Modular BERT for Dialogue State Tracking

Osman Ramadan
osmanio.ramadan@gmail.com

Abstract

In this paper, We propose a modular approach that utilises one pre-trained language model as a base model and employs small layers, known as adapter layers, to each specialised sub-task. We also provide an efficient strategy to train these adapter layers, modelling each sub-task as a well-tackled Natural Language Processing task, such as Reading Comprehension, Multiple Choice Questions, ...etc. Adopting this modularisation strategy, using adapter layers, not only leverages the work done in other NLP areas, significantly simplifying the DST task, it also makes the usage of large-scale language models in DST practical for training and inference and also achieves competitive performance to state-of-the-art models that utilise multiple language models. Furthermore, to the best of our knowledge, this the first time the concept of adapter layers is used in DST, opening a new area of research in modularising neural models to tackle tasks with complex data structures or that involve multiple sub-tasks.

1 Introduction

Task-oriented dialogue systems are becoming of increasing importance these days, from virtual assistants that help users accomplish tasks, such as finding flights and booking restaurants, to commercial chatbots that serves as front-line customer services. One of the main components of such systems is the Dialogue State Tracking (DST), which keeps track of the user's intention and goal throughout the dialogue, effectively summarising the dialogue history in terms of belief states. What makes this task specifically difficult is; firstly the complexity of the output/belief state data structure that consists of domains, slots and values. The slots can be informable or requestable slots and their values are either categorical belonging to the well-define discrete set of values or exhibits a free-form such as names and some numerical values. Secondly, DST operates on multiple evolving domains, emphasising the importance of their generalisation ability to unseen data. Data-driven deep learning approaches are getting more popular in modelling DST, mainly because of the large scale and complexity of the domains makes classical machine learning approaches and feature engineering infeasible and hard to scale to new domains. However, since deep learning models are data starving and provide labelling for DST data, with such a complex structure is very expensive. As a result, the Schema-Guided Dialogue (SGD) dataset was introduced, providing a large collections of multi-domain dialogues that models the a domain/service schema as an input, facilitating work on generalisation to unseen domains. Large-scale pre-trained large models such as BERT have shown promising results in SGD and MultiWoz, another large multi-domain dialogues dataset. However, the top performing models combines multiple of these pre-trained models each fine-tuned to a specific DST sub-task making the computational resources requirements to train and run these models impractical to deploy in real-world dialogue systems. We propose a modular approach that utilises one pre-trained language model as a base model and employs small layers, known as adapter layers, to each specialised sub-task. We also provide an efficient strategy to train these adapter layers, modelling each sub-task as a well-tackled Natural Language Processing task, such as Reading Comprehension, Multiple Choice Questions, ...etc. Adopting this modularisation strategy, using adapter layers, not only leverages the work done in other NLP areas, significantly simplifying the DST task, it also makes the usage of large-scale language models in DST practical for training and inference and also achieves competitive performance to state-of-the-art models that utilise multiple language models. Furthermore, to the best of our knowledge, this the

first time the concept of adapter layers is used in DST, opening a new area of research in modularising neural models to tackle tasks with complex data structures or that involve multiple sub-tasks.

2 Related Work

There is plethora of research in applying data-driven deep learning methods in modelling DST. Previous methods that treated the DST as a sequence classification task over the set of all possible values of a slot and individually score all possible slot values, such as NBT. Others have considered feeding the semantic meanings of the slots and values, modelling belief state prediction as a similarity task between the user and system utterances and the corresponding slot and value names. However, all these methods assumed a fixed list of possible values that can be taken by any slot, which is not feasible in reality for some slots, such as names and places. Large pre-trained models such as BERT, which showed promising performances on many down-stream tasks, have also been utilized in DST, such as SUMBT and BERT-DST, by encoding all or part of the dialogue history in addition to slot and values information and have achieved significant improvements in multi-domain dialogues datasets such as MultiWoz. Since the release of the Schema-Guided Dialogue (SGD) dataset in Dialogue State Tracking Challenge 8 (DSTC8), there have been a rising focus on using these large models in a zero-shot settings by encoding the schema or the dialogue ontology that is then fed to the model as an input. The Zero-Shot Dialogue State Tracking model that was offered as a baseline to SGD dataset used a single pre-trained BERT model to encode the system and user utterances and the intents, slots and categorical slot values in each service schema, then applied a projection depending on the sub-task. The encoding obtained by BERT was fixed and during trained only the projection layers were trained. Other proposed models in DSTC8 achieved significant improvements over the baseline, however, they used multiple large pre-trained models. The top performing model fine-tuned 4 pre-trained models (one XLNet and 3 RoBERTa-based models) to solve each sub-task, in addition to several manually engineered features and data augmentation to help with generalisation. Others used 6 BERT-based models with several contextual features that capture information such as which slot is transferred from a different domain or/and has been offered before by the system. In our proposed approach, we only used one BERT model and fed only the current turn user utterance and previous turn system utterance, significantly reducing sequence length of the input and thus the computational overhead that follows. Moreover, we feed in all contextual features, like whether a slot was offered before, as natural text to the model to further exploit the power of the large-scale pre-training these models have undergone.

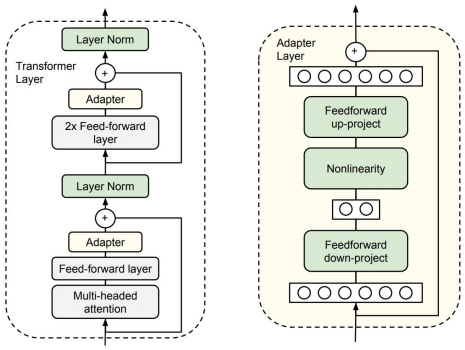

Figure 1: Architecture of the adapter module and its integration with the Transformer. Left: We add the adapter module twice to each Transformer layer: after the projection following multiheaded attention and after the two feed-forward layers. Right: The adapter consists of a bottleneck which contains few parameters relative to the attention and feedforward layers in the original model. The adapter also contains a skip-connection. Figure taken from here

2.1 Adapter Layers

To be able to utilise the power of fine-tuning pre-trained models in each specialised sub-task while still reduce the number of base pre-trained models, we employed the concept of adapter modules proposed in here. Adapter layers are small feed-foward layers with a non-linear activation inserted twice in each transformer layer as show in Figure 1. When trained while keeping the transformer weights fixed they achieve close performance to full fine-tuning big models for each task, while still maintained one base model for all the different tasks.

3 Schema-Guided Dialogue Dataset

The SGD dataset consists of 16,142 dialogues between a human and a virtual assistant that span 16 domains. It is by far the largest Multi-domain goal-oriented dialogues dataset. The dialogue state tracking task in SGD consist of predicting the user's intent for each user utterance, the slot value and

Sub-Task	Modelled as	Input	Output
Slot Classification	Sequence Classification	Previous turn system utterance, current turn user utterance (turn utterances) and services and slots descriptions	1 of 4 classes: `filled`, `dontcare`, `transferred`, `none`
Categorical Slot Prediction	Multiple Choice Question	Turn utterances and a given service and slot descriptions as a question, and the slot values for each choice	1 of N choices, where N is the number of values a slot can take
Free-form Slot Prediction	Question Answering	Turn utterances as the passage and a given service and slot descriptions as a question	Start and end indices of the value string in passage
Transfer Slot Prediction	Multiple Choice Question	Turn utterances and a given service and slot descriptions as a question and the descriptions of all other services and slots that the selected slot value can be transferred from for each choice	1 of N choices, where N is the number of the other services slots
Intent Classification	Multiple Choice Question	Turn utterances and a given service description as a question and intent descriptions for each choice	1 of N choices, where N is the number of intents of the given service
Request Slot prediction	Multiple Choice Question	Turn utterances and a given service description as a question and slot descriptions for each choice	k of N choices, where N is the number of slots of the given service and k are the number of requested slots

Table 1: Our modular design of the DST problem where each sub-task is modelled as standard downstream NLP task and trained on a specialised adapter layer

the slots requested by the user. The slots can either be categorical, taking one of a finite set of possible values or free-form, taking any string value inferred from the dialogue history. The slot values may be from the corresponding user utterance or transferred from the system history actions or user history states. Each domain/service, slot and intention in the schema is provided with a description to facilitate training and evaluating models to handle unseen domain/services given their description.

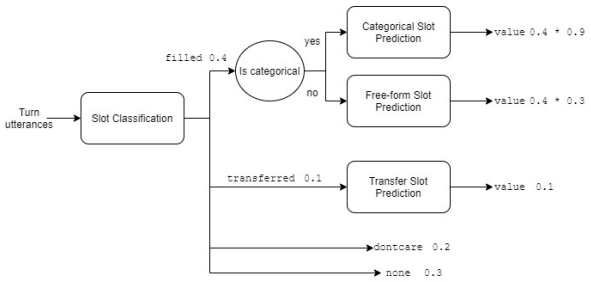

Figure 2: Beam Search strategy for 4 sub-tasks to predict the current turn belief state during inference

4 Modular Dialogue State Tracking

We tackle the problem of DST by decomposing it into several self-isolated sub-tasks and each adapter layer is trained on specific sub-task. From an engineering point of view, this design allows us to model each sub-task to a standard NLP downstream task and employ the best techniques that large pre-trained language models, e.g. BERT, use to solve such task. Table 1 summarises all the sub-tasks, their input and output and what NLP downstream task they are modelled as.

4.1 Belief State Prediction

The belief state task of the SGD dataset consists of predicting the user goal, the active intent and requested slot for each turn. The user goal is defined as the user constraints specified over the dialogue context till the current user utterance. During training, the labels for each sub-task is extracted from the difference between the user goal for the current turn and preceding user turn. In inference time, for each slot in a given service we experimented with two search strategies; firstly, is a greedy approach where we run the Slot Classification module and

take the top prediction, if it is `filled` we run the Categorical Slot Prediction module if the slot is categorical otherwise the Free-form Slot Prediction module. If the prediction `transferred` we use the Transfer Slot Prediction module to get the slot its value is copied to this slot and use this value as the prediction. Finally, predict `dontcare` if this the output of the Slot Classification module or ignore the slot if the output is `none`. The second approach is a beam search over all the 4 possible classes and combine their probabilities with the probability of the module called by the class, then find the value with the highest probability in the beam (Figure 2).

Intent classification and request slot prediction are trained and run as self-contained tasks modelled as shown in Table 1.

4.2 Implementation

For all the sub-tasks listed in Table 1, we use one BERT model and only train a small adapter layer of size 256 for each sub-task. We use Pytorch implementation of BERT-base with pretrained model file `bert-base-uncased` provide by Google[1]. We use AdamW optimizer with a linear schedule warm-up and a learning rate of 6e-5 and 0.006 weight decay. The batch size used is different for each sub-task but lies within 6-32. During inference, we re-insert the module adapter layer in to BERT transformer heads in order to run the corresponding sub-task.

5 Evaluation

We used DST official evaluation metrics provided for SGD dataset, which are:

- **Active intent accuracy**: The fraction of user turns for which the active intent has been correctly predicted.
- **Requested slots F1**: The macro-averaged F1 score for requested slots over the turns. Turns with no requested slots in ground truth and predictions are skipped.
- **Average Goal Accuracy**: This is the average accuracy of predicting the value of a slot correctly, where fuzzy matching is used for scoring non-categorical slots. Only the slots with non-empty assignment in the ground truth dialogue state are considered.

[1]https://github.com/huggingface/transformers

Metric	Baseline	Our Model
Intent Acc.	0.9482	0.9692
Req Slot F1	0.9846	0.9651
Avg. GA	0.5605	0.8036
Joint GA	0.2537	0.4883

Table 2: Our approach performance on the test set of the SGD dataset compared to the baseline

- **Joint Goal Accuracy**: The average accuracy of predicting all slot assignments for a turn correctly. Fuzzy matching is used for non-categorical slots.

6 Results and Discussion

The results in Table 2 shows that our model significantly outperforms SGD baseline model in the average and joint accuracies and slightly improves on the intent accuracy. Even though both models are based on a single Bert model, the use of adapter layers as opposed to pre-trained textual representations for each sub-tasks gives a close performance to fine-tuning multiple Bert models for each these sub-tasks. Moreover, modeling each sub-task independently as one of the standard NLP downstream tasks, e.g. question answering, multiple choice ...etc, simplifies the design of the DST by leveraging the best performing architectures of these downstream tasks. Since each sub-tasks is trained independently, we required one GPU with a relatively small memory, between 6 - 12GB, and thus all experiments were done in Google Colab[2].

7 Conclusion

In this paper, We propose a modular approach that utilises one pre-trained language model as a base model and employs small layers, known as adapter layers, to each specialised sub-task. We also provide an efficient strategy to train these adapter layers, modelling each sub-task as a well-tackled Natural Language Processing task, such as Reading Comprehension, Multiple Choice Questions, ...etc. Adopting this modularisation strategy, using adapter layers, not only leverages the work done in other NLP areas, significantly simplifying the DST task, it also makes the usage of large-scale language models in DST practical for training and inference and also achieves competitive performance to state-of-the-art models that utilise multiple language models. Furthermore, to the best of our knowledge, this the first time the concept of

[2]https://colab.research.google.com/

adapter layers is used in DST, opening a new area of research in modularising neural models to tackle tasks with complex data structures or that involve multiple sub-tasks.

References