Abstract

Slot-filling is an essential component for building task-oriented dialog systems. In this work, we focus on the zero-shot slot-filling problem, where the model needs to predict slots and their values, given utterances from new domains without training on the target domain. Prior methods directly encode slot descriptions to generalize to unseen slot types. However, raw slot descriptions are often ambiguous and do not encode enough semantic information, limiting the models’ zero-shot capability. To address this problem, we introduce QA-driven slot filling (QASF), which extracts slot-filler spans from utterances with a span-based QA model. We use a linguistically motivated questioning strategy to turn descriptions into questions, allowing the model to generalize to unseen slot types. Moreover, our QASF model can benefit from weak supervision signals from QA pairs synthetically generated from unlabeled conversations. Our full system substantially outperforms baselines by over 5% on the SNIPS benchmark.

1 Introduction

Automatic slot filling, which extracts task-specific slot fillers (e.g. flight date, cuisine) from user utterances, is an essential component to spoken language understanding (Bapna et al., 2017). As shown in Figure 1, the model predicts the slot filler “Joe A. Pass” for the slot type “artist” given an input utterance. However, fully supervised slot filling models (Young, 2002; Goo et al., 2018) require labeled training data for each type of slot (Shah et al., 2019). It is even more of a problem for data-intensive models (Mesnil et al., 2014). This makes the development of new domains in these systems a challenging and resource-intensive task.

This has motivated studies in cross-domain zero-shot slot learning for the slot-filling task (ZSSF), where the goal is to achieve good slot-filling performance on new domains without requiring additional training data. Previous work (Bapna et al. (2017); Shah et al. (2019)) often uses a sequence tagging approach (similar to the upper image in Figure 1 in a high-level way). To achieve zero-shot domain transfer, they directly encode raw slot descriptions or names, such as “playlist”, “music item”, to enable models to generalize to slot types unseen at training time. However, slot descriptions are often ambiguous and typically do not encode enough semantic information by themselves.

Instead of directly encoding slot descriptions and examples, we introduce a QA-driven slot filling framework (QASF) (Figure 1). Inspired by the recent success of QA-driven approaches (McCann et al., 2018; Logeswaran et al., 2019; Gao et al., 2019; Li et al., 2020; Namazifar et al., 2020), we tackle the slot-filling problem as a reading comprehension task, where each slot type (e.g. “artist”) is associated with a natural language question (e.g. “Who is the artist to play?”). A span-based reading comprehension model is then used to extract a slot filler span from the utterance by answering the question.1 In this work, we use a linguisti-

---

1 It can be seen as an extension of the QA-driven meaning representations (He et al., 2015; Michael et al., 2017), where
cally motivated question generation strategy for converting slot descriptions and example values into natural questions, followed by a BERT-based QA model for extracting slot fillers by answering questions. As shown in our experiments, this QA-driven method is better at exploiting the semantic information encoded in the questions, therefore it generalizes better to new domains without any additional fine-tuning, as long as the questions are meaningful enough. To the best of our knowledge, we are the first to leverage weakly supervised synthetic QA pairs extracted from unlabeled conversations for a second-stage pretraining. Drawing insights from Mintz et al. (2009), we create a weakly supervised QA dataset from unlabeled conversations and an associated ontology. The synthetic QA pairs are constructed by matching unlabeled utterances against possible slot values in the ontology. This provides a general and cost-effective way to improve QA-based slot filling performance with easily obtainable data.

Experimental results show that (1) our QASF model significantly outperforms previous zero-shot systems on SNIPS (Coucke et al., 2018) and TOP (Gupta et al., 2018); (2) encoding natural questions help models better leverage weakly supervised signals in the pretraining phase, compared to encoding raw descriptions.

2 Task Definition

Given an input utterance $u$, a slot filling model extracts a set of (slot type, span) pairs $(s_i, a_k)$, $i = 1, \ldots, k$ where $s_i$ comes from a fixed set of slot types $S$, and each $a_k = (j, k), 1 \leq j < k \leq |u|$ is a span in $u$. Each slot type is accompanied with a short textual description that describes its semantic meaning (Table 1). We also assume that a small amount of example slot values are given, following Shah et al. (2019).

Our goal is to build a slot filling model that performs well on a new target domain with unseen slot types. Our training data consists of utterances from $N$ source domains $D_1, D_2, \ldots, D_N$. Each domain $D_i$ is associated with a set of predefined slot types $S_i$. At test time, utterances are drawn from a new domain $D_{N+1}$. The new domain contains both seen and unseen slot types from the source domain. For example, in the SNIPS dataset (Coucke et al., 2018), domains “GetWeather” and “BookRestaurant” both have a slot type called “city”, while “condition_temperature” only appears in the “GetWeather” domain.

3 Methodology

In this section, we describe our framework for Question Answer-driven Slot Filling (QASF). The framework consists of (1) a question generation strategy that turns slot descriptions into natural language questions based on linguistic rules; (2) a generic span-extraction-based question answering model; (3) an intermediate pretraining stage with generated synthetic QA pairs from unlabeled conversations, which is before task-specific training.

3.1 Question Generation Strategy

To benefit from both language model pretraining and QA supervision, we design a question generation strategy to turn slot descriptions into natural questions. During this process, a considerable amount of knowledge and semantic information is encoded (Heilman, 2011). A generated question consists of a WH word and a normalized slot description following the template below:

$$\text{WH_word is slot_description?}$$

Generating WH_word We draw insights from the literature on automatic question generation. Heilman and Smith (2010) propose to use linguistically motivated rules. In their more general case of question generation from the sentence, answer phrases can be noun phrases (NP), prepositional phrases (PP), or subordinate clauses (SBAR). Complicated rules are designed with help from superTagger (Ciaramita and Altun, 2006).

For our spoken language understanding (SLU) tasks, slot fillers are mostly noun phrases.

Therefore, we design a simpler set of conditions based on named entity types and part-of-speech (POS) tags. For each slot type, we sample 10 (utterance, slot value) examples from the validation set. Then we run a NER and a POS tagging model to obtain entity types and POS tags for each of the sampled answer spans. Finally, we select WH_word based on a set of rules described in Table 6 in Appendix.

Generating slot_description Instead of directly adding a raw description phrase in the question template, we normalize the phrase with the

---

1Around 90% cases in the SNIPS dataset.
2Provided by spaCy: https://spacy.io/
Pre-given Ontologies:

| Attribute            | Possible Values          |
|----------------------|--------------------------|
| hotel-price_range     | expensive, cheap, moderate |
| hotel-star_rating     | 1-star, 2-star, 3-star, 4-star, 5-star |
| ...                  | African, Polish, Indian, Chinese |
| restaurant-cuisine    | Chinese, ...

Unlabeled Conversation Example:

**USER**: I am looking for a place to stay in the north of the city. I would prefer a 4-star hotel please.

**SYS**: There are several guesthouses available. Do you have a price reference?

**USER**: The restaurant should be in the moderate price range.

**String matching**

**Questioning**

**SYS**: How is the price?

**USER**: The restaurant price should be in the moderate [price_range].

...
Mintz et al. (2009) for creating a weakly supervised dataset. Figure 2 illustrates the process. Given an ontology or database of slot types and all possible values for each slot type, we find all utterances containing those value strings in a large set of unlabeled conversations. For example, in Figure 2, for the “hotel_price_range” slot, there are three possible values “expensive”, “cheap” and “moderate” in the ontology. We then form question-answer-utterance triples using the question generation strategy proposed in Section 3.1.

To obtain the pre-defined ontology and unlabeled conversations, we use MultiWOZ 2.2 (Zang et al., 2020), which is an improved version of MultiWOZ (Budzianowski et al., 2018). We do not use annotations in the dataset such as the (changes of) states in the conversations and we treat each utterance independently. We remove slot types that exist in the task-specific training/test data (i.e., SNIPS and TOP) from the ontology and end up with 67,370 QA examples for pretraining.

4 Experiments

4.1 Datasets and Baselines

SNIPS (Coucke et al., 2018) is an SLU dataset consisting of crowdsourced user utterances with 39 slots across 7 domains – “AddToPlaylist” (ATP), “BookRestaurant” (BR), “GetWeather” (GW), “PlayMusic” (PM), “RateBook” (RB), “SearchCreativeWork” (SCW), “FindScreeningEvent” (FSE). It has around 2000 training instances per domain. The slot types of each domain do not overlap with each other. Following previous work (Shah et al., 2019; Liu et al., 2020), we use this dataset to evaluate zero-shot cross-domain transfer learning – train on all training instances from domains other than $D_i$, and test exclusively on $D_i$, for $i = 1, \ldots, 7$. TOP (Gupta et al., 2018) is a task-oriented utterance parsing dataset. It is based on a hierarchical annotation scheme for annotating utterances with nested intents and slots. Each slot type also comes with a description. In our setup, we train on all seven domains of SNIPS as well as varying amounts of training data from the TOP training set (0, 20, and 50 examples), and use the TOP test set as an out-of-distribution domain for evaluation. We report span-level F1 (micro-average).

We compare our method against a number of representative baselines. Concept Tagger (CT) (Bapna et al., 2017) is a slot-filling framework that directly uses original slot descriptions to generalize to unseen slot types. Robust Zero-shot Tagger (RZT) (Shah et al., 2019) is an extension of CT, which incorporates example values of slots to improve the robustness of the model’s zero-shot capability. Coach (Liu et al., 2020) is a coarse-to-fine model for slot-filling. It also encodes raw slot descriptions. We also include a Zero-Shot BERT Tagger (ZSBT) based on BERT (Devlin et al., 2019) as an additional baseline. ZSBT directly encodes raw slot descriptions and utterances and predicts a tag (B, I, or O) for each token in the utterance.

4.2 Results and Analysis

We report F-1 of the baselines and our model on each target domain test set of SNIPS as well as average F-1 across domains. All models are trained on the other six domains for each target domain. As shown in Table 2, our QA-driven slot filling framework (QASF) significantly outperforms all baselines in five of the seven domains, with slightly lower performance on BookRestaurant than ZSBT, and lower performance on FindScreeningEvent than Coach. The average F-1 of QASF is around 7% higher than the prior published state-of-the-art Coach model, and about 2% higher than the Zero-shot BERT Tagger baseline. Adding the intermediate pre-training stage on weakly supervised data further improves performance on top of QASF in six of the seven domains except for AddToPlaylist. On average, adding pre-training improves over QASF by 2.9% F-1. The zero-shot performance of all models are relatively worse on PlayMusic, RateBook and FindScreeningEvent. A more detailed discussion is in Appendix Section C.

Table 3 summarizes TOP test results: (1) In both the zero-shot and few-shot settings, our QASF outperforms ZSBT, with a bigger improvement on the zero-shot setting. (2) Pretraining on the weakly supervised QA pairs helps more in the zero-shot setting than in the few-shot setting, with a 20% relative improvement. This shows that QASF (w/ pretraining) is more robust to the domain shift when there is no target domain training data.

Impact of QG strategy and pretraining To understand the influence of question generation and impact of pretraining with synthetic QA pairs, we perform ablation studies of both components on the SNIPS dataset. The table below shows ablation results (F-1). “w/o QG” refers to a model trained with raw slot descriptions and utterances.

Firstly, the question generation strategy consis-
| Model Type | Zero-shot | Few-shot (20) | Few-shot (50) |
|-----------|-----------|--------------|--------------|
| Random NE | 1.34      | -            | -            |
| ZSBT      | 8.82      | 37.60        | 42.73        |
| QASF (ours) | 10.27    | 36.86        | 46.49        |
| w/ pre-training on WS | 12.35 | 39.78 | 47.91 |

Table 2: Experimental results (F-1) on SNIPS dataset. * indicates statistical significance (p < 0.05), **: p < 0.01.

| Model Type | Zero-shot | Few-shot (20) | Few-shot (50) |
|-----------|-----------|--------------|--------------|
| Random NE | 1.34      | -            | -            |
| ZSBT      | 8.82      | 37.60        | 42.73        |
| QASF (ours) | 10.27    | 36.86        | 46.49        |
| w/ pre-training on WS | 12.35 | 39.78 | 47.91 |

Table 3: Evaluation results on TOP test. Models trained on SNIPS, and varying amount of utterances of TOP train – zero-shot, 20-shot (1%), 50-shot (2.5%).

Table 4: Ablation Study

| Model Type | w/o pretraining | w/ pretraining |
|-----------|-----------------|----------------|
| QA SF     | 44.45           | 41.97          |
| Δ (F1)    | +5.91%          | +9.79%         |

The variance between the source and target domains. Sometimes even slot types of the same name refer to different kinds of objects in different domains. For example, slot type “object_type” in the “RateBook” domain refers to object types like textbook, essay and novel; while in the “FindScreeningEvent”, it refers to event types like movie times/schedules. In the two domains, they have the same raw descriptions. In the table below, we show the performance of models on utterances with “object_type” and “object_name” spans (according to gold annotations). We can see that the performances on these special slots are significantly lower than the general average on all the examples (40–50%). But still, the questioning strategy helps improve the transferring of semantic information.

Annotation artifacts of SNIPS dataset and sparsity of vocabulary for certain slot types. Our QASF framework does not perform well on the target domain “BookRestaurant”, thus we take a close look at it. We find that there are only 25 possible values in total for slot restaurant_type, over 51% of them are of a single token “restaurant” (Table below). A very simple approach (assigning type “restaurant_type” to all tokens “restaurant” can obtain decent performance). This does not happen for other slot types in BookRestaurant (e.g., cuisine, restaurant_name). The possible values are more diverse and the distribution is more balanced.

| Slot Value | Prop | % |
|------------|------|---|
| restaurant | 51.81 | 7.26 |
| bar        | 6.60 | 6.23 |
| pub        | 6.23 | 6.23 |
| brasserie  | 6.23 | 6.23 |

5 Conclusion

We propose a QA-driven method with weakly supervised pretraining for zero-shot slot filling. Our experimental results and analyses demonstrate the benefits of QA-formulation, especially in the setting with synthetic QA pairs pretraining.

Acknowledgments

We thank Kenton Lee and Emily Pitler, and anonymous reviewers for their constructive suggestions.
References

Ankur Bapna, Gokhan Tür, Dilek Hakkani-Tür, and Larry Heck. 2017. Towards zero-shot frame semantic parsing for domain scaling. In Proc. Interspeech 2017, pages 2476–2480.

T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, P. Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, Tom Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. ArXiv, abs/2005.14165.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Itigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pages 594–602.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint arXiv:1805.10190.

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.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

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.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Hsu, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 753–757.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.

Hangfeng He, Qiang Ning, and Dan Roth. 2020. QuASE: Question-answer driven sentence encoding. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8743–8758, Online. Association for Computational Linguistics.

Luheng He, M. Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In EMNLP.

Michael Heilman. 2011. Automatic factual question generation from text. Ph.D. thesis, Ph. D. thesis, Carnegie Mellon University.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 609–617.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5849–5859, Online. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 19–25, Online. Association for Computational Linguistics.
Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3449–3460, Florence, Italy. Association for Computational Linguistics.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. arXiv preprint arXiv:1806.08730.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2014. Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(3):530–539.

Julian Michael, Gabriel Stanovsky, Luheng He, I. Dagan, and Luke Zettlemoyer. 2017. Crowdsourcing question-answer meaning representations. ArXiv, abs/1711.05885:560–568.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Mahdi Namazifar, Alexandros Papangelis, Gokhan Tur, and Dilek Hakkani-Tur. 2020. Language model is all you need: Natural language understanding as question answering. arXiv preprint arXiv:2011.03023.

Feng Pan, Rutu Mulkar-Mehta, and Jerry R Hobbs. 2011. Annotating and learning event durations in text. Computational Linguistics, 37(4):727–752.

Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5484–5490, Florence, Italy. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nanwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. Linguistic Data Consortium, Philadelphia, PA, 23.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6953–6963, Online. Association for Computational Linguistics.

Steve Young. 2002. Talking to machines (statistically speaking). In Seventh International Conference on Spoken Language Processing.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pages 109–117, Online. Association for Computational Linguistics.
A Details of Question Generation

The part-of-speech tagset is based on the Universal Dependencies scheme \(^4\). The named entity labels are based on OntoNotes 5.0 (Weischedel et al., 2013). In Table 6, we describe the set of rules for the selection of WH_word.

B Inference Constraints

At inference time, predicting the slot filler spans is more complex – for each slot type, as there can be several or no spans to be extracted. After the output layer, we have the probability of each token \(x_i\) being the start \((P_s(i))\) and end \((P_e(i))\) of the span.

We harvest all the valid candidate spans for each slot type with the following heuristics:

1. Enumerate all possible combinations of start offset \(\text{start}\) and end offset \(\text{end}\) of the spans \((\frac{M(M-1)}{2}\) candidates in total);

2. Eliminate the spans not satisfying the constraints: (1) start and end token must be within the utterance; (2) the length of the span should be shorter than a maximum length constraint; (3) spans should have a larger probability than the probability of “no answer” (which is represented with [CLS] token), namely,

\[
P_s(\text{start}) > P_s([\text{CLS}]), P_e(\text{end}) > P_e([\text{CLS}])
\]

C Further Analysis and Discussions

We conduct further analysis to understand how and why the models are effective.

C.1 Impact of question generation strategy and pretraining

Table 7 shows full ablation results.

C.2 Analysis on Seen versus Unseen Slots

To understand the transferring capability of our models, we further split the SNIPS test for each target domain into “seen” and “unseen” slots. An example is categorized into “unseen” as long as there is an unseen slot (i.e., the slot does not exist in the remaining six source domains in its gold annotation.) Otherwise, it counts as “seen”. A full list of unseen slots for each target domain can be found in the Appendix.

As is shown in Table 5, we can see that (1) both ZSBT baseline and our models perform better on the “seen” slots than the “unseen” ones – the numbers substantially drop on the “unseen” slots. This proves that transferring from the source domains to the unseen slots in the target domain is a hard problem. (2) On the portion of examples with “seen” slots, our best model outperforms ZSBT with around a 2% margin. (3) On the “unseen” portion of examples, the margin is larger – our QASF and pretraining step help improve the performance more (over 4%). The second and third observation together demonstrates that the questioning strategy help improve the model’s capability of transferring between related but not exactly same slot types (e.g., “object_name” and “entity_name”).

|            | Seen  | Unseen |
|------------|-------|--------|
| ZSBT       | 54.75 | 37.41  |
| QASF       | 56.79 | 39.99  |
| w/ pretraining | 56.23 | 41.73  |

Table 5: Averaged F-1 scores over all target domains of SNIPS dataset (for “unseen” and seen “slots”).

D Hyper-parameters and Training Details

We use the uncased version of the BERT-base (Devlin et al., 2019) model for QA finetuning and pretraining. The model is fine-tuned for 5 epochs with a starting learning rate of 3e-5 on the SNIPS dataset. The model is pretrained for 5 epochs with a starting learning rate of 5e-7 on the synthetic QA dataset. Our implementations are based on https://github.com/google-research/bert/blob/master/run_squad.py
| WH_word | Conditions | Answer Examples |
|---------|------------|----------------|
| How long | The answer phrase is modified by a cardinal number (CARDINAL) or quantifier phrase (QUANTITY) whose object is a temporal unit, as is defined in (Pan et al., 2011), i.e., second/minute/hour/day/week/month/year/decade/century. | 2 nights |
| How many | The answer phrase is modified by a cardinal number (CARDINAL) or quantifier phrase (QUANTITY) and the object is not a temporal unit. | 2 stars, 3 tickets |
| How | adjective ADJ | moderate, expensive |
| When | The answer phrase’s head word is tagged DATE or TIME | 1:30 PM, 1999 |
| Who | The answer phrase’s head word is tagged PERSON or is a personal pronoun PRON (I, he, herself, them, etc.) | mother, Dr. Williams |
| Where | The answer phrase is a prepositional phrase whose object is tagged GPE or LOC, whose preposition is one of the following: on, in, at, over, to | amc theaters, fort point san francisco, east, west, ... |
| Which | The answer phrase is a determiner DT (this, that) or an ordinal ORDINAL | this, first, current, last |
| What | all other cases | |

Table 6: Strategy for Generating the WH_word (question phrase).

| | ATP | BR | GW | PM | RB | SCW | FSE | Average F1 |
|---|---|---|---|---|---|---|---|---|
| QASF w/o pretraining | 59.29 | 43.13 | 59.02 | 33.62 | 33.34 | 59.90 | 22.83 | 44.45 |
| QASF w/ pretraining | 55.30 | 46.71 | 53.06 | 35.79 | 25.28 | 59.77 | 17.85 | 41.97 |
| w/o question | 57.57 | 48.75 | 61.27 | 38.54 | 36.51 | 60.82 | 27.72 | 47.31 |
| w/o question | 56.11 | 42.42 | 55.70 | 33.07 | 33.13 | 60.03 | 21.20 | 43.09 |

Table 7: Full ablation analysis on SNIPS dataset.
### Schema of SNIPS dataset

| Domain                  | All Slots                                      | Unseen Slots                        |
|-------------------------|------------------------------------------------|--------------------------------------|
| AddToPlaylist (ATP)     | entity_name, playlist_owner, playlist, artist, music_item | entity_name, playlist_owner          |
|                         | restaurant_type, served_dish, restaurant_name, party_size_description, cuisine, party_size_number, timerange, facility, poi, state, city, country, sort, spatial_relation | restaurant_type, served_dish, restaurant_name, party_size_description, cuisine, party_size_number, timerange, facility, poi          |
| BookRestaurant (BR)     |                                             |                                      |
| GetWeather (GW)         | timerange, current_location, condition_description, geographic_poi, condition_temperature, country, state, city, spatial_relation | timerange, current_location, condition_description, geographic_poi, condition_temperature          |
| PlayMusic (PM)          | year, genre, service, album, track, sort, music_item, artist, playlist | year, genre, service, album, track |
| RateBook (RB)           | object_select, rating_value, best_rating, rating_unit, object_part_of_series_type, object_type, object_name | object_select, rating_value, best_rating, rating_unit, object_part_of_series_type, object_name |
| SearchCreativeWork (SCW)| object_type, object_name | -                                    |
| FindScreeningEvent (FSE)| object_location_type, movie_name, movie_type, timerange, location_name, object_type, spatial_relation | object_location_type, movie_name, movie_type, timerange, location_name, object_type, spatial_relation |

Table 8: Schema of SNIPS dataset
## F Question Templates for SNIPS

| Domain                | Slot                        | Slot Name            | Natural Question                      |
|-----------------------|-----------------------------|----------------------|---------------------------------------|
| AddToPlaylist         | music_item                  | music item           | what’s the music item?                |
| AddToPlaylist         | playlist_owner              | owner                | who’s the owner?                      |
| AddToPlaylist         | entity_name                 | entity name          | what’s the entity name?               |
| AddToPlaylist         | playlist                    | playlist             | what’s the playlist?                  |
| AddToPlaylist         | artist                      | artist               | who’s the artist?                     |
| BookRestaurant        | city                        | city                 | what’s the city?                      |
| BookRestaurant        | facility                    | facility             | what’s the facility?                  |
| BookRestaurant        | timeRange                   | time range           | when’s the time range?                |
| BookRestaurant        | restaurant_name             | restaurant name      | what’s the name?                      |
| BookRestaurant        | country                     | country              | what’s the country?                   |
| BookRestaurant        | cuisine                     | cuisine              | what’s the cuisine?                   |
| BookRestaurant        | served_dish                 | served dish          | what’s the served dish?               |
| BookRestaurant        | particle_size_number        | number               | how many people?                      |
| BookRestaurant        | poi                         | position             | where’s the location?                 |
| BookRestaurant        | sort                        | type                 | what’s the type?                      |
| BookRestaurant        | spatial_relation            | spatial relation     | what’s the spatial relation?          |
| BookRestaurant        | state                       | location             | what’s the state?                     |
| BookRestaurant        | party_size_description      | person               | who are the persons?                  |
| GetWeather            | city                        | city                 | what’s the city?                      |
| GetWeather            | state                       | location             | what’s the state?                     |
| GetWeather            | timeRange                   | time range           | when’s the time range?                |
| GetWeather            | current_location            | current location     | what’s the current location?          |
| GetWeather            | country                     | country              | what’s the country?                   |
| GetWeather            | geographic_poi              | geographic position  | where’s the location?                 |
| GetWeather            | condition_temperature       | temperature          | how is the temperature?               |
| GetWeather            | condition_description       | weather              | how is the weather?                   |
| PlayMusic             | genre                       | genre                | what’s the genre?                     |
| PlayMusic             | music_item                  | music item           | what’s the music item?                |
| PlayMusic             | service                     | service              | what’s the service                    |
| PlayMusic             | year                        | year                 | when’s the year?                      |
| PlayMusic             | playlist                    | playlist             | what’s the playlist?                  |
| PlayMusic             | album                       | album                | what’s the album?                     |
| PlayMusic             | sort                        | type                 | what’s the type?                      |
| PlayMusic             | track                       | track                | what’s the track?                     |
| PlayMusic             | artist                      | artist               | who’s the artist?                     |
| RateBook              | object_part_of_series_type  | series               | what’s the series?                    |
| RateBook              | object_select               | this current          | which to select?                     |
| RateBook              | rating_value                | rating value         | how many rating value?                |
| RateBook              | object_name                 | object name          | what’s the object name?               |
| RateBook              | object_type                 | object type          | what’s the object type?               |
| RateBook              | rating_unit                 | rating unit          | what’s the rating unit?               |
| RateBook              | best_rating                 | best rating          | how many rating points in total?      |
| SearchCreativeWork    | object_name                 | object name          | what’s the object name?               |
| SearchCreativeWork    | object_type                 | object type          | what’s the object type?               |
| SearchScreeningEvent  | timeRange                   | time range           | when’s the time range?                |
| SearchScreeningEvent  | movie_type                  | movie type           | what’s the movie type?                |
| SearchScreeningEvent  | object_location_type        | location type        | what’s the location type?             |
| SearchScreeningEvent  | object_type                 | object type          | what’s the object type?               |
| SearchScreeningEvent  | location_name               | location name        | where’s the location name?            |
| SearchScreeningEvent  | spatial_relation            | spatial relation     | what’s the spatial relation?          |

Table 9: Question Templates for SNIPS