XQA-DST: Multi-Domain and Multi-Lingual Dialogue State Tracking

Anonymous ACL submission

Abstract

In a task-oriented dialogue system, Dialogue State Tracking (DST) keeps track of all important information by filling slots with values given through the conversation. Existing methods generally rely on a predefined set of values and struggle to generalise to previously unseen slots in new domains. In this paper, we propose a multi-domain and multi-lingual dialogue state tracker in a neural reading comprehension approach. Our approach fills the slot values using span prediction, where the values are extracted from the dialogue itself. With a novel training strategy and an independent domain classifier, empirical results demonstrate that our model is a domain-scalable and open-vocabulary model that achieves 53.2% Joint Goal Accuracy (JGA) on MultiWOZ 2.1. We show its competitive transferability by zero-shot domain-adaptation experiments on MultiWOZ 2.1 with an average JGA of 31.6% for five domains. In addition, it achieves cross-lingual transfer with state-of-the-art zero-shot results, 64.9% JGA from English to German and 68.6% JGA from English to Italian on WOZ 2.0.

1 Introduction

Task-oriented dialogue systems are designed to provide natural conversation with users and assist them in achieving daily goals. With the growth of task-oriented dialogue systems, there is an increasing interest in supporting dialogues among many domains and languages to fit the users' demands. However, either modelling a multi-domain or multi-lingual dialogue system requires substantial data collected in real scenarios. This data acquisition procedure is extremely expensive, and it motivates us to resolve this challenge by leveraging dialogue data in rich-resource domains and languages via zero-shot transfer learning.

DST is crucial for accurately extracting user intents and goals over multiple turns within the dialogue. Based on the tracked dialogue states, the dialogue manager makes corresponding next actions with back-end results, where the accuracy of the DST becomes absolutely vital. With a fully predefined ontology, traditional approaches tackle the DST as a classification problem by enumerating every possible combination of slot-value pairs (Mrkšić et al., 2017; Zhong et al., 2018). Those approaches are strongly limited by their scalability, as some slots (e.g. *name*) have an unbounded set of slot values. Secondly, they are generally not flexible to unseen slot-value pairs, making them more difficult to adapt for zero-shot transfer learning. Moreover, a completely predefined ontology is hard to acquire and not scalable for task-oriented dialogue systems in real applications.

To overcome those challenges, we take inspiration from Gao et al. (2019) and Gao et al. (2020) and investigate how DST can be tackled by extracting slot values from user utterances directly. In this paper, we propose a domain-independent and transferable dialogue state tracker with neural reading comprehension. Our model is responsible for filling the slot value by recognising specially designed domain-slot prompts by span prediction, which extracts answers from the input utterance by predicting the token positions. In addition, we introduce a novel training strategy for DST in reading comprehension such that we only ask slot questions that appear in the current turn domain. For example, given *hotel* as the current turn domain, all slots under the *taxi* domain are filtered out as there is no overlapping between them. This simple but effective filtering strategy significantly reduces the noise from unnecessary questions in both training and evaluation phases.

We call the final model XQA-DST: **X**LM-R based **D**ialogue **S**tate **T**racker in **Q**uestion **A**nswering. Our main contributions are summarised below:

- We introduce XQA-DST, a novel domain-independent and transferable dialogue state tracker inspired by neural reading comprehension models. The model is able to recognise slot values by reformulating the task as an answer to a specially designed domain-slot prompt by span prediction, which extracts answers from the input utterance by predicting the token positions.

- We enable XQA-DST on reading comprehension by zero-shot domain adaptation scenarios, showing its transferability capabilities. The final model shows competitive domain adaptation performance with an average JGA of 31.6% for five domains on MultiWOZ 2.1.

- We show that our model is capable of both domain adaptation and cross-lingual transfer learning. We demonstrate its cross-lingual transferability by achieving state-of-the-art zero-shot results, 64.9% JGA from English to German and 68.6% JGA from English to Italian on WOZ 2.0.

2 Related Work

Dialogue State Tracking Traditional dialogue state tracking approaches mostly rely on hand-crafted features and domain lexicons for delexicalisation (Wang and Lemon, 2013; Williams, 2014; Henderson et al., 2014), which make them difficult to scale to new domains. With the assumption of a full ontology in advance, classification based approaches tackle DST by enumerating through every possible combination of slot-value pairs (Mrkšić et al., 2017; Liu and Lane, 2017; Ramadan et al., 2018; Zhong et al., 2018). Though a performance improvement is obtained by using a predefined ontology, their scalability is strongly limited by the availability of the ontology, especially for unseen slot values in new domains. The performance on DST is further improved by utilising the pretrained language model BERT (Devlin et al., 2019) as the context encoder. Lee et al. (2019) encode the utterance and slot-value pair separately and implement a slot-utterance matching module that computes the similarity between them. Lai et al. (2020) use BERT to encode the dialogue context concatenated with the candidate pair and generate a relevance score for every candidate. However, both of them rely on a predefined ontology, and none of the approaches has resolved the scalability issue above.

To alleviate this issue, span prediction methods are proposed to tackle DST so that the slot can be filled by directly addressing values in the context. Chao and Lane (2019) propose BERT-DST that encodes the context by BERT and trains independent span projection layers for every slot. Zhou and Small (2019) and Gao et al. (2020) formulate the DST as a question answering problem, and it prepares questions for asking the model to answer the value for every slot. However, Span prediction methods suffer when the value is not explicitly expressed in the context. Heck et al. (2020) remedy this problem by proposing copy mechanisms and achieving competitive results on multi-domain DST. Recent approaches start bringing both the pick-list and span prediction methods into a hybrid architecture. Zhang et al. (2020) split slots into categorical and non-categorical slots. Hence, it benefits from the accuracy brought by the pick-list and the scalability of span prediction methods, but the prediction for categorical slots still relies on a given ontology.

Generative approaches provide an alternative way to handle DST without relying on the predefined ontology. Xu and Hu (2018) construct a pointer network that has an encoder-decoder architecture so that the values of slots can be generated by the decoder. Wu et al. (2019) and Kumar et al. (2020) propose similar sequence-to-sequence models with a state generator that gives a value sequence. However, the main drawback of generative approaches is potentially ill-formatted strings at the output, which can be fatal for the subsequent DST.

Zero/Few-shot Transfer Learning for DST TRADE (Wu et al., 2019) focuses on domain adaptation for DST by transferring prior knowledge of trained domains to an unseen domain. Kumar et al. (2020) propose MA-DST that introduces cross-attention to capture the domain semantics. Campagna et al. (2020) propose a data augmentation approach by synthesising in-domain data from an abstract dialogue model. Li et al. (2021) introduce a generative question answering approach, GPT2-m, that leverages an autoregressive language model. Similarly, Lin et al. (2021) propose the T5DST model that bases on the T5 model (Raffel et al., 2020), and they study the impacts of slot descriptions for domain adaptation.

Cross-lingual transfer learning for DST is to leverage the labelled data in rich-resource languages and transfer learned knowledge to low-resource languages. Chen et al. (2018) study the problem of cross-lingual DST, and propose the

XL-NBT teacher-student framework. Liu et al. (2020) introduce an Attention-informed Mixed-Language Training (AMLT) method that uses bilingual word pairs to build code-switching training sentences. Moreover, they study the effectiveness of multi-lingual pretrained language models with their AMLT approach, including XLM (Conneau and Lample, 2019) and mBERT (Devlin et al., 2019). Qin et al. (2020) further propose a data augmentation framework, which encourages cross-lingual alignment by fine-tuning mBERT on generated code-switching data. To the best of our knowledge, we are the first work that studies the effectiveness of a multi-lingual pretrained language model, XLM-R (Conneau et al., 2020), on DST without implementing additional cross-lingual alignment strategies.

3 Multi-Domain and Multi-Lingual DST

To tackle the task of dialogue state tracking, our model reads the current user utterance U_t, preceding system utterance M_t, dialogue history H_t, and the domain-slot prompt Q_t as inputs for each turn. Followed by that, our model is responsible for firstly determining the dialogue domains D_t from the input sequence. Then, it predicts the class of answers for domain-slot prompts in the predicted domains. If an answer is present in utterances, the model will predict the value for that domain-slot question using span extraction. Otherwise, its value will be predicted in accordance with the predicted class. Finally, our model tracks the dialogue states by a rule-based update mechanism along with the progress of the dialogue across turns.

3.1 Context and Domain-slot Questions

In neural reading comprehension, the context is used to provide the background information, and the answer is usually contained in the context. When it comes to DST, it is equivalent to model the system message and the user response together as the context for the current turn. The complete context C_t is then collected by concatenating the current user utterance U_t and the preceding system utterance M_t with dialogue history H_t at turn t. We implement XLM-R as the context encoder for the purpose of cross-lingual transfer learning.

Each context is paired with N questions, which iterate through every slot that we are interested in. We append the domain-slot prompt at the end of the context as an analogue question for each domain-slot pair. Hence, the model can learn to correlate different questions to the same context and provide corresponding answers to fill the slot values. For the same context with nth question Q_t^n at turn t, the input sequence S_t^n can be written as:

$$\begin{aligned} S_t^n = & [\text{CLS}] \oplus U_t \oplus [\text{SEP}] \oplus M_t \oplus [\text{SEP}] \\ & \oplus H_t \oplus [\text{SEP}] \oplus Q_t^n \oplus [\text{SEP}] \end{aligned} \tag{1}$$

where H_t represents the dialogue history that is collected in a reversed order from turn $t-1$ to $t=1$, and it is defined as follows:

$$\begin{aligned} H_t = & U_{t-1} \oplus M_{t-1} \oplus \dots \\ & \oplus U_1 \oplus M_1 \text{ for } t > 1 \end{aligned} \tag{2}$$

To utilise the question as a distinct feature for each slot, we propose the analogue question in the format of a domain-slot prompt. Here, additional special tokens are introduced to assist the model in recognising the domain-slot pair as distinct parts. Moreover, they provide clear signals for the start and end positions for each domain-slot pair. The equation for constructing the domain-slot prompt Q_t^n is defined below:

$$\begin{aligned} Q_t^n = & \langle \text{dom.} \rangle \oplus d_t^n \oplus \langle /\text{dom.} \rangle \\ & \oplus \langle \text{slot} \rangle \oplus s_t^n \oplus \langle /\text{slot} \rangle \end{aligned} \tag{3}$$

where d_t^n refers to the name of the domain and s_t^n is the slot for n-th question at turn t.

3.2 Shared Classification Gate

Our model contains a shared classification gate θ_{gate} for every domain-slot question. This shared gate provides shared knowledge among various domain-slot pairs, as it is neither domain-specific nor slot-specific.

For each input sentence S_t, this shared gate classifies it to one of six classes as described in three main categories. Special cases, *none/dontcare*, indicate that there is either no observable value from the input sequence S_t or any value that can become the answer for that slot question. Copy mechanism, *span*, indicates that the answer can be extracted from the current user utterance U_t by the span prediction module. Similarly, *Inform* is to copy from the system inform memory that tracks values mentioned in the preceding system utterance M_t. Boolean values *true/false* are used to deal with binary categorical values for Boolean slots where the value cannot be directly extracted from the input utterance.

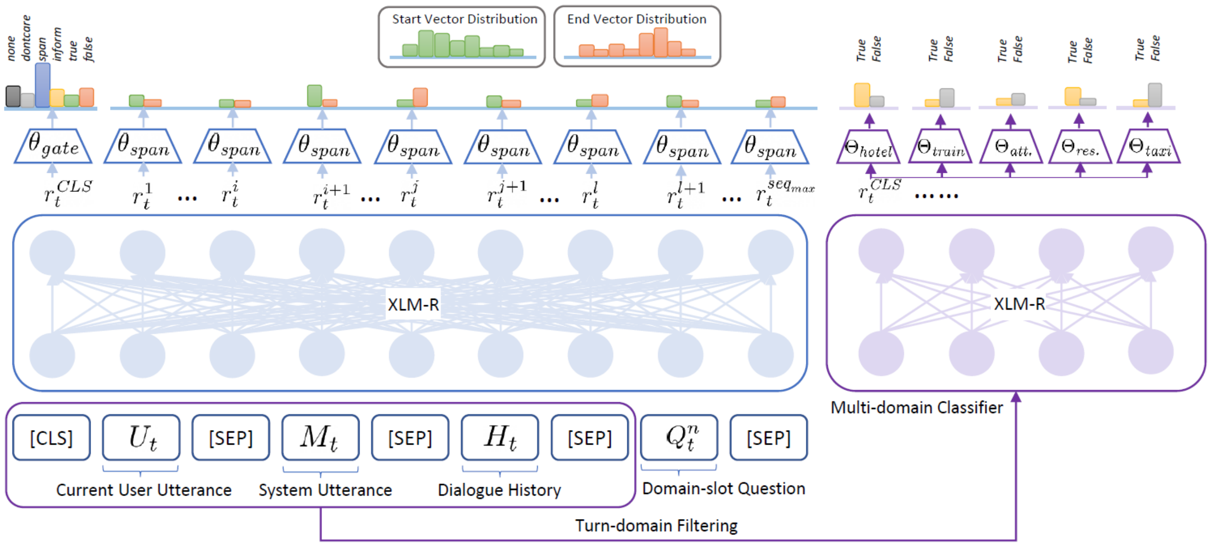

Figure 1: The model architecture of our XQA-DST for multi-domain and multi-lingual DST, where the right part is the independent multi-domain classifier that outputs active domains.

With these designed classes, it takes the pooled output r_t^{CLS} from the encoder as its only input. It generates a probability distribution $p_t^{\text{gate}} \in \mathbb{R}^6$ over six classes as in the following equation:

$$p_t^{\text{gate}} = \text{softmax}(W_{\text{gate}} \cdot r_t^{\text{CLS}} + b_{\text{gate}}) \qquad (4)$$

where W_{gate} represents the weights for our shared gate that is achieved by a linear classification layer, and b_{gate} is the corresponding bias term. The class is then determined by taking the maximal argument of $\text{argmax}(p_t^{\text{gate}})$.

3.3 Shared Span Prediction Layer

If the predicted class for the current input sequence S_t is *span*, the answer for that domain-slot question Q_t will be filled by predicting the start and end positions of the value from the input sequence. We implement a shared span prediction layer for every domain-slot question for the purpose of domain-adaptable design. This is achieved by constructing a linear layer that takes the entire token representations from r_t^1 to $r_t^{seq_{max}}$ as inputs, and it generates two outputs for each token, the start and end position distribution, p_t^{start} and p_t^{end}, after the softmax layers.

$$[p_t^{\text{start}}, p_t^{\text{end}}] = \text{softmax}(W_{\text{span}} \cdot r_t^i + b_{\text{span}}) \qquad (5a)$$

$$\text{start}_t = \text{argmax}(p_t^{\text{start}}) \qquad (5b)$$

$$\text{end}_t = \text{argmax}(p_t^{\text{end}}) \qquad (5c)$$

The start and end positions of the predicted value are then determined by picking the largest probability from distributions p_t^{start} and p_t^{end}. Followed by that, we sequentially collect the tokens from the predicted start_t position to end_t position and detokenize them to form the final predicted value for that domain-slot question.

3.4 Turn-domain Filtering

For a task-oriented dialogue, the user may shift the domain of conversation across turns so that a dialogue can have multiple domains. We introduce a turn-domain filtering strategy that puts a strict constraint and only allows the model to pay attention to the current domain. Turn-domain filtering indicates that only the slots within the current domains D_t are used to prepare training features since slots are domain-specific. Hence, turn-domain filtering can reduce the potential noises introduced by unnecessary domains. Mathematically, this filtering strategy puts an additional constraint for slot domain d_t^n in Eq. 3:

$$d_t^n \in D_t \qquad (6)$$

3.5 Independent Multi-domain Classifier

Turn-domain filtering allows the model to answer questions only within the interested domains. However, the domain information is no longer a given feature in the evaluation stage. Here, we propose a multi-domain sequence classifier as shown in Fig. 1. The input sequence is the complete dialogue context C_t without domain-slot questions. We then collect the entire sequence representation r_t^{CLS} by the context encoders as XLM-R(C_t). Followed by that, r_t^{CLS} is fed into $|D|$ softmax layers, thereby allowing a binary prediction that decides whether each domain d_t is present in the input context or not. Finally, we collect the domains that have been assigned to the '*True*' class, which indicates the presence of that domain in the context.

$$p_t^d = \text{softmax}(W_{\text{MSC}}^d \cdot r_t^{\text{CLS}} + b_{\text{MSC}}^d) \tag{7a}$$

$$d_t = \text{argmax}(p_t^d) \tag{7b}$$

$$D_t = \{d_1, \ldots, d_{|D|}\} \tag{7c}$$

Though this domain classifier is not domain scalable, it is extremely effective when the range of domains is given so that we can have fixed weights for each domain projection layer.

3.6 System Inform Memory and Update Rules

To further reduce the error of our span extractor, we have employed the same inform copy mechanism as Heck et al. (2020). This memory is a simple dictionary that records all values informed by the preceding system utterance M_t into a system inform memory I_t=$\{I_t^1$, ..., $I_t^N\}$. Then, the value answer A_t^n for nth question Q_t^n asked at turn t can be predicted by the following copy mechanism, given that *inform* $= \text{argmax}(p_t^{\text{gate}})$:

$$A_t^n = I_t^n \text{ for } Q_t^n \tag{8}$$

We implement a simple rule-based mechanism that is used to update dialogue states across turns as same as Chao and Lane (2019). In each turn, if the model assigned class for the current input sequence S_t^n with Q_t^n is not *none*, the dialogue state will be updated by obtaining A_t^n from our value prediction modules. On the other hand, if the classification gate predicts that there is no value for S_t^n, the dialogue state will be kept unchanged.

4 Experimental Setup

4.1 Dataset

The datasets that we carry out experiments on are WOZ 2.0 (Wen et al., 2017) and MultiWOZ 2.1 (Eric et al., 2020) for single-domain and multi-domain task-oriented dialogues, respectively. WOZ 2.0 is a restaurant reservation dataset and it contains three slots: *area*, *food*, and *price range*. Moreover, it provides the conversation in three languages: English, German, and Italian, so that we can carry out cross-lingual transfer learning experiments on this dataset. By contrast, MultiWOZ 2.1 contains multi-domain conversations for more than 10000 dialogues over seven domains. Moreover, the dialogue domain can change across turns, thereby making MultiWOZ 2.1 the most challenging dataset for task-oriented dialogue systems. There are two domains, *hospital* and *police*, that only appear in the train set but not in the validation and test sets. Hence, we exclude these domains with very few dialogues, and the remaining dataset contains five domains (*hotel*, *train*, *attraction*, *restaurant*, and *taxi*) with 30 slots in total.

4.2 Implementation Details

We employ the pretrained *XLM-RoBERTa-base* model from the Huggingface library of Transformers (Wolf et al., 2020), which consists of 12 hidden layers of 768 units. For all implementations, we limit the maximal input sequence length to be 180 tokens for saving the cost while keeping a reasonable length for including dialogue history. We truncate from the earliest dialogue history when the input sequence length exceeds the limit. The training objective is to minimise the summations of individual loss functions for each module, where each loss is defined as the cross-entropy loss. The coefficients for each part of the joint loss of our question answering model are:

$$\mathcal{L}_{total} = 0.8 \cdot \mathcal{L}_{gate} + 0.2 \cdot \mathcal{L}_{span} \tag{9}$$

During the training process, we implement the Adam optimiser (Kingma and Ba, 2015) with an initial learning rate of 10^{-5}, where the other parameters for Adam are within their default settings. Then, we employ a linear scheduler with a warm-up proportion of 10% so that the learning rate will decay linearly until reaching zero after the warm-up steps. We put a dropout layer with a rate of 30% at the output of our context encoders. We use an early stopping strategy by monitoring the accuracy of the validation dataset until it stops increasing for at least 3 epochs. The batch size is fixed at 16 for XLM-R. The multi-domain classifier is trained independently with the same experimental setting, and it is only involved in the evaluation stage. We report the mean of supervised DST and cross-lingual experimental results for three runs with different random seeds.

5 Experimental Results

5.1 Supervised DST

We first rank our XQA-DST model with prior methods capable of zero-shot domain adaptation on MultiWOZ 2.1. Table 1 comprises the JGA for each method, where the JGA is defined as the ratio of dialogue turns that have been perfectly predicted over the number of turns for all dialogues. We implement the same label mapping as TripPy (Heck et al., 2020) for a fair evaluation. In Table 1, our approach has outperformed all prior methods capable of zero-shot generalisation, including most generative approaches such as TRADE, T5DST, and GPT2-m. Moreover, our XQA-DST model is competitive with state-of-the-art approaches that only focus on supervised DST. It is worth noting that SOM-DST (Kim et al., 2020) and SimpleTOD (Hosseini-Asl et al., 2020) are generative approaches, but they are not designed with domain-slot prompts, which make them not naturally domain adaptable. SST (Chen et al., 2020) relies on a predefined schema to learn slot relations. Since candidate values are given, it gives a slightly higher JGA than our approach, but it is neither domain-adaptable nor open-vocabulary. Lastly, TripPy is not domain scalable because it has trained N projection layers for N given slots, which makes it completely have no knowledge for new slots in new domains.

Based on the shared span prediction module, our model is able to extract values from the dialogue context directly, thereby being open-vocabulary and domain scalable. At the same time, it has successfully overcome the challenge of an unavailable ontology set. Moreover, our model presents as the best-performed model in any framework with span prediction modules, where it has improved the margin of JGA by more than 3.5% from the STARC approach. None of the other approaches has ever studied their DST with multi-lingual pretrained models. By utilising the pretrained XLM-R model as the context encoder, our approach is the only method with cross-lingual transferability. Given its distinct advantages for being domain-adaptable and language transferable, a promising result in multi-domain DST at 53.2% builds a good foundation for zero-shot domain adaptation and cross-lingual experiments.

5.2 Zero-shot Domain Adaptation

The zero-shot domain adaptation experiment is used to evaluate the transfer performance of our model when it is tested with dialogues in a completely unseen domain. We train our model on the other four domains by excluding the target domains. We strictly follow the experimental steps reported by Kumar et al. (2020). Since there is a single domain defined in the target domain, the domain classifier is not utilised here because the dialogue domain is given information. Table 3 shows a comparison of our XQA-DST model to baselines and recent approaches. It is clear that our model has generated more accurate results than both MA-DST (Kumar et al., 2020) and SUMBT (Lee et al., 2019) baselines by at least 3.4% JGA on average in domain adaptation. SUMBT tracks the dialogue states by classifying through every slot-value pair. Hence, it is a classification based method, whereas our approach is mainly relying on the value filling by the span prediction module. It can be seen that our model has outperformed baselines by a significant (3-9%) margin on the *hotel*, *restaurant*, and *taxi* domains. This is because the classification based method requires a predefined ontology for its enumeration of values, which inevitably makes it not robust to unseen values in new domains and results in relatively low performance for domain adaptation.

There is another class of methods that utilises generative value filling to handle the DST, including TRADE, GPT2-m, and T5DST. Given GPT2-m as an example, it is in the framework of generative

Models tested on MultiWOZ 2.1	**JGA (%)**
TRADE (Wu et al., 2019)	45.60
SUBMT (Lee et al., 2019)	46.70
STARC (Gao et al., 2020)	49.48
MA-DST (Kumar et al., 2020)	51.88
T5DST (Lin et al., 2021)	52.21
GPT2-m (Li et al., 2021)	52.58
XQA-DST	**53.21**

Table 1: The performance of DST for our proposed XQA-DST model with prior methods capable of zero-shot inference on MultiWOZ 2.1.

Models tested on MultiWOZ 2.1	**JGA (%)**
DSTQA (Zhou and Small, 2019)	51.17
DS-DST (Zhang et al., 2020)	51.21
XQA-DST	**53.21**
SOM-DST (Kim et al., 2020)	53.68
SST (Chen et al., 2020)	55.23
TripPy (Heck et al., 2020)	55.30
SimpleTOD (Hosseini-Asl et al., 2020)	55.72

Table 2: The performance of DST for our XQA-DST model against state-of-the-art DST incapable of zero-shot inference on MultiWOZ 2.1.

Models	Type	Hotel	Train	Att.	Res.	Taxi	Avg.
MA-DST (Kumar et al., 2020)	G	16.3	22.8	22.5	13.6	59.3	26.9
SUMBT (Lee et al., 2019)	C	19.8	22.5	22.6	16.5	59.5	28.2
TRADE (Wu et al., 2019)	G	19.5	22.9	22.8	16.4	59.2	28.2
SimpleTOD++* (Lin et al., 2021)	G	17.7	27.8	28.0	15.6	59.2	29.7
XQA-DST	**S**	**22.9**	**23.2**	**24.0**	**25.7**	**62.2**	**31.6**
GPT2-m (Li et al., 2021)	G	24.4	29.1	31.3	26.2	59.6	34.1
T5DST* (Lin et al., 2021)	G	21.2	35.4	33.1	21.7	64.6	35.2

Table 3: The joint goal accuracy (%) of zero-shot domain adaptation experiments on each domain with recent models on MultiWOZ 2.1. The abbreviations for model types are: G: Generative; C: Classification; S: Span prediction. *Results from MultiWOZ 2.0 are reported by (Lin et al., 2021).

question answering, which also coincides with the underlying idea of our XQA-DST model but has a decoder to generate candidate values. It provides higher accuracy than our approach for about 6% improvement of JGA on *train* and *attraction* domains. Then, it leads to a higher average JGA at 34.1%, which is 2.5% higher than our approach. However, our model still achieves higher average JGA than MA-DST, TRADE, and SimpleTOD++ (Lin et al., 2021), which are also generative approaches.

Although our approach is less competitive to state-of-the-art generative approaches in domain adaptation, our model has outperformed both GPT2-m and T5DST in multi-domain supervised DST as shown in Table 1. Furthermore, our approach is designed to be applicable for both domain adaptation and cross-lingual transfer learning, whereas all generative methods listed above can only do mono-lingual learning. Therefore, our XQA-DST model has shown very competitive results in the zero-shot domain adaptation, and we can conclude that it is able to effectively generalise to task-oriented dialogues in new domains by understanding the linguistics behind our domain-slot questions.

5.3 Error analysis

We analyse the individual slot accuracy for every domain-slot pair in 5 domains to study the impact of shared slots over domains on the performance of domain adaptation. The results are obtained by computing the slot accuracy on each target domain by XQA-DST. The slot accuracy is defined as the ratio of dialogue turns where the value for that slot is correctly predicted. Fig. 2 shows the slot accuracy for 16 slots over 5 domains, where multiple domain bars for the same slot indicate that the slot is shared across these domains.

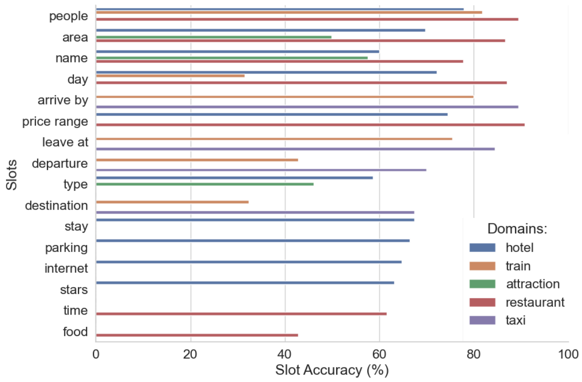

Figure 2: The categorical plot of slot accuracy (%) for each slot over 5 domains for the zero-shot domain adaptation experiment by XQA-DST.

It is observable that the slots that have been shared among multiple domains lead to a relatively higher domain adaptation performance. By contrast, it is also distinctive that slots that have not been shared among multiple domains have much lower accuracy. For instance, most slots in the *hotel* domain are not shared with other domains, so the slot accuracy for '*internet*' and '*stars*' slots (64.7% and 63.1%, respectively) are reasonably lower than others. The same rule applies to the '*time*' and '*food*' slots in the *restaurant* domain. Therefore, the number of shared domains for the slot is the foremost factor for achieving a good domain adaptation result on that slot. Secondly, we notice that slots with digital values such as '*people*' and '*day*' have very high slot accuracy (89.4% and 87.0% in the *restaurant* domain) even in the zero-shot setting. It validates the effectiveness of our model to domain adaptation for successfully extracting candidate values from the message. Last but not least, it is naturally hard to predict location slots, '*departure*' and '*destination*', that are not categorical with unseen values. Hence, even though they are shared in both *train* and *taxi* domains, they give the lowest slot accuracy in the set of shared slots. Overall speaking, our XQA-DST model has generated reasonably well domain adaptation results on most domain-slot pairs and has shown a certain level of common knowledge across domains.

5.4 Zero-shot Cross-lingual Transfer Learning

The zero-shot cross-lingual transfer learning is to train our XQA-DST on the source language, English. Then, it is sequentially evaluated on the test sets in German and Italian with labels that are kept in English. Since WOZ 2.0 is a single domain dataset with relatively short dialogues, the dialogue history is not included as inputs, and the domain classifier is deactivated. To provide a fair comparison to the ground truth, we implement Google Translator (Wu et al., 2016) to translate the values filled by span prediction in the target language back to the source language.

In Table 4, our XQA-DST model gives strong a zero-shot performance on both German and Italian languages (64.9% and 68.6% JGA, respectively). In comparison to recent approaches on zero-shot cross-lingual DST, our XQA-DST model has generated results that significantly increase the margin by an absolute 7% on Italian. It is worth noting that both XLM+CLCSA and mBERT+CLCSA (Qin et al., 2020) are data augmentation based approaches on multi-lingual models with the same model architecture as XL-NBT (Chen et al., 2018). Even without any data augmentation, our model in neural reading comprehension still outperforms all of them and appears as the state-of-the-art results in the zero-shot cross-lingual transfer learning on WOZ 2.0.

Besides the above approaches, we include XLM-R-DST as a baseline that we replace the context encoder of BERT-DST (Lai et al., 2020) with XLM-R. Then, we can study the effectiveness of different model architectures in cross-lingual transfer learning. We recall that XLM-R-DST fills the slot values by iterating through every candidate slot value with a relevance scorer. Table 4 shows a huge improvement of our approach by increasing the average JGA on target domains from 23.1% to 66.8% by more than 40%. It indicates that our specially designed reading comprehension framework has a strong generalisation ability across languages, whereas the XLM-R-DST appears as only recognising each value as distinct features without understanding the deep semantics behind them.

Models	**Joint Goal Accuracy (%)**		
	EN	**GE**	**IT**
XLM-R-DST	88.46	20.78	25.39
XL-NBT	-	30.80	41.20
MUSE + AMLT	-	36.51	39.35
XLM+CLCSA	-	48.70	-
mBERT+CLCSA	-	63.20	61.30
XQA-DST	92.38	**64.88**	**68.63**

Table 4: The zero-shot cross-lingual results for target languages, German (GE) and Italian (IT), on WOZ 2.0, where the results on English (EN) are only used to indicate the supervised performance on the source language. There are no results on Italian by XLM due to the absence of Italian in its pretraining as reported by (Liu et al., 2020).

Lastly, we notice that the cross-lingual result on Italian has a slightly higher joint goal accuracy than German in our experiments. We suppose that this is because of the declension in German, which leads to more diverse word forms with the same semantics. Since our cross-lingual experiment relies on a back-translation from the target language to the source language, a diverse declension still introduces noises to the translation process. Even with the predefined label dictionary that collects vocabulary with similar semantics, it cannot perfectly handle a more flexible word list.

6 Conclusion

We introduce a new multi-domain and multi-lingual dialogue state tracker, XQA-DST, within a neural reading comprehension framework. It gives distinct advantages for avoiding relying on any predefined ontology and being open-vocabulary to new slots with unseen values. We have demonstrated its competitive performance in multi-domain DST with a novel turn domain filtering strategy and a multi-domain classifier in parallel. We have shown a strong domain and cross-lingual transferable ability of our model by outperforming famous baselines. With the design of an XLM-R based multi-domain classifier, our approach is feasible for tracking states in multi-domain and multi-lingual scenarios. Therefore, it holds a strong potential to overcome the challenging data scarcity problem for either domains or languages in the real application of task-oriented dialogue systems.

References

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics.

Guan-Lin Chao and Ian R. Lane. 2019. BERT-DST: scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. In *INTERSPEECH*, pages 1468–1472. ISCA.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7521–7528.

Wenhu Chen, Jianshu Chen, Yu Su, Xin Wang, Dong Yu, Xifeng Yan, and William Yang Wang. 2018. XL-NBT: A cross-lingual neural belief tracking framework. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 414–424, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *NeurIPS*, pages 7057–7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. 2020. From machine reading comprehension to dialogue state tracking: Bridging the gap. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 79–89, Online. Association for Computational Linguistics.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273, Stockholm, Sweden. Association for Computational Linguistics.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *NeurIPS*.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Adarsh Kumar, Peter Ku, Anuj Kumar Goyal, Angeliki Metallinou, and Dilek Hakkani-Tür. 2020. MA-DST: multi-attention-based scalable dialog state tracking. In *AAAI*, pages 8107–8114. AAAI Press.

Tuan Manh Lai, Quan Hung Tran, Trung Bui, and Daisuke Kihara. 2020. A simple but effective bert model for dialog state tracking on resource-limited systems. In *ICASSP*, pages 8034–8038. IEEE.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: slot-utterance matching for universal and scalable belief tracking. In *ACL (1)*, pages 5478–5483. Association for Computational Linguistics.

Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. 2021. Zero-shot generalization in dialog state tracking through generative question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1063–1074, Online. Association for Computational Linguistics.

Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021. Leveraging slot descriptions for zero-shot cross-domain dialogue StateTracking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5640–5648, Online. Association for Computational Linguistics.

Bing Liu and Ian R. Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. In *INTERSPEECH*, pages 2506–2510. ISCA.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. In *AAAI*, pages 8433–8440. AAAI Press.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. In *IJCAI*, pages 3853–3860. ijcai.org.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437, Melbourne, Australia. Association for Computational Linguistics.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432, Metz, France. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Jason D. Williams. 2014. Web-style ranking and SLU combination for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 282–291, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1457, Melbourne, Australia. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, Philip Yu, Richard Socher, and Caiming Xiong. 2020. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 154–167, Barcelona, Spain (Online). Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, Melbourne, Australia. Association for Computational Linguistics.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *CoRR*, abs/1911.06192.