Knowledge Base Pre-filtering for End-to-End Task-Oriented Dialogue Systems

Mohammad Hanif Dean Nadhif
HKUST
mhdnadhif@connect.ust.hk

Kenneth Lee
HKUST
kleeaj@connect.ust.hk

Abstract

Recent successes in improving the performance of task-oriented dialogue systems have revolved around the introduction of novel architectures to address pre-existing problems in the space. In this paper, we propose a simple yet effective method in achieving performance gains in such systems through a knowledge base pre-filtering process. This is aimed to reduce incorporating unnecessary external knowledge base information. We conduct experiments on a baseline sequence-to-sequence (*seq2seq*) model as the end-to-end dialogue system and show that our method improves the performance of this model. Despite some limitations, our model achieves around a 10% increase in BLEU score on the Multi-Turn In-Car Assistant (KVR) dataset. An alternative pre-filtering method is proposed and shows a 28% improvement in BLEU over the baseline, highlighting the potential for significant performance gains of future improvements in KB prefiltering methods.

1 Introduction

Task-oriented dialogue systems are designed to assist humans in performing specific tasks using natural language as its input. Conventionally, these systems are built as a pipelined process with complex interdependencies between different modules, from language understanding, dialogue management, to language generation. Moreover, to further incorporate external knowledge base information requires lots of human effort.

To address some of these complexities while still maintaining end-to-end trainability, recent end-to-end based approaches have shown positive results (Serban et al., 2015). These approaches utilizes neural encoder-decoder architectures that maps dialogue history to output responses without the need for hand-crafting state labels due to its latent behavior. Some works (Eric and Manning, 2017a; Gülçehre et al., 2016) has extended upon this idea by introducing attention-based copy mechanisms to copy words directly from the input to output responses, which has produced relevant entities despite the appearance of some unknown tokens.

Nevertheless, while the approaches above have been quite successful through altering the model itself, we propose that the overall performance of such models can be further improved with an additional step in the pre-processing stage.

In this work, we experiment with improving the performance of a simple end-to-end task oriented dialogue system, implemented as a vanilla sequence-to-sequence (*seq2seq*) model, by pre-filtering the knowledge base. We test this model on the dialogue corpus that spans three distinct domains in the in-car personal assistant space released by the work of (Eric and Manning, 2017b). Consequently, we find that this approach can improve the performance of this baseline model considerably well, proportional to how much information is filtered out.

2 Methodology

Here, we briefly describe the overall framework applied to filter the knowledge base and in producing the results described in the paper. First, each of the original dataset's dialogues are annotated with a target vector based on the entities representing important knowledge base (KB) entries as given by the original dataset. This is done by retaining the KB entries that contained words presented in the original list of entities. Second, by filtering the KB using the proposed model, we produce a list of indices which indicate the relevant KB information. Lastly, this list is used to produce a filtered dataset that is fed to the seq2seq model and

produce output responses to the dialogue queries. In addition, we compare the results of our model with an alternative pre-filtering method using regular expressions. We describe each component of our model in the subsequent sections.

2.1 Knowledge Base Filter Model

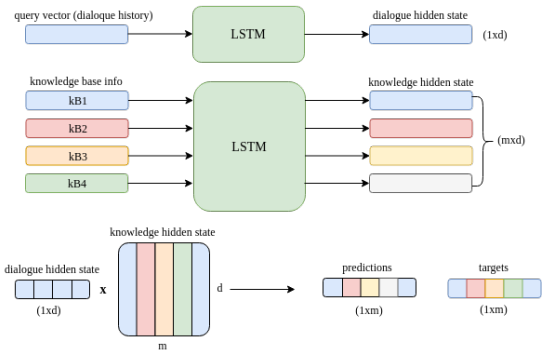

Figure 1: Proposed KB Filtering Architecture

The KB filter model is composed of two long short-term memory (LSTM) networks, one for representing the learned neural representation of the user's dialogue history and another for learning the relevant entries to the knowledge base (KB), as shown in figure 1.

In this in-car assistant (KVR) dataset, a dialogue between the user (*u*) and the in-car assistant (*r*) is composed of a set of utterances, modeled as $\{(u_1, r_1), (u_2, r_2), ..., (u_k, r_k)\}$ where k denotes the number of turns in the dialogue. At the i^{th} dialogue turn, each user utterance (u_i) has a corresponding set of entities (relevant words) $\{(e_1, e_2, ..., e_k)\}$, that correspond to important KB entries modeled as $\{(kb_1, kb_2, ..., kb_k)\}$. The task is then to predict which of these KB entries are relevant based on the dialogue history (query).

This dialogue history is composed of a set of tokens that we first embed via a trained embedding function (ϕ^{emb}) to map each token to a fixed-dimensional vector. This query is then fed into the encoder, implemented as a 1-layer LSTM unit, to produce a hidden representation $(h_1, ..., h_d)$ of this query. Correspondingly, for each dialogue query, each embedded KB entry $(kb_1, ... \ kb_m)$ is fed through another 1-layer LSTM network to produce the hidden representation of the knowledge base. Each of these KB entries are concatenated to form an *mxd* dimensional matrix, where m represents the size of the KB and d denotes the hidden size of the LSTM unit respectively.

Subsequently, taking the dot product between the concatenated KB entries and the dialogue query produces a prediction vector of dimensions *1xm*. Each item in this vector represents how important a single KB entry is with respect to the dialogue (query). During training, this prediction vector is compared to its corresponding target vector produced during the initial annotation process. During inference, the predictions are used to filter out unnecessary KB entries from the dataset to be used in the seq2seq model.

2.2 Sequence-to-Sequence (*seq2seq*) Model

In this work, we use a vanilla sequence-to-sequence (*seq2seq*) model with an encoder-decoder architecture to show that the simplest version of an end-to-end dialogue system can experience an improvement in performance with our proposed pre-filtering method.

The implementation of this model utilizes the seq2seq framework as described in (Sutskever et al., 2014). At the i^{th} dialogue turn, the encoder network takes an aggregated input of the dialogue history and all the KB information for that particular dialogue as its source sequence $\{kb_1, ..., kb_m, u_1, r_1, ... \ u_i\}$. The encoder itself applies a trained embedding function followed by a recurrence function in the form of an LSTM unit (Hochreiter and Schmidhuber, 1997), to encode this source sequence.

The decoder network is essentially an RNN language model conditioned on the input sequence. The decoder network takes in as input the fixed-dimensional representation of the source sequence given by the last hidden state of the encoder network and produces a vector that predicts the response to the user dialogue. During training, this predicted response is compared to the target sequence from the dataset.

3 Experimental Setup

3.1 Dataset

In this paper we used a publicly available multi-turn task-oriented dialogue dataset, namely, the In-Car Assistant (KVR) dataset (Eric and Manning, 2017a). The train/validation/test sets of this dataset is split in advance by the provider. The relevant statistics of this dataset to our research topic can be seen in Table 1.

The KVR dataset is a multi-domain dataset collected using Amazon Mechanical Turk, it

Training Dialogues	2425
Validation Dialogues	302
Test Dialogues	304
Vocabulary Size	1601
Num. of Distinct Entities	284
Num. of Entity(or Slot) Types	15

Table 1: KVR Dataset Statistics

has three distinct domains: calendar scheduling, weather information retrieval, and point-of-interest-navigation. An important feature to note of this dataset is the complexity of the KB information that can span numerous lines. For our project, we took a pre-annotated text file from the Mem2Seq (Madotto et al., 2018) GitHub repository instead of the original JSON file.

As a prerequisite to training our KB filtering model, we removed conversations which has no KB information and conversations that do not have any dialogue turns. Additionally, we annotated the text file with the target KB lines for each dialogue turn. To facilitate the training of our model, we constructed our PyTorch data loader with three of the following elements for each sample: the specific dialogue turn, KB information, and the annotated target entities.

3.2 Training

Our KB filter model is trained using Adam optimizer and a learning rate of 0.01 with binary cross-entropy loss as its loss function. This also means applying a sigmoid layer to the predictions before comparing it with the target labels. Additionally, we used a word embedding size of 200 and a hidden size of 128. We also applied gradient clipping of 10 in our model to prevent gradient explosion. All weights in our model are randomly initialized.

For the sequence-to-sequence model we used the suggested hyperparameters detailed in the Mem2Seq repository with learning rate of 0.001, a four-layer layer LSTM unit with hidden size 128 for both encoder and decoder, dropout of 0.2, and a batch size of 8. Cross entropy loss and Adam optimizer was used to train the model.

3.3 Evaluation Metrics

BLEU: The BLEU metric is usually used to evaluate machine translation models, however, it has also been used in the past to evaluate task oriented dialogue systems. We primarily report this metric in order to evaluate the performance of our model in generating the underlying language patterns seen in our data.

4 Experimental Results

The overall performance achieved by the seq2seq model is dependent on how well the KB Filter Model accurately predicts the relevant entities in the KB. In this work, the baseline performance is given by the seq2seq model's BLEU score on the original KVR dataset. We compare results achieved by the two pre-filtering methods relative to this baseline.

	Original	**Model**	**Regex**
# Scheduling KB Entries	8919	7589	2788
# Weather KB Entries	117808	117754	34934
# Navigation KB Entries	30020	29001	7365
# Overall KB Entries	156747	154344	45087

Table 2: Filtered Train Set Statistics

Table 2 shows the number of knowledge base entries in the training set for the original KVR and pre-filtered datasets. As shown, the model is able to reduce around 1400 entries (14.6%) in the scheduling domain but only about 1000 entries (3.33%) and a mere 50 entries (0.05%) for the navigation and weather domains respectively. This can be attributed to the KB filter model's relatively mediocre accuracy score of 72% when predicting which KB entries to filter out. A discussion on the possible causes of this performance is described in the next section. Meanwhile, we also did a pre-filtering algorithm based on regular expressions which works by combining all the gold entities in each turn of a dialogue and eliminating all KB entries that do not contain any gold entities in the set. This filter eliminates about 111660 entries (71.23%) from the train set while retaining only 31%, 29.6%, and 24.53% from the scheduling, weather, and navigation domains respectively.

Moreover, table 3 shows the BLEU scores of the seq2seq models on the 3 datasets. The results of our experiment clearly shows that pre-filtering the knowledge base produced an improvement in our attained BLEU scores, with the regex filter achieving a score of 8.45 on the test set. This is aligned with the fact that the regex filter removed the most information from the knowledge base. Despite the fact that our KB filter model only removed 1.53% of overall data, it may have removed non-essential information from the scheduling domain and resulted in shorter and more accurate sequences of

information for the seq2seq model to learn from, hence, resulting in an increase in BLEU score.

	BLEU
Baseline (*seq2seq*)	6.60
KB Filtered	7.31
Regex Filtered	8.45

Table 3: Model Performance

5 Discussion

As shown by the results, a reduction in the number of KB entries mean that the encoder network in the seq2seq model encodes shorter sequences of information. Moreover, by placing importance on which particular entries to retain, this narrows down the entries to only more relevant ones. Despite the RNN's inefficiency in encoding long sequences, the pre-filtering process can still improve upon the results of this baseline model. This indicates that for better-performing models like Mem2Seq, the addition of this method can also increase its performance, at least in BLEU score.

That being said, the poor performance of the filter model in removing KB entries from the weather and navigation domains may be attributed to a combination of the following reasons. First, during the annotation step, it was decided that dialogues with no given entities from the KVR dataset should have target vectors of 1s, meaning that the knowledge base entries are retained. Given that there is a significant portion of the KB consisting of the weather domain, where there are a lot of these cases, it is presumed that the model would repeatedly adjust the weights to retain the KB information, introducing a bias. In future works, this could have possibly been avoided by removing such cases where original entities don't exist. The regex filtering method shows a more ideal case where the retention of almost completely relevant information may improve the performance of the baseline by around 28%. The addition of Entity F1 as a metric in future works is important for better evaluation of this experiment.

6 Related Work

Other researchers have implemented an end-to-end task-oriented dialogue systems. By implementing a single RNN-based model, the need for artificial state labels is circumvented. Seq2seq models used in task-oriented dialogue systems exhibit better language modeling performance, however, they do not seem to perform very well in KB retrieval. To mitigate this problem, copy based attention seq2seq based models and Mem2Seq models (Eric and Manning, 2017a; Madotto et al., 2018) have been developed and showed improved performances in this aspect. More recent works such as the Sequicity framework with its two stage copy-net implementation has further improved performance in terms of Entity F1, BLEU, with faster training time (Lei et al., 2018).

7 Conclusion

In this work, we present a method to improve upon the results of end-to-end trainable task-oriented dialogue systems by pre-filtering the knowledge base. We propose a knowledge base filter model which is made up of two LSTM modules to encode the dialogue history and knowledge base entries respectively. Taking the inner product of the hidden representations from these two modules result in a prediction vector that indicates the relevant knowledge base entries to retain. Despite the relatively poor performance of the filter model, it has been shown that the BLEU score of a simple implementation of a task-oriented dialogue system in the form of a seq2seq model have improved due to this pre-filtering. An alternative pre-filtering method using regular expressions is also proposed and serves as a benchmark due to its considerable improvement in performance over the baseline. This implies that better-performing models can achieve even better results when RNN-based limitations have been alleviated. Nevertheless, this paper have shown that the addition of knowledge base pre-filtering can improve the performance of task-oriented dialogue systems.

Acknowledgments

This work is made possible by the help of Andrea Madotto, Chien-Sheng Wu, and HKUST Human Language and Technology Center.

References

Mihail Eric and Christopher D. Manning. 2017a. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. *CoRR*, abs/1701.04024.

Mihail Eric and Christopher D. Manning. 2017b. Key-value retrieval networks for task-oriented dialogue. *CoRR*, abs/1705.05414.

Çaglar Gülçehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. *CoRR*, abs/1603.08148.

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

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *ACL*.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478, Melbourne, Australia. Association for Computational Linguistics.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2015. Hierarchical neural network generative models for movie dialogues. *CoRR*, abs/1507.04808.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.