Fast Domain Adaptation of Semantic Parsers via Paraphrase Attention

Avik Ray, Yilin Shen and **Hongxia Jin**
Samsung Research America, Mountain View, California, USA
{avik.r, yilin.shen, hongxia.jin}@samsung.com

Abstract

Semantic parsers are used to convert user's natural language commands to executable logical form in intelligent personal agents. Labeled datasets required to train such parsers are expensive to collect, and are never comprehensive. As a result, for effective post-deployment domain adaptation and personalization, semantic parsers are continuously retrained to learn new user vocabulary and paraphrase variety. However, state-of-the art attention based neural parsers are slow to retrain which inhibits real time domain adaptation. Secondly, these parsers do not leverage numerous paraphrases already present in the training dataset. Designing parsers which can simultaneously maintain high accuracy and fast retraining time is challenging. In this paper, we present novel paraphrase attention based sequence-to-sequence/tree parsers which support fast near real time retraining. In addition, our parsers often boost accuracy by jointly modeling the semantic dependencies of paraphrases. We evaluate our model on benchmark datasets to demonstrate upto 9X speedup in retraining time compared to existing parsers, as well as achieving state-of-the-art accuracy.

1 Introduction

Semantic parsers are used in modern intelligent personal agents (e.g. Alexa, Bixby, Jibo) to allow users carry out a wide variety of tasks using natural language commands/queries. Specifically, these parsers convert the input query to an executable logical form representation. However, labeled datasets required to train state-of-the-art neural semantic parsers are difficult to collect due to their annotation complexity. Secondly, users from different locale tend to use different vocabulary, and paraphrases making it nearly impossible to collect a comprehensive dataset covering all possible variety of queries. As a result, once deployed, the semantic parsers require frequent retraining for adaptation to the locale and user specific vocabulary (Thomason et al., 2015; Azaria et al., 2016; Ray et al., 2018). Such domain adaptation and personalization is a key feature in current commercial personal agents (Kim et al., 2018).

Recently, neural semantic parsers based on attention based sequence-to-sequence/tree models were proposed (Jia and Liang, 2016; Dong and Lapata, 2016). These are attractive for commercial personal agents, since unlike previous approaches these can be trained end-to-end without requiring hand crafted domain specific grammar/lexicon, thereby improving scalability. However, these parsers are particularly prone to error when queries contain out-of-vocabulary (OOV) words (Ray et al., 2018). They are also slow to retrain since the attention layer, which is critical for boosting accuracy (Dong and Lapata, 2016), also constraints the encoder and decoder to be retrained simultaneously. As an example, in benchmark ATIS dataset with 4,485 training queries, a sequence-to-sequence semantic parser requires over 1 hour retraining time using a single GPU.

In this paper, we present novel sequence-to-sequence/tree parsers with two key advantages over previous parsers. First, our parser is trained to use either attention from input query or attention from its paraphrase (referred as paraphrase attention) when available. For learning new vocabulary from paraphrased queries (Azaria et al., 2016; Ray et al., 2018), this naturally enables our parsers to be retrained much faster, since in our parser only the encoder requires retraining. Secondly, by jointly modeling the semantic dependencies between paraphrases, our parser often achieves better accuracy over previous models. Our main contributions are summarized below.

- We propose novel sequence-to-sequence and

Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo), pages 94–103
Hong Kong, China, November 3, 2019.
https://doi.org/10.18653/v1/P17

tree parsers with paraphrase attention which can be retrained much faster than previous models, enabling real time domain adaptation of intelligent agents.

- Our models explicitly leverage paraphrases in the training dataset resulting in better semantic understanding. On benchmark datasets our models achieve similar or better parsing accuracy over previous models.

- On OOV datasets, our models can learn new personalized words/phrases upto 9X faster than previous attention based parsers after retraining.

1.1 Related work

In this section we highlight the most related prior literature. In the last few decades a wide variety of semantic parsers have been proposed using both rule based and supervised approaches (Zelle and Mooney, 1996; Wong and Mooney, 2007; Zettlemoyer and Collins, 2005, 2007; Kwiatkowski et al., 2010, 2011; Artzi and Zettlemoyer, 2013). More recently, end-to-end neural network models are being explored due to their superior performance and ease of training (Jia and Liang, 2016; Dong and Lapata, 2016; Iyer et al., 2017; Dong and Lapata, 2018). The use of paraphrases to boost performance of semantic parsers have been studied (Berant and Liang, 2014; Ray et al., 2018).

Domain adaptation of semantic parsers have been explored in both pre–deployment (Herzig and Berant, 2017; Fan et al., 2017) and post–deployment (Thomason et al., 2015; Azaria et al., 2016; Iyer et al., 2017; Ray et al., 2018) settings, and using both CCG based and neural network parsers. In (Ray et al., 2018), the authors propose new models to effectively learn user specific OOV words by retraining neural semantic parsers.

Neural semantic parsers are mainly based on attention based sequence-to-sequence networks. Although sequence-to-sequence networks were first proposed to solve the problem of machine translation (Sutskever et al., 2014; Bahdanau et al., 2015), it has been applied successfully in a wide range of NLP tasks (Cho et al., 2014; Vinyals et al., 2015b; Prakash et al., 2016). While adding extra context information from the input in the form of attention network greatly improves the performance of these models (Bahdanau et al., 2015; Vinyals et al., 2015a; Dong and Lapata, 2016), they also slow down their retraining time by constraining both the encoder and decoder networks to be retrained simultaneously.

Our work lie in the intersection of these areas. We propose new sequence-to-sequence/tree parsers using paraphrase attention, which facilitates faster domain adaptation, while maintaining competitive parsing accuracy as current models.

This paper is organized as follows. Section 2 formally defines our problem and discuss related background. We describe our new paraphrase attention based parsers in Section 3. In Section 4 we present our numerical evaluation results. Finally, we conclude in Section 5.

2 Problem and Background

In this section, we concretely define our problem and discuss related notations. A semantic parser $\mathcal{P}$ converts an user provided query $\mathbf{q} = (w_1, \ldots, w_n)$ to its corresponding logical form representation $\mathbf{l}(\mathbf{q}) = (l_1, \ldots, l_m)$, where w_i-s represents words from vocabulary $\mathcal{V}$, and l_j-s correspond to logical expression tokens. The parser $\mathcal{P}$ is trained over a labeled training set T. After deployment, users often use their own personal or locale specific vocabulary in queries, some of which are absent in the training vocabulary $\mathcal{V}$. Let $\mathbf{p}^*$ be a query with OOV words which parser $\mathcal{P}$ cannot parse. We follow the post–deployment domain adaptation settings similar to (Azaria et al., 2016; Ray et al., 2018), where using user feedback/dialog, a paraphrased query $\mathbf{q}^*$ of $\mathbf{p}^*$ is obtained which is parsable. The main task of domain adaptation is to retrain $\mathcal{P}$ using both the given paraphrased sample $(\mathbf{p}^*, \mathbf{q}^*, \mathbf{l}(\mathbf{q}^*))$, and the training set T to obtain an improved personalized parser $\mathcal{P}'$.

2.1 Sequence-to-sequence/tree parsers

In (Dong and Lapata, 2016; Jia and Liang, 2016), the authors demonstrate that attention based sequence-to-sequence/tree models can be utilized to solve the semantic parsing task. A basic attention based sequence-to-sequence/tree parser consists of an encoder, a decoder, and an attention layer. The encoder, and the decoder again consists of recurrent neural networks (e.g. LSTM). The model is trained by maximizing the simplified likelihood function:

$$P(l_1, .., l_m | w_1, .., w_n) = \Pi_{t=1}^{m} P(l_t | l_1, .., l_{t-1}, \mathbf{c}) \quad (1)$$

where $\mathbf{c}$ is the context vector (or final encoder hidden state).

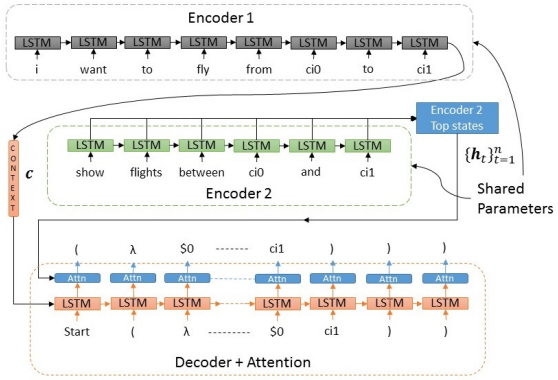


Figure 1: Figure showing our new sequence-to-sequence parser with paraphrase attention. During training, encoder 1 encodes a query (*"i want to fly from ci0 to ci1"*), and encoder 2 encodes its paraphrase (*"show flights between ci0 and ci1"*). For decoding, the context state is provided by encoder 1 and attention is computed from encoder 2 top states. Both encoders share same LSTM parameters. During inference, a single encoder is used to compute both context and attention states.

3 Our Model

In this section we describe our new sequence-to-sequence/tree parser using paraphrase attention. The motivation behind our parser is as follows. Existing attention based sequence-to-sequence/tree parsers are slow to retrain since both the encoder and decoder needs to be retrained simultaneously to achieve satisfactory accuracy, hindering real time domain adaptation. While it may be possible to freeze the decoder parameters, and finetune only the encoder, however this is still slow since the error gradients need to be propagated all the way back to the encoder. Freezing the encoder parameters and finetuning just the decoder results in poor performance (shown in evaluation Section 4) since the model fails to learn the proper encoder representation corresponding to new OOV words.

Recall that, to teach a query $\mathbf{p}^*$ with OOV words to an intelligent agent, user provides a paraphrased query $\mathbf{q}^*$ with known words which the parser $\mathcal{P}$ understands. Since $\mathbf{p}^*$ and $\mathbf{q}^*$ have the same meaning (hence the same logical form), their context representation $\mathbf{c}$ should be the same. We make two key observations. First, to learn new query $\mathbf{p}^*$ we can finetune just the encoder

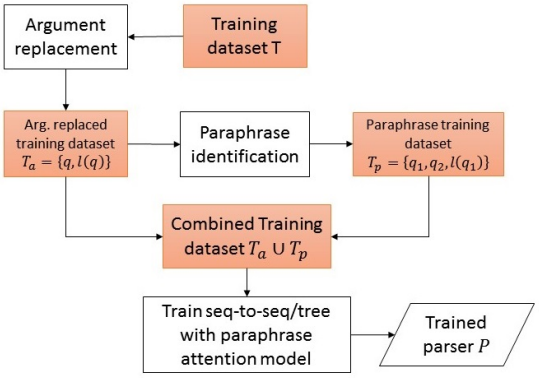


Figure 2: Illustration of data pre-processing and training process of our sequence-to-sequence/tree with paraphrase attention parser.

by treating the context vector $\mathbf{c}$ (computed from paraphrased query $\mathbf{q}^*$) as ground-truth. However, the top encoder states corresponding to query $\mathbf{p}^*$, which are required for attention computation, are still unknown. We make a second observation that, if during training the model is taught to use attention either from a query, or its paraphrase, we can simply use attention from $\mathbf{q}^*$ to decode $\mathbf{p}^*$. Therefore, we do not require knowledge of top encoder states of $\mathbf{p}^*$, instead we just need to know those of $\mathbf{q}^*$. This can be obtained since $\mathcal{P}$ can correctly parse $\mathbf{q}^*$. Before describing our model details, we discuss another essential data processing step which identifies paraphrased sentences for model training.

3.1 Data preprocessing

We now describe two key preprocessing steps required to train our paraphrase attention model. Figure 2 provides an overview of these data processing steps.

Argument replacement: In (Dong and Lapata, 2016; Ray et al., 2018), the authors use an important preprocessing step called argument replacement. This step replaces certain words/phrases in the user query (e.g. entities or numbers), which correspond to logical form arguments, using special argument tokens before training. This greatly improves parsing accuracy by reducing the input variability (Dong and Lapata, 2016). Figure 1 shows an example of argument replaced query *"i want to fly from ci0 to ci1"* for the original query *"i want to fly from atlanta to philadelphia"* in ATIS dataset, where $ci0, ci1$ are special argument tokens. As a first preprocessing step, we perform this argument replacement to convert original training set T to an argument replaced training set

T_a.

Paraphrase identification: In order to train a sequence-to-sequence parser which can use attention either from a query or its paraphrase, first we need to identify sentential paraphrases in the training dataset. Intuitively, if two paraphrased queries have the same meaning, they must share the same logical form. However, in the original training set T there are not many paraphrases, since often logical forms differ only by a constant. Instead, if we consider the argument replaced training set T_a, where such constants have been replaced by argument tokens, many identical logical forms exist. In our second paraphrase identification step, using the queries in T_a whose paraphrase exist, we construct a new **paraphrase training set** $T_p = \{\mathbf{q}_1^i, \mathbf{q}_2^i, \mathbf{l}(\mathbf{q}_1^i)\}_{i=1}^{p}$ of a given size p, where $\mathbf{q}_1^i, \mathbf{q}_2^i$ are paraphrases. An example of such paraphrase pair is shown in Figure 1.

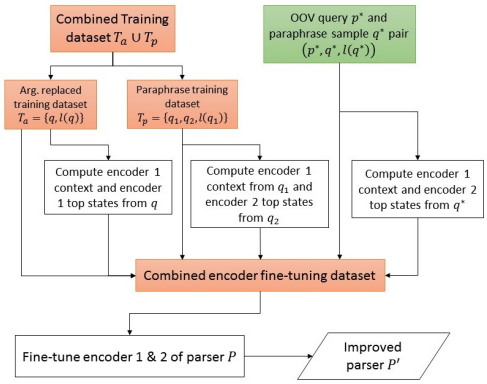


Figure 3: Illustration of fast encoder fine-tuning process in our sequence-to-sequence/tree parser $\mathcal{P}$.

3.2 Sequence-to-sequence/tree with paraphrase attention

Now we describe our sequence-to-sequence with paraphrase attention model. Our model consists of two encoders (with shared parameters), one decoder, and one attention layer as shown in Figure 1. Sequence encoders 1 and 2 encode a query and its paraphrase respectively. During, decoding the decoder context is initialized from encoder 1 final hidden state, however the attention states are computed using the top hidden states of encoder 2. This enables our model to jointly capture the semantic dependence between the paraphrase pair.

Training: A key feature of our model is that the attention network is able to generate attention signals from both the query, or its paraphrase. To ensure this, we train our model on a combined dataset $T_a \bigcup T_p$ using a multi-task objective. For a sample without a paraphrase $(\mathbf{q}, \mathbf{l}(\mathbf{q})) \in T_a$, we perform forward/backward propagation identical to sequence-to-sequence with attention parsers using a single encoder (either 1 or 2 since they share parameters). For a sample with paraphrase $(\mathbf{q}_1, \mathbf{q}_2, \mathbf{l}(\mathbf{q}_1)) \in T_p$, we perform forward/backward propagation using both encoders, and use the attention from encoder 2. We use the overall negative log-likelihood function as our training objective.

$$\mathcal{L} = -\log P(\mathbf{l}(\mathbf{q}_1)|\mathbf{q}_1, \mathbf{q}_2)$$

Figure 2 provides an overview of the data processing steps and training procedure of our new parser.

Inference: During model inference/testing we only have one user provided query $\mathbf{q}$ and no paraphrase. However, thanks to the shared encoder parameters, both encoder context and top states can be computed from this query $\mathbf{q}$. Therefore, during inference the model essentially acts as a normal sequence-to-sequence/tree parser.

Fast domain adaptation: For domain adaptation, user provides a OOV query $\mathbf{p}^*$ and its paraphrase $\mathbf{q}^*$. In end-to-end neural network models it is straight-forward to fine-tuning the parser $\mathcal{P}$ after adding this new sample $(\mathbf{p}^*, \mathbf{q}^*, \mathbf{l}(\mathbf{q}^*))$ to the training set. As mentioned before, in previous sequence-to-sequence parsers such fine-tuning is slow since it involves updating the entire model parameters. In our model, a faster alternative is to fine-tune just the encoder. We can perform this by using a MSE objective as follows.

$$\mathcal{L} = \sum_{t=1}^{|\mathbf{q}|} \|\bar{\mathbf{h}}_t - \hat{\mathbf{h}}_t\|^2 + \|\bar{\mathbf{c}} - \hat{\mathbf{c}}\|^2$$

where $\{\hat{\mathbf{h}}_t\}_{t=1}^{|\mathbf{q}|}, \hat{\mathbf{c}}$ are the predicted top encoder states, and context vector respectively; while $\{\bar{\mathbf{h}}_t\}_{t=1}^{|\mathbf{q}|}, \bar{\mathbf{c}}$ are their ground-truths. For all training samples of $\mathcal{P}$, the ground-truths can be computed by a single forward pass using $\mathcal{P}$. Unfortunately, in sequence-to-sequence with attention parser, the ground-truth top encoder states $\{\bar{\mathbf{h}}_t\}_{t=1}^{|\mathbf{p}^*|}$, for the new OOV query $\mathbf{p}^*$, are unknown. This cannot be computed even from the paraphrase $\mathbf{q}^*$ since they may have different lengths (example in Figure 1). In our paraphrase attention model, we can naturally fine-tune the encoder, since the attention is computed from a paraphrase $\mathbf{q}^*$. Specifically,

1. The ground-truth encoder 1 context $\bar{\mathbf{c}}$ is computed by encoding $\mathbf{q}^*$ using encoder 2, since query $\mathbf{p}^*$ should have the same meaning representation as $\mathbf{q}^*$.

2. The ground-truth encoder 2 states $\{\bar{\mathbf{h}}_t\}_{t=1}^{|\mathbf{q}^*|}$ are also computed by encoding $\mathbf{q}^*$ using encoder 2. Since only encoder 2 provides the attention signal, the different length of query $\mathbf{p}^*$ is irrelevant.

Figure 3 illustrates the encoder fine-tuning process for our parser. Note that, so far we have mainly described our sequence-to-sequence with paraphrase attention parser. However we can easily construct a similar sequence-to-tree with paraphrase attention parser by simply replacing the sequence decoder with the tree decoder in (Dong and Lapata, 2016).

Discussion: Note that, by fine-tuning using a MSE objective may result in a new context vector $\hat{\mathbf{c}}$ which is perturbed from the original semantic feature space that represented training queries. Thankfully, we observe in our experiments that the intermediate neural network layers are robust to small perturbations from the context feature space, and may not result in any significant changes in final classification output (or accuracy). Such robustness of intermediate layers have also been observed in works on neural network model compression (Denton et al., 2014; Aghasi et al., 2017; Kasiviswanathan et al., 2018).

To the best of our knowledge, this is the first work to use attention from paraphrase to improve parsing accuracy, and retraining time. Observe that, we harness accurate paraphrases from the training dataset itself as opposed to noisy auto-generated paraphrases from external resource like PPDB (Ganitkevitch et al., 2013; Dong et al., 2017), or a domain specific KB (Berant and Liang, 2014) used in recent literature. Moreover, in low resource languages such external paraphrase resource are generally unavailable. In (Ray et al., 2018) the authors train a paraphrase generator by first training an auto-encoder, and subsequently fine-tuning it with user provided paraphrases. In contrast, our model is trained using paraphrases identified within the training data even without any user input. Our model can also be trained end-to-end, unlike the hybrid parser model of (Ray et al., 2018).

4 Experiments

In this section we present our evaluation results. We have the following objectives. First we show that our new parsers using paraphrase attention can achieve a competitive or better parsing accuracy over previous models on benchmark datasets. Next, we present the main result that our new models can be retrained significantly faster to learn new OOV words/phrases than previous models.

4.1 Datasets

In order to test the performance of our model we consider three benchmark semantic parsing datasets:

1. *airline queries* dataset (ATIS) with 5,410 queries (4,480 training, 480 validation, 450 test)
2. *geographical queries* dataset (GEO) with 880 queries
3. *job queries* dataset (JOB) with 640 queries

For ATIS dataset we use the standard train-test split for our evaluation. However, for GEO and JOB datasets, owing to their small size, the parsing accuracy can vary significantly depending on the chosen split. Hence, in these smaller datasets we perform a 10 fold validation similar to (Wong and Mooney, 2007; Lu et al., 2008; Ray et al., 2018).

To test domain adaptation, we use OOV datasets used in (Ray et al., 2018) referred as PARA-ATIS and PARA-GEO datasets respectively (examples in Table 1). These datasets were constructed from benchmark datasets by substituting words w in the benchmark queries by synonymous OOV words and phrases $s \in Syn(w)$, to generate candidate paraphrases. For a given train–test split, the dataset is in the form of tuple pairs (word w, synonym s, $T_{trn}(w,s)$, $T_{tst}(w,s)$), where $T_{trn}(w,s)$/$T_{tst}(w,s)$ denotes the subset of queries from original train/test set where w has been replaced by s. The PARA-GEO dataset contains 180 word–synonym pairs and 5,783 OOV queries; while the PARA-ATIS dataset contains 161 word–synonym pairs and 13,501 OOV queries. Note that, the crowdsourced benchmark datasets contain typical queries that most users may ask. However, in order to test domain adaptation we need to consider atypical queries which are rare overall, but important for a particular user or locale. Hence, these OOV datasets containing atypical queries are suitable for this evaluation task.

Benchmark	Original benchmark query $\mathbf{q}^*$	OOV substituted query $\mathbf{p}^*$	Logical form $\mathbf{l}(\mathbf{q}^*)$	OOV dataset
ATIS	list all flights departing from $ap0$	list all flights taking off from $ap0$	(λ \$0 (and (flight \$0) (from \$0 $ap0$)))	PARA-ATIS
ATIS	i need a flight from $ci0$ to $ci1$	i require a flight from $ci0$ to $ci1$	(λ \$0 (and (flight \$0) (from \$0 $ci0$) (to \$0 $ci1$)))	PARA-ATIS
GEO	how many big cities are in $s0$	how many large cities are in $s0$	(count (λ \$0 (and (major \$0) (city \$0) (loc \$0 $s0$))))	PARA-GEO
GEO	which state has the highest elevation	which state has the highest natural elevation	(argmax (λ \$0 (state \$0)) (λ \$1 (elevation \$1)))	PARA-GEO

Table 1: Table showing examples from OOV datasets PARA-GEO and PARA-ATIS which were constructed from the benchmark GEO and ATIS datasets (Ray et al., 2018). Underlined words in the original benchmark queries are replaced with synonymous out-of-vocabulary words and phrases.

Model	10 fold accuracy %
COCKTAIL (Tang and Mooney, 2001)	79.40
argument transfer (Ray et al., 2018)	88.59
seq-to-seq + attention (our baseline)	93.75
seq-to-tree + attention (our baseline)	**95.31**
seq-to-seq + paraphrase attention (our model)	**95.31**
seq-to-tree + paraphrase attention (our model)	**95.31**

Table 2: Comparison of best 10 fold accuracy of all models on benchmark JOB dataset. In our paraphrase attention models we use $p = 50$ paraphrase pairs.

4.2 Methodology

First we train our parsers $\mathcal{P}$ on the combined dataset $T_a \cup T_p$, where T_a correspond to the argument replaced benchmark dataset, and T_p contain p[1] randomly sampled paraphrase pairs from the dataset T_a. We compare the parsing accuracy (computed as exact logical form match) on the test set with baseline attention based sequence-to-sequence and tree models by (Dong and Lapata, 2016). Next, to evaluate retraining performance, we follow the same experimental setup as (Ray et al., 2018). We fine-tune the parser $\mathcal{P}$ adding at most 5 samples (i.e. user provides 5 paraphrase pairs) from OOV training set $T_{trn}(w, s)$ to training set of $\mathcal{P}$, and test accuracy on the corresponding OOV test set $T_{tst}(w, s)$ (referred as the **retrained accuracy**). Within an appropriate retraining period, let t_b be the minimum time required by the baseline model to achieve best retrained accuracy, and t_p be the minimum time required by our paraphrase attention model to achieve the same retrained accuracy. We compute the retraining speedup t_b/t_p achieved by our parser over baseline. An alternative evaluation methodology involves crowdsourcing sentence level paraphrase datasets (from benchmark dataset) and split it into train–test sets containing different sentential paraphrases. However, such evaluation is less interpretable, since it is not clear exactly which words/phrases are leaned by the model. We defer this for our future work.

[1]The number of paraphrase pairs p is treated as an additional hyperparameter.

Model	10 fold accuracy %
λ-WASP (Wong and Mooney, 2007)	86.60
generative model + EM (Lu et al., 2008)	81.80
paraphrase + arg. transfer (Ray et al., 2018)	88.30
seq-to-seq + attention (our baseline)	89.77
seq-to-tree + attention (our baseline)	90.91
seq-to-seq + paraphrase attention (our model)	90.91
seq-to-tree + paraphrase attention (our model)	**92.05**

Table 3: Comparison of best 10 fold accuracy of all models on benchmark GEO dataset. In our paraphrase attention models we use $p = 150$ paraphrase pairs.

4.3 Parameters

We implemented our models using Torch 7. All baseline model hyper-parameters were tuned on validation data. To test the performance gain, our models use the same hyper-parameters as the baseline model. To compare retraining time, all models were trained/retrained on a server with NVIDIA Tesla K80 GPU. At the encoder, we initialize all embedding vectors (including OOV words) with GLOVE embeddings (Pennington et al., 2014). RMSProp was used as the optimization algorithm. We restrict the embedding dimension, and hidden state dimension $d \in \{100, 200, 300\}$. The learning rate was chosen in the range $[0.0125, 0.005]$, and dropout rates among $\{0.5, 0.4, 0.3, 0.2\}$. For paraphrase attention models we choose the number of paraphrase pairs $p \in \{50, 100, 150, 200, 250\}$. For the baseline model, we use the code made available by the authors of (Dong and Lapata, 2016).

Model	test accuracy %
online CCG (Zettlemoyer and Collins, 2007)	84.60
seq-to-seq + attn + copy (Jia and Liang, 2016)	83.30
seq-to-seq + attn (Dong and Lapata, 2016)	84.20
seq-to-tree + attn (Dong and Lapata, 2016)	84.60
seq-to-seq + attn + arg. transfer (Ray et al., 2018)	85.27
coarse2fine (Dong and Lapata, 2018)	**87.70**
seq-to-seq + attention (our baseline)	85.71
seq-to-tree + attention (our baseline)	82.59
seq-to-seq + paraphrase attention (our model)	86.16
seq-to-tree + paraphrase attention (our model)	82.37

Table 4: Comparison of best test accuracy of all models on benchmark ATIS dataset. In our paraphrase attention models we use $p = 200$ paraphrase pairs.

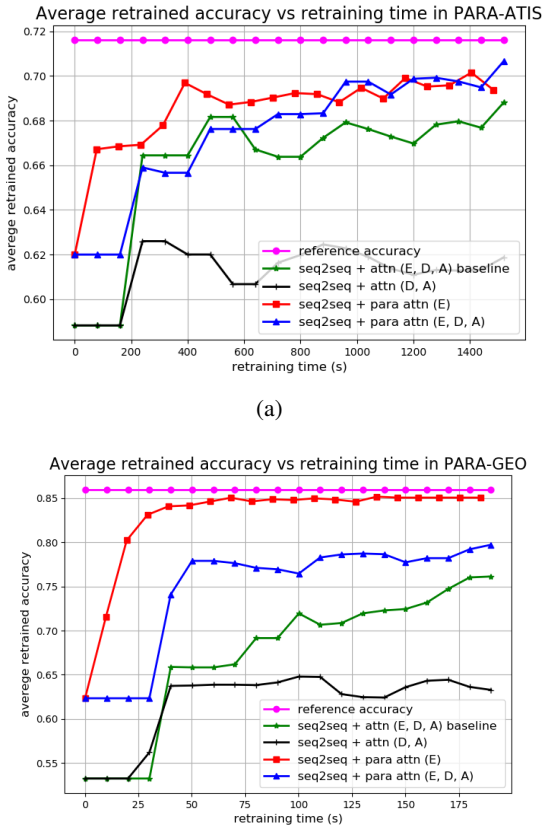


Figure 4: Comparison of the average retrained accuracy of all sequence-to-sequence based models with retraining time in (a) PARA-ATIS (b) PARA-GEO datasets. We show in brackets the part of the model being fine-tuned; where **Encoder=E**, **Decoder=D**, **Attention=A**. Our paraphrase attention model with encoder fine-tuning **seq2seq + para attn (E)**, reaches the retrained accuracy of baseline seq-to-seq + attention model 4X faster in PARA-ATIS, and 9X faster in PARA-GEO dataset.

4.4 Results

First, we compare the accuracy of our models to baseline sequence-to-sequence/tree with attention parsers (Dong and Lapata, 2016). Table 2 compares the best 10 fold accuracy achieved by all models in JOB dataset, while Table 3 compares the same in GEO dataset. For our paraphrase attention models we randomly choose $p = 50$ paraphrase pairs in JOB dataset, and $p = 150$ paraphrase pairs in GEO dataset. We observe our paraphrase attention parsers to outperform most baseline models achieving state-of-the-art 10 fold accuracy. In Table 4 we report the best test accuracy achieved in ATIS dataset. In this dataset we use $p = 200$ paraphrase pairs for our paraphrase attention models. Our sequence-to-sequence + paraphrase attention model achieves a highly competitive accuracy of 86.16% on the benchmark test set outperforming all baselines except (Dong and Lapata, 2018). We remind that, our models do not use any external data compared to baselines since the paraphrases are harnessed from the training data itself.

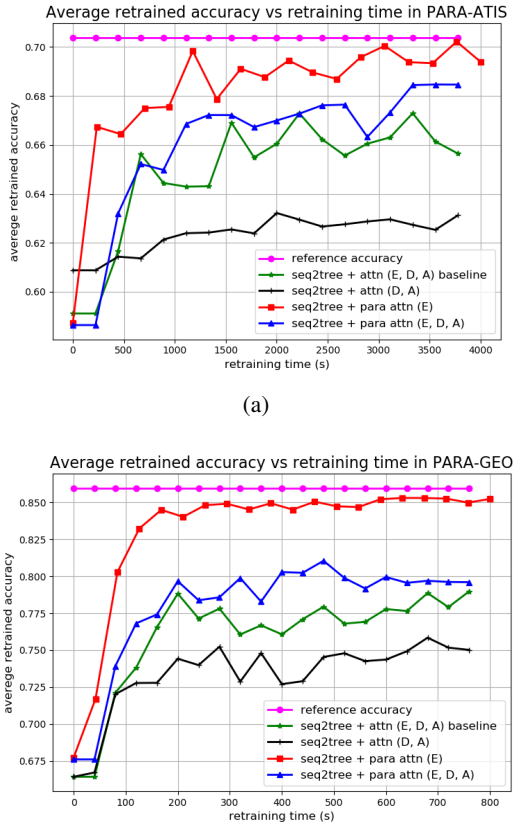
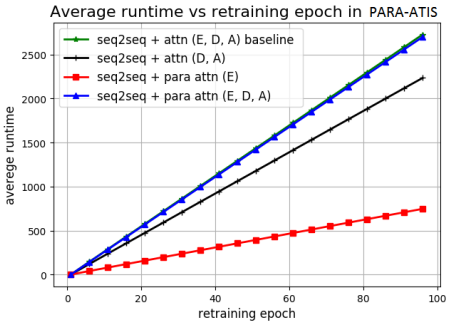


Figure 5: Comparison of the average retrained accuracy of all sequence-to-tree based models with retraining time in (a) PARA-ATIS (b) PARA-GEO datasets. We show in brackets the part of the model being fine-tuned; where **Encoder=E**, **Decoder=D**, **Attention=A**. Our paraphrase attention model with encoder fine-tuning **seq2tree + para attn (E)**, reaches the retrained accuracy of baseline seq-to-tree + attn model 3X faster in PARA-ATIS, and 5X faster in PARA-GEO dataset.

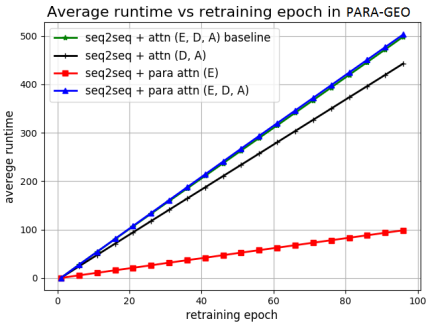
Next, we present our main domain adaptation results by comparing the retraining performance of all models using the OOV datasets. In Figure 4, we plot the average retrained accuracy versus retraining time for PARA-ATIS and PARA-GEO datasets using sequence-to-sequence based models. The average retrained accuracy is computed on the OOV test set $T_{tst}(w, s)$, and further averaged over all word–synonym pairs in this dataset. We observe that, our sequence-to-sequence + paraphrase attention model with fast encoder fine-tuning (referred as *seq2seq+para*

attn (E)), achieves the maximum retrained accuracy of baseline sequence-to-sequence + attention model (denoted as *seq2seq+attn (E,D,A)*) **4X faster** in PARA-ATIS, and **9X faster** in PARA-GEO dataset. The **reference accuracy** denotes the accuracy of the original parser $\mathcal{P}$, on the subset of test queries from which the OOV test set $T_{tst}(w, s)$ was obtained, and acts as a soft upper bound on retrained accuracy. Ideally, the fine-tuned parser $\mathcal{P}'$ should achieve retrained accuracy comparable to this target reference. In PARA-GEO dataset, our model achieves accuracy close to the reference. As discussed in Section 3, the baseline parser can also be fine-tuned faster by freezing encoder parameters, and retraining only the decoder + attention layers. This however achieves a poor retrained accuracy as shown in Figure 4 (denoted as *seq2seq+attn (D,A)*) since proper encoder representations corresponding to OOV words are not learned. In Figure 5 we compare the retraining performance of all sequence-to-tree based models. We again observe that sequence-to-tree with paraphrase attention model achieves maximum retrained accuracy of baseline model **3X faster** in PARA-ATIS, and **5X faster** in PARA-GEO dataset.

Finally, in Figure 6 we plot the average retraining time with epochs, for all sequence-to-sequence models. As expected, our paraphrase attention model with fast encoder fine-tuning (seq2seq + para attn (E)) is the fastest, and it shows a runtime speedup of 3X-4X over baseline models in both OOV datasets. When we fine-tune the entire paraphrase attention model, this too takes similar runtime as the baseline (with full model fine-tuning). When the baseline model is fine-tuned with frozen encoder parameters, it is relatively faster since the gradients need not be back-propagated to the encoder. However, as shown earlier in Figure 4, this model achieves very poor retrained accuracy. Note that, it is possible to fine-tune the baseline model with frozen decoder + attention layers, updating only encoder parameters. However, this is not expected to be significantly faster than full model fine-tuning, since it still needs to compute all decoder and attention gradients in order to back-propagate the gradients to the encoder.



(a)



(b)

Figure 6: Figure showing the average runtime of all sequence-to-sequence models with retraining epochs in (a) PARA-ATIS (b) PARA-GEO datasets. We show in brackets the part of the model being fine-tuned; where **Encoder=E**, **Decoder=D**, **Attention=A**. Our paraphrase attention model, with fast encoder fine-tuning, achieves a 3X-4X runtime speedup over baseline seq-to-seq + attention model in both dataset.

5 Conclusion

Post-deployment domain adaptation of intelligent agent to better understand user and locale specific vocabulary require frequent retraining of its semantic parser. In this paper, we propose novel paraphrase attention based sequence-to-sequence/tree models for semantic parsing, which enables near real-time domain adaptation. Our parsers can be retrained quickly by fine-tuning just the encoder network; which was not possible in previous attention based parsers. On OOV datasets our parsers are shown to achieve target retrained accuracy over 3-9X faster than baseline parsers. Moreover, by jointly learning the semantic relationship between paraphrases within the model, our parsers can achieve better or comparable parsing accuracy to previous models on benchmark datasets. Our models can also be easily adapted to transformer based sequence networks, which outperform recurrent networks for many NLP tasks (Vaswani et al., 2017; Devlin et al., 2019), as shown recently.

References

Alireza Aghasi, Afshin Abdi, Nam Nguyen, and Justin Romberg. 2017. Net-trim: Convex pruning of deep neural networks with performance guarantee. In *Proc. of NIPS, 2017*, pages 3180–3189.

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *TACL*, 1:49–62.

Amos Azaria, Jayant Krishnamurthy, and Tom M Mitchell. 2016. Instructable intelligent personal agent. In *Proc. of the 30th AAAI*, pages 2681–2689.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the ICLR, San Diego, California*.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proc. of ACL (1)*, pages 1415–1425.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. of the 2014 EMNLP*, pages 1724–1734.

Emily L. Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. 2014. Exploiting linear structure within convolutional networks for efficient evaluation. In *Annual Conference on Neural Information Processing Systems NIPS, 2014*, pages 1269–1277.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proc. of the 54th ACL 2016*.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proc. of ACL 2018, Volume 1: Long Papers*, pages 731–742.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proc. of EMNLP 2017*, pages 875–886.

Xing Fan, Emilio Monti, Lambert Mathias, and Markus Dreyer. 2017. Transfer learning for neural semantic parsing. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 48–56.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: the paraphrase database. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 758–764.

Jonathan Herzig and Jonathan Berant. 2017. Neural semantic parsing over multiple knowledge-bases. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 623–628.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. In *Proc. of the ACL 2017, Volume 1: Long Papers*, pages 963–973.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proc. of the 54th ACL*.

Shiva Prasad Kasiviswanathan, Nina Narodytska, and Hongxia Jin. 2018. Network approximation using tensor sketching. In *Proc. of the 27th IJCAI 2018*, pages 2319–2325.

Young-Bum Kim, Dongchan Kim, Anjishnu Kumar, and Ruhi Sarikaya. 2018. Efficient large-scale neural domain classification with personalized attention. In *Proc. of the ACL 2018, Volume 1: Long Papers*, pages 2214–2224.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic ccg grammars from logical form with higher-order unification. In *Proc. of the 2010 EMNLP*, pages 1223–1233. ACL.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in ccg grammar induction for semantic parsing. In *Proc. of EMNLP, 2011*, pages 1512–1523. ACL.

Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *Proc. of EMNLP 2008*, pages 783–792.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP 2014*, pages 1532–1543.

Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek V. Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual LSTM networks. In *Proc. of COLING 2016*, pages 2923–2934.

Avik Ray, Yilin Shen, and Hongxia Jin. 2018. Learning out-of-vocabulary words in intelligent personal

agents. In *Proc. of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 4309–4315.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proc. of NIPS*, pages 3104–3112.

Lappoon R. Tang and Raymond J. Mooney. 2001. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Proc. of EMCL 2001*, pages 466–477.

Jesse Thomason, Shiqi Zhang, Raymond J. Mooney, and Peter Stone. 2015. Learning to interpret natural language commands through human-robot dialog. In *Proc. of IJCAI 2015,*, pages 1923–1929.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015a. Pointer networks. In *Proc. of NIPS*, pages 2692–2700.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. 2015b. Grammar as a foreign language. In *Proc. of NIPS*, pages 2773–2781.

Yuk Wah Wong and Raymond J Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *ACL*, volume 45, page 960.

John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055.

Luke S Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proc. of the 21st UAI*, pages 658–666.

Luke S Zettlemoyer and Michael Collins. 2007. Online learning of relaxed ccg grammars for parsing to logical form. In *EMNLP-CoNLL*, pages 678–687.