manageable given the size of our dataset; for larger datasets, crowdsourcing could also be used in quality control (Mitchell et al., 2014).

⁶We experimented with using pictographs instead of textual descriptions, but they proved to be rather confusing to CF users.

⁷A testing run with response examples did not bring a better quality of the responses.

⁸In our testing runs, CF users would often fabricate irrelevant information and include it in their responses.

⁹<http://hunspell.github.io>

total response paraphrases	5,577
unique (delex.) context + response DA	1,859
unique (delex.) context	552
unique (delex.) context with min. 2 occurrences	119
unique response DA	83
unique response DA types	6
unique slots	13

Table 1: Dataset size statistics

DA	count
<code>inform_no_match</code>	380
<code>iconfirm</code>	403
<code>iconfirm&inform</code>	23
<code>iconfirm&request</code>	252
<code>inform</code>	549
<code>request</code>	252

Table 2: System response DA counts in the dataset

5. Dataset Properties

The dataset was created over the course of three months, with an estimated net data collection time of one month. The final size statistics are shown in Table 1. There are 1,859 pairs of (delexicalized) context user utterances and system response DA in total, with three natural language paraphrases per pair. The set contains 83 different system response DA, which is lower than similar NLG datasets (see Section 6.), but sufficient to cover our domain. The 552 distinct context utterances provide ample space for entrainment experiments. Based on an estimate measured on a portion of the collected data, around 59% response paraphrases are syntactically aligned to context utterances, around 31% reuse their lexical items, and around 19% show both behaviors (see Figure 3). Statistics of the different DA types used in the dataset are given in Table 2.

The dataset is released in CSV and JSON formats and includes the following for each of the 1,859 items:

- context user utterance
- occurrence count of the user utterance in recorded calls
- SLU parse of the user utterance
- generated system response DA
- 3 natural language paraphrases of the system response

6. Related Work

Other publicly available datasets known to us which are specifically designed for NLG in SDS are those by Mairesse et al. (2010) and Wen et al. (2015b). Both works involve a restaurant information domain, the latter provides an additional set covering hotels. All sets have been obtained using crowdsourcing and contain around 200 distinct system response DA, with ca. 400 paraphrases in the former and around 5,000 in the latter case, which is comparable to our set. None of the sets include context user utterances.

Also related to our work are large-scale datasets of unstructured dialogues (cf. the survey of Serban et al. (2015, p. 21)). They are an order of magnitude larger than our dataset and include up to a full dialogue history, but they contain no semantic annotation, provide no explicit way of controlling the dialogue flow, and are not directly applicable to task-oriented SDS.

context utterance	response DA	response paraphrase
<i>how bout the next ride</i>	<code>inform_no_match(alternative=next)</code>	Sorry, I did not find a later option. I'm sorry, <u>the next ride</u> was not found.
<i>what is the distance of this trip</i>	<code>inform(distance=10.4 miles)</code>	<u>The distance is</u> 10.4 miles. <u>It is</u> around 10.4 miles. The <u>trip</u> covers a <u>distance</u> of 10.4 miles.

Figure 3: Entrainment examples from our dataset (entraining elements marked in color: lexical, syntactic, both).

7. Conclusion

We have presented a novel NLG dataset for the dialogue covering the domain of English public transport information, along with the method to obtain the data using crowd-sourcing. It is, to our knowledge, the first publicly available dataset applicable to experiments with entrainment, or dialogue alignment, in a SDS. The dataset is released under the Creative Commons 4.0 BY-SA license at the following URL:¹⁰

<http://hdl.handle.net/11234/1-1675>

We intend to use the dataset with a fully trainable NLG system in the Alex SDS (Jurčiček et al., 2014) and evaluate perceived naturalness of system responses.

8. Acknowledgments

This work was funded by the Ministry of Education, Youth and Sports of the Czech Republic under the grant agreement LK11221 and core research funding, SVV project 260 224, and GAUK grant 2058214 of Charles University in Prague. 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).

9. Bibliographical References

- Friedberg, H., Litman, D., and Paletz, S. B. (2012). Lexical entrainment and success in student engineering groups. In *Proc. of SLT*, pages 404–409.
- Hu, Z., Halberg, G., Jimenez, C., and Walker, M. (2014). Entrainment in pedestrian direction giving: How many kinds of entrainment. In *Proc. IWSDS*, pages 90–101.
- Jurčiček, F., Keizer, S., Gašić, M., Mairesse, F., Thomson, B., Yu, K., and Young, S. (2011). Real user evaluation of spoken dialogue systems using Amazon Mechanical Turk. In *Proc. of Interspeech*, pages 3068–3071.
- Jurčiček, F., Dušek, O., Plátek, O., and Žilka, L. (2014). Alex: A statistical dialogue systems framework. In *Proc. of Text, Speech and Dialogue*, pages 587–594.
- Levitan, R. (2014). *Acoustic-Prosodic Entrainment in Human-Human and Human-Computer Dialogue*. Ph.D. thesis, Columbia University.
- Lopes, J., Eskenazi, M., and Trancoso, I. (2013). Automated two-way entrainment to improve spoken dialog system performance. In *ICASSP*, pages 8372–8376.
- Lopes, J., Eskenazi, M., and Trancoso, I. (2015). From rule-based to data-driven lexical entrainment models in spoken dialog systems. *Computer Speech & Language*, 31(1):87–112.
- Mairesse, F. and Walker, M. A. (2011). Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*, 37(3):455–488.
- Mairesse, F., Gašić, M., Jurčiček, F., Keizer, S., Thomson, B., Yu, K., and Young, S. (2010). Phrase-based statistical language generation using graphical models and active learning. In *Proc. of ACL*, pages 1552–1561.
- Mitchell, M., Bohus, D., and Kamar, E. (2014). Crowd-sourcing language generation templates for dialogue systems. In *Proc. of INLG and SIGDIAL*, pages 24–32.
- Nenkova, A., Gravano, A., and Hirschberg, J. (2008). High frequency word entrainment in spoken dialogue. In *Proc. of ACL-HLT*, pages 169–172.
- Parent, G. and Eskenazi, M. (2010). Lexical entrainment of real users in the Let’s Go spoken dialog system. In *Proc. of Interspeech*, pages 3018–3021.
- Raux, A., Langner, B., Bohus, D., Black, A. W., and Eskenazi, M. (2005). Let’s go public! taking a spoken dialog system to the real world. In *Proc. of Interspeech*.
- Reitter, D., Keller, F., and Moore, J. D. (2006). Computational modelling of structural priming in dialogue. In *Proc. of NAACL-HLT*, pages 121–124.
- Rudnicky, A. I., Thayer, E. H., Constantinides, P. C., Tchou, C., Shern, R., Lenzo, K. A., Xu, W., and Oh, A. (1999). Creating natural dialogs in the Carnegie Mellon Communicator system. In *Proc. of Eurospeech*.
- Serban, I. V., Lowe, R., Charlin, L., and Pineau, J. (2015). A survey of available corpora for building data-driven dialogue systems. *arXiv:1512.05742*.
- Stoyanchev, S. and Stent, A. (2009). Lexical and syntactic priming and their impact in deployed spoken dialog systems. In *Proc. of NAACL-HLT*, pages 189–192.
- Vejman, M. (2015). *Development of an English public transport information dialogue system*. Master’s Thesis, Charles University in Prague.
- Wen, T.-H., Gasic, M., Kim, D., Mrksic, N., Su, P.-H., Vandyke, D., and Young, S. (2015a). Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *Proc. of SIGDIAL*, pages 275–284.
- Wen, T.-H., Gasic, M., Mrkšić, N., Su, P.-H., Vandyke, D., and Young, S. (2015b). Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proc. EMNLP*, pages 1711–1721.
- Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., and Yu, K. (2010). The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

¹⁰Development continues on GitHub at https://github.com/UFAL-DSG/alex_context_nlg_dataset.