How to Tame Your Data:
Data Augmentation for Dialog State Tracking

Adam Summerville **Jordan Hashemi** **James Ryan** **William Ferguson**
California State Polytechnic University, Pomona Raytheon BBN Technologies
asummerville@cpp.edu jordan.hashemin@rtx.com
james.o.ryan@rtx.com
bill.ferguson@rtx.com

Abstract

Dialog State Tracking (DST) is a problem space in which the effective vocabulary is practically limitless. For example, the domain of possible movie titles or restaurant names is bound only by the limits of language. As such, DST systems often encounter out-of-vocabulary words at inference time that were never encountered during training. To combat this issue, we present a targeted data augmentation process, by which a practitioner observes the types of errors made on held-out evaluation data, and then modifies the training data with additional corpora to increase the vocabulary size at training time. Using this with a RoBERTa-based Transformer architecture, we achieve state-of-the-art results in comparison to systems that only mask trouble slots with special tokens. Additionally, we present a data-representation scheme for seamlessly retargeting DST architectures to new domains.

1 Introduction

Dialog State Tracking (DST) is a common problem for modern task-oriented dialog systems that need to be capable of tracking user requests. Commonly, there is an ontology that defines slots that must be filled according to a user's utterances – e.g., a `restaurant` slot that is filled in with a restaurant name given by the user. A key problem for DSTs is that the values that fill a slot at inference may have never been encountered at training time (consider that the set of all possible restaurant names is bound only by the limits of language).

In this work, we address the problems of training on a domain with effectively limitless possible vocabulary, and aim to create a DST system capable of scaling to unseen vocabulary at inference. We do this by first utilizing a language model (LM) based Transformer that is capable of handling any possible input and output in a textual manner, letting the same exact architecture scale to new intents, slots, and slot values, with no modifications needed. Additionally, we present a practical *data augmentation* procedure for analyzing and addressing issues in the development of a DST system, leading to state-of-the-art performance.

2 Related Work

Work in DST has taken a number of different approaches. The annual DST Challenge (DSTC) has undergone eight iterations (although from the sixth competition on, it has been the more broad Dialog *System Technology* Challenge) (Williams et al., 2013; Henderson et al., 2014a,b). The M2M:Simulated Dialogue (Shah et al., 2018) dataset for dialog state tracking has been addressed by a number of different approaches. Rastogi et al. (2017) used a bi-directional GRU (Chung et al., 2014) along with an oracle delexicalizer to generate a candidate list for slot filling. Rastogi et al. (2018) later used a bi-directional LSTM (Hochreiter and Schmidhuber, 1997) without the oracle delexicalization to generate candidate lists for slot filling. Liu et al. (2018) use two bi-directional LSTMs – one at the utterance level, the other at the dialog level – to perform the dialog state tracking. However, this work is only tested on the simulated dataset Sim-GEN, meaning there is no comparison with the more challenging human crafted utterances contained in Sim-R and Sim-M.

The closest approach to the one detailed in this paper is that of Chao and Lane (2019). They used a system based off of BERT (Devlin et al., 2019), but removed the language-model head and instead used two specialized heads: one that does per-slot utterance level classification to determine whether a given slot is active in the utterance or is the special `dontcare` token, and another per-slot head that predicts whether a token represents the beginning or end of the span for that type of slot. Our

Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pages 32–37
July 9, 2020. ©2020 Association for Computational Linguistics

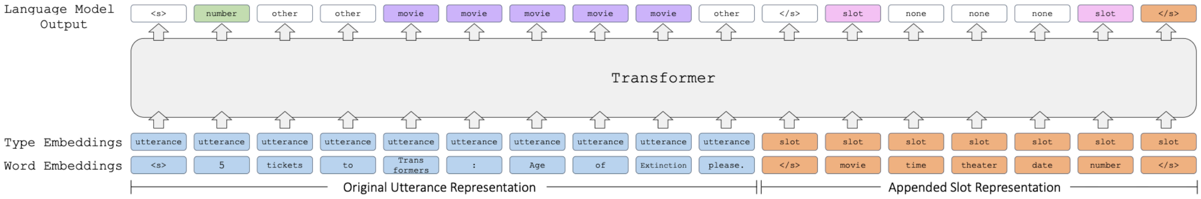

Figure 1: A depiction of the language model based Transformer architecture used in this work. For each token in the user utterance (light blue), the model predicts what slot it belongs to (green or purple), if any, else `other` (white). A token for each of the slots is concatenated to the end of the user utterance (orange) and the model predicts whether that slot is active in the utterance (pink), not active (white), or should be set to the special `dontcare` token (not in this example).

model differs in that we do not need to alter the architecture of the model with specialized heads, and instead fine-tune the existing language model head. In their experimentation, they adjusted the level of slot-specific dropout using targeted feature dropout, first used by Xu and Sarikaya (2014), where slots are replaced with a special `[UNK]` token. Our approach also differs in that instead of simply dropping out slots, we use the more nuanced method of targeted data augmentation.

Finally, data augmentation has been widely used for improving the robustness of dialog systems. Hou et al. (2018) used a LSTM-based sequence-to-sequence network to map from generic utterances (e.g., "show me the `<distance> <poitype>`") to a variety of different utterances (e.g., "where is the `<distance> <poitype>`" and "can you find the `<distance> <poitype>` to me"). This approach requires delexicalization and only alters grammatical structure, which is quite different from our approach which leaves grammatical structure alone, instead altering the non-delexicalized slot values. Quan and Xiong (2019) perform data augmentation via four different approaches: (1) replace words (excluding proper nouns, qualifiers, personal pronouns, and modal verbs) with their synonyms, (2) remove all stop words, (3) use existing neural machine-translation technology to translate from the source language to another and back again (similar to that of Hou et al. (2018), except they do not train their own seq2seq network), and (4) use an existing paraphraser to paraphrase the utterance.

3 Method

Our goal in this work is to to create a robust, readily extensible Dialog State Tracking system that requires minimal to no alteration of network architecture if the schema and/or domain of the dialog task changes. For instance, imagine a system that is being developed for the restaurant domain under a schema in which a set of slots are specified: cuisine, price, location. Now imagine that later it becomes necessary to add a new slot: kid-friendliness. Instead of changing the architecture and retraining from scratch, we would prefer to be able to fine-tune the existing model with the new slot now present. Additionally, we incorporate targeted data augmentation to combat over-fitting when a domain has limited vocabulary.

3.1 Language Model Based Transformer

To produce such a versatile DST system, we reformulate our data such that the problem is fully encoded textually, with no reliance on specialized output heads. Specifically, we carry out:

1. Utterance-level slot activation. Is the slot active in the current utterance? If it is, does the slot map to the special `dontcare` token? That is, for each slot we predict one of `slot`, `none`, or `dontcare`.

2. Token-level slot filling. For each token in the input, is it used in a slot or is it `other`?

To achieve (1), we modify the input utterance with an additional sequence. The additional sequence contains all of the slots present in the dialog schema. For instance, the sentence "5 tickets to *Transformers: Age of Extinction* please." is concatenated with "movie time theater date number". Adding a new slot(s) is handled by simply concatenating to the list – e.g., if the above movie domain was extended to add restaurants "cuisine restaurant location" could be concatenated to the list of slots.

For (2), at the output level a slot is predicted for every token in the original utterance and a slot intent is predicted for every schema token that is concatenated to that utterance: "5[number] tickets to *Transformers:*[movie] *Age*[movie] *of*[movie] *Extinction*[movie] please. <s>movie[slot] time[none] theater[none] date[none] number[slot]' See Figure 1 for a more detailed illustration. Despite the two objectives, the loss is simply the Categorical Cross-Entropy loss over the entire (combined) sequence.

The model aims to track the *joint goal* at each turn in the dialog, represented as all the slot values accumulated to that point. Rather than estimating the entire joint goal each turn, we predict changes to it – additions of slots, modifications to slot values – and maintain the joint goal by applying these changes.

4 Data Augmentation

There are a number of common issues in the datasets for these dialog tasks, including:

1. Small datasets. It is tedious and time-consuming to annotate, gather, or hand-modify believable dialogs.

2. Open classes. Given the open-ended nature of many of these tasks, training data cannot provide coverage of open classes (e.g., restaurant names or movie titles).

To counteract these issues, researchers have proposed a number of different data augmentation schemes (see Section 2). At the outset of our study, we tried the 10% slot-specific dropout used by Chao and Lane (2019), but our model still overfit to the training set. To combat this, we devised the following procedure:

1. **Determine problem slots.** Examine the incorrect predictions on the held-out evaluation set to determine whether there is a certain slot or intent that is not being predicted well.

2. **Augment for problem slots.** Find a corpus of values for that slot, and randomly insert a value from that corpus at training time.

In our work, we were using the Sim-R and Sim-M datasets (Shah et al., 2018), which are concerned with restaurant reservations and movie tickets respectively. We noticed that our system was nearly perfectly able to handle requests related to time, date, and number of people – slots whose values come from small structured sets – but was having difficulty with movie titles, restaurant names, and locations, even with the targeted 10% dropout.

We found corpora for movie names (42,306 movie titles found on Wikipedia as of 2013 (Bamman et al., 2013)), restaurant names (1445 humorous restaurant names (Samuel et al., 2016)), and locations (2067 US settlement names from 1880 to 2010 (Samuel et al., 2016)) which we then used to randomly replace the respective slots at training time at a rate of 50%.

We note that our replacement has two major effects. (1) By randomly replacing with real values instead of simply masking, the model is capable of learning a wider variety of slot values and value structures, instead of simply relying on syntactic information surrounding the names. (2) By randomly replacing values, the dialog becomes more difficult to follow – akin to a user who is prone to changing their mind – and this forces the system to learn to track a user's (fickle) goals better.

5 Experiments

As previously mentioned, we used the Sim-R and Sim-M datasets (Shah et al., 2018). This is because we found them to be of high quality (but with room for improvement), and there was a recent state-of-the-art approach that used a similar Transformer-based architecture to compare against (Chao and Lane, 2019). To assess the performance of the models, we use *joint goal accuracy* (Henderson et al., 2014a), the standard metric for assessing DST systems. At each turn of dialog, the ground truth must be perfectly matched.

For this specific work, we fine-tuned the RoBERTa masked language model of Liu et al. (2019); specifically, we used the Huggingface Transformers library (Wolf et al., 2019). All models were trained with the ADAM optimizer with an initial learning rate of $5e-5$, epsilon of $1e-8$, a linear learning rate schedule over 20 epochs, and an attention mask rate of 15%.

We compare three approaches in the experiment. (1) **RoBERTa-LM**, the RoBERTa LM architecture with 10% slot-specific dropout; (2) **RoBERTa-Separate**, the RoBERTa LM architecture with 50% slot-specific replacement, with separate models trained on the Sim-M and Sim-R datasets; and (3) **RoBERTa-Combined**, the RoBERTa LM architecture with 50% slot-specific replacement, with a single model trained on the combined Sim-M and

DST Model	Sim-M	Sim-R	Sim-M + Sim-R
DST+Oracle	96.8%	94.4%	95.2%
DST+LU	50.4%	87.1%	76.7%
BERT-DST	80.1%	89.6%	86.9%
RoBERTa-LM	71.1%	84.5%	80.8%
RoBERTa-Separate	84.2% *	92.5% *	90.2% *
RoBERTa-Combined	86.5% *	93.1% *	**91.2%** *

Table 1: Comparison of our approaches with prior work. * indicates that the approach is statistically significantly better than BERT-DST (Fisher's exact test with $p < 0.01$).

Sim-R datasets.

5.1 Baselines

To assess our model, we compare against three previous systems. The first work by Rastogi et al. (2017) uses a bi-directional GRU along with an oracle delexicalizer to generate a candidate list for slot filling (**DST+Oracle**). The follow-on work of Rastogi et al. (2018) uses a bi-directional LSTM to build a set of candidates without delexicalization (**DST+LU**). Finally, the most recent approach, by Chao and Lane (2019), builds off of the BERT Transformer architecture which achieved state-of-the-art results (**BERT-DST**).

5.2 Evaluation Results

A summary of the results can be seen in Table 1. We draw attention to the following results. **(1)** The language model based version of RoBERTa without data augmentation performs relatively poorly: it beats the non-Transformer based DST+LU at Sim-M but is worse at Sim-R, and is worse at both than BERT-DST. We did not perform a comprehensive hyperparameter search, so we are unable to discern if it is a critical failing of the model, or whether it was a result of our chosen hyperparameters. **(2)** The RoBERTa language model with data augmentation performed much better than the previous state-of-the-art – with 4.1% and 3.1% point gains respectively on Sim-M and Sim-R. **(3)** Finally, we note that the language model that was trained jointly on both the movie and restaurant data is significantly better than the models trained separately. In part, we believe that this is because the datasets have a lot of overlap – e.g., requesting dates, times, etc. We also believe that due to the relatively small sizes of the datasets, the increase in the size helps combat overfitting in the model – the Sim-M is a smaller dataset than Sim-R (1364 turns vs. 3416) and commensurately, while there is a small gain in Sim-R performance, Sim-M performance is drastically improved (significant at $p < 0.00001$ with Fisher's exact test).

5.3 Discussion

We note that while we have achieved state-of-the-art performance on the Sim-M and Sim-R datasets, there is certainly a possibility that a better choice of augmenting corpora could help the generality of the final model. For instance, the corpus of restaurant names was focused mostly on humorous names, such as “A Brisket a Tasket” and “Et Tu New Brew.” It will take further experimentation to determine if these names are more of a help (the model must be capable of handling a variety of names) or a hindrance (these names are not representative of most restaurant names).

Furthermore, we note the US-centric bias found in the training and evaluation datasets for the location names, and the corresponding bias in our chosen corpus. Similarly, it is an open question as to whether a wider – less US-focused – corpus of location names would help. Certainly, for a system deployed in the world, a wider corpus would likely be of use, but for the purpose of achieving state-of-the-art test accuracy, it is unknown.

6 Conclusions and Future Work

In this paper, we make two contributions. First, we introduce a process for a) examining the source of errors in Dialog State Tracking on held-out evaluation data, and b) correspondingly augmenting the dataset with corpora to vastly increase the vocabulary at training time. Like earlier work that selectively masked slot values, this prevents the system from overfitting to specific values found in the training data. Furthermore, however, it forces the system to learn a wider range of values, rather than syntactic features only, vastly improving the performance. Second, we do this in the context of a language model based Transformer, that due to the language-based nature of its representation – slots are simply represented as tokens concatenated to user utterances – is capable of transferring seamlessly between and working jointly on different datasets without the need to change the underlying architecture. In the future, we would like to address other forms of targeted data augmentation, addressing grammatical differences in addition to vocabulary modifications.

7 Acknowledgements

The information provided in this document is derived from an effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), and awarded to Raytheon BBN Technologies under contract number FA865018-C-7885.

References

David Bamman, Brendan OConnor, and Noah A Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361.

Guan-Lin Chao and Ian Lane. 2019. BERT-DST: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. In *INTERSPEECH*.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014b. The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 324–329. IEEE.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. *arXiv preprint arXiv:1804.06512*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jun Quan and Deyi Xiong. 2019. Effective data augmentation approaches to end-to-end task-oriented dialogue. In *2019 International Conference on Asian Language Processing (IALP 2019)*.

Abhinav Rastogi, Raghav Gupta, and Dilek Hakkani-Tur. 2018. Multi-task learning for joint language understanding and dialogue state tracking. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 376–384, Melbourne, Australia. Association for Computational Linguistics.

Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568. IEEE.

Ben Samuel, James Ryan, Adam J Summerville, Michael Mateas, and Noah Wardrip-Fruin. 2016. Bad news: An experiment in computationally assisted performance. In *International Conference on Interactive Digital Storytelling*, pages 108–120. Springer.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Puyang Xu and Ruhi Sarikaya. 2014. Targeted feature dropout for robust slot filling in natural language understanding. In *Fifteenth Annual Conference of the International Speech Communication Association*.