Cross-domain Semantic Parsing via Paraphrasing

Yu Su
Department of Computer Science
University of California, Santa Barbara
ysu@cs.ucsb.edu

Xifeng Yan
Department of Computer Science
University of California, Santa Barbara
xyan@cs.ucsb.edu

arXiv:1704.05974v1 [cs.CL] 20 Apr 2017

Abstract

Existing studies on semantic parsing mainly focus on the in-domain setting. We formulate cross-domain semantic parsing as a domain adaptation problem: train a semantic parser on some source domains and then adapt it to the target domain. Due to the diversity of logical forms in different domains, this problem presents unique and intriguing challenges. By converting logical forms into canonical utterances in natural language, we reduce semantic parsing to paraphrasing, and develop an attentive sequence-to-sequence paraphrase model that is general and flexible to adapt to different domains. We discover two problems, *small micro variance* and *large macro variance*, of pre-trained word embeddings that hurdle their direct use in neural networks, and propose standardization techniques as a remedy. On the popular OVERNIGHT dataset, which contains eight domains, we show that both cross-domain training and standardized pre-trained word embeddings can bring significant improvement.

1 Introduction

Semantic parsing, which maps natural language utterances into computer-understandable logical forms, has drawn substantial attention recently as a promising direction for developing natural language interfaces to computers. Semantic parsing has been applied in many domains, including querying data/knowledge bases (Woods, 1973; Zelle and Mooney, 1996; Berant et al., 2013), controlling Internet-of-Things (IoT) devices (Campagna et al., 2017), and communicating with robots (Chen and Mooney, 2011; Tellex et al., 2011; Artzi and Zettlemoyer, 2013; Bisk et al., 2016).

Despite the wide applications, studies on semantic parsing have mainly focused on the *in-domain* setting, where both training and testing data are drawn from the same domain. How to build semantic parsers that can learn across domains remains an under-addressed problem. In this work, we study *cross-domain semantic parsing*. We model it as a domain adaptation problem (Daumé III and Marcu, 2006), where we are given some *source* domains and a *target* domain, and the core task is to adapt a semantic parser trained on the source domains to the target domain (Figure 1). The benefits are two-fold: (1) by training on the source domains, the cost of collecting training data for the target domain can be reduced, and (2) the data of source domains may provide information complementary to the data collected for the target domain, leading to better performance on the target domain.

This is a very challenging task. Traditional domain adaptation (Daumé III and Marcu, 2006; Blitzer et al., 2006) only concerns natural languages, while semantic parsing concerns both natural and formal languages. Different domains often involve different predicates. In Figure 1, from the source BASKETBALL domain a semantic parser can learn the semantic mapping from natural language to predicates like `team` and `season`, but in the target SOCIAL domain it needs to handle predicates like `employer` instead. Worse still, even for the same predicate, it is legitimate to use arbitrarily different predicate symbols, e.g., other symbols like `hired_by` or even `predicate1` can also be used for the `employer` predicate. Therefore, directly transferring the mapping from natural language to predicate symbols learned from source domains to the target domain may not be much beneficial.

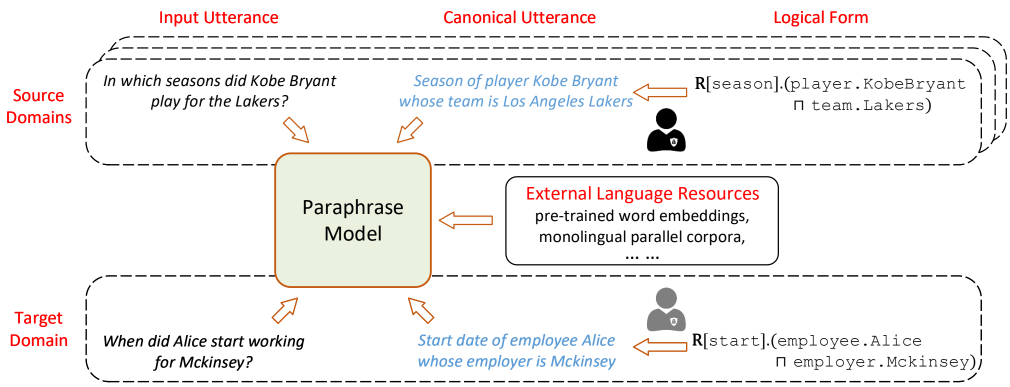

Figure 1: Cross-domain semantic parsing via paraphrasing framework. In a deterministic way, logical forms are first converted into canonical utterances in natural language. A paraphrase model then learns from the source domains and adapts to the target domain. External language resources can be incorporated in a consistent way across domains.

Inspired by the recent success of paraphrasing based semantic parsing (Berant and Liang, 2014; Wang et al., 2015), we propose to use natural language as an intermediate representation for cross-domain semantic parsing. As shown in Figure 1, logical forms are converted into canonical utterances in natural language, and semantic parsing is reduced to paraphrasing. It is the knowledge of paraphrasing, at lexical, syntactic, and semantic levels, that will be transferred across domains.

Still, adapting a paraphrase model to a new domain is a challenging and under-addressed problem. To give some idea of the difficulty, for each of the eight domains in the popular OVERNIGHT (Wang et al., 2015) dataset, 30% to 55% of the words never occur in any of the other domains, a similar problem observed in domain adaptation for machine translation (Daumé III and Jagarlamudi, 2011). The paraphrase model therefore can get little knowledge for a substantial portion of the target domain from the source domains. We introduce pre-trained word embeddings such as WORD2VEC (Mikolov et al., 2013) to combat the vocabulary variety across domains. Based on recent studies on neural network initialization, we conduct a statistical analysis of pre-trained word embeddings and discover two problems that may hurdle their direct use in neural networks: *small micro variance*, which hurts optimization, and *large macro variance*, which hurts generalization. We propose to *standardize* pre-trained word embeddings, and show its advantages both analytically and experimentally.

On the OVERNIGHT dataset, we show that cross-domain training under the proposed framework can significantly improve model performance. We also show that, compared with directly using pre-trained word embeddings or normalization as in previous work, the proposed standardization technique can lead to about 10% absolute improvement in accuracy.

2 Cross-domain Semantic Parsing

2.1 Problem Definition

Unless otherwise stated, we will use u to denote input utterance, c for canonical utterance, and z for logical form. We denote $\mathcal{U}$ as the set of all possible utterances. For a domain, suppose $\mathcal{Z}$ is the set of logical forms, a semantic parser is a mapping $f: \mathcal{U} \rightarrow \mathcal{Z}$ that maps every input utterance to a logical form (a `null` logical form can be included in $\mathcal{Z}$ to reject out-of-domain utterances).

In cross-domain semantic parsing, we assume there are a set of K source domains $\{\mathcal{Z}_i\}_{i=1}^K$, each with a set of training examples $\{(u_j^i, z_j^i)\}_{j=1}^{N_i}$. It is in principle advantageous to model the source domains separately (Daumé III and Marcu, 2006), which retrains the possibility of separating domain-general information from domain-specific information, and only transferring the former to the target domain. For simplicity, here we merge the source domains into a single domain $\mathcal{Z}_s$ with training data $\{(u_i, z_i)\}_{i=1}^{N_s}$. The task is to learn a semantic parser $f: \mathcal{U} \rightarrow \mathcal{Z}_t$ for a target domain $\mathcal{Z}_t$, for which we have a set of training examples $\{(u_i, z_i)\}_{i=1}^{N_t}$. Some characteristics can be summarized as follows:

- $\mathcal{Z}_t$ and $\mathcal{Z}_s$ can be totally disjoint.
- The input utterance distribution of the source and the target domain can be independent and differ remarkably.
- Typically $N_t \ll N_s$.

In the most general and challenging case, $\mathcal{Z}_t$ and $\mathcal{Z}_s$ can be defined using different formal languages. Because of the lack of relevant datasets, here we restrain ourselves to the case where $\mathcal{Z}_t$ and $\mathcal{Z}_s$ are defined using the same formal language, e.g., λ-DCS (Liang, 2013) as in the OVERNIGHT dataset.

2.2 Framework

Our framework follows the research line of semantic parsing via paraphrasing (Berant and Liang, 2014; Wang et al., 2015). While previous work focuses on the in-domain setting, we discuss its applicability and advantages in the cross-domain setting, and develop techniques to address the emerging challenges in the new setting.

Canonical utterance. We assume a *one-to-one* mapping $g\colon \mathcal{Z} \rightarrow \mathcal{C}$, where $\mathcal{C} \subset \mathcal{U}$ is the set of canonical utterances. In other words, every logical form will be converted into a unique canonical utterance deterministically (Figure 1). Previous work (Wang et al., 2015) has demonstrated how to design such a mapping, where a domain-general grammar and a domain-specific lexicon are constructed to automatically convert every logical form to a canonical utterance. In this work, we assume the mapping is given[1], and focus on the subsequent paraphrasing and domain adaptation problems.

This design choice worths some discussion. The grammar, or at least the lexicon for mapping predicates to natural language, needs to be provided by domain administrators. This indeed brings an additional cost, but we believe it is reasonable and even necessary for three reasons: (1) Only domain administrators know the predicate semantics the best, so it has to be them to reveal that by grounding the predicates to natural language. (2) Otherwise, predicate semantics can only be learned from supervised training data of each domain, bringing a significant cost on data collection. (3) Canonical utterances are understandable by average users, and thus can also be used for training data collection via crowdsourcing (Wang et al., 2015; Su et al., 2016), which can amortize the cost.

Take comparatives as an example. In logical forms, comparatives can be legitimately defined using arbitrarily different predicates in different domains, e.g., `<`, `smallerInSize`, or even predicates with an ambiguous surface form, like `lt`. When converting logical form to canonical utterance, however, domain administrators have to choose common natural language expressions like "*less than*" and "*smaller*", providing a shared ground for cross-domain semantic parsing.

Paraphrase model. In the previous work based on paraphrasing (Berant and Liang, 2014; Wang et al., 2015), semantic parsers are implemented as log-linear models with hand-engineered domain-specific features (including paraphrase features). Considering the recent success of representation learning for domain adaptation (Glorot et al., 2011; Chen et al., 2012), we propose a paraphrase model based on the sequence-to-sequence (Seq2Seq) model (Sutskever et al., 2014), which can be trained end to end without feature engineering. We show that it outperforms the previous log-linear models by a large margin in the in-domain setting, and can easily adapt to new domains.

Pre-trained word embeddings. An advantage of reducing semantic parsing to paraphrasing is that external language resources become easier to incorporate. Observing the vocabulary variety across domains, we introduce pre-trained word embeddings to facilitate domain adaptation. For the example in Figure 1, the paraphrase model may have learned the mapping from "play for" to "whose team is" in a source domain. By acquiring word similarities ("play"-"work" and "team"-"employer") from pre-trained word embeddings, it can establish the mapping from "work for" to "whose employer is" in the target domain, even without in-domain training data. We analyze statistical characteristics of the pre-trained word embeddings, and propose standardization techniques to remedy some undesired characteristics, which hurdle their direct use in neural networks.

Domain adaptation protocol. We will use the following protocol: (1) train a paraphrase model using the data of the source domain, (2) use the learned parameters to initialize a model in the target domain, and (3) fine-tune the model using the training data of the target domain.

2.3 Prior Work

While most studies on semantic parsing so far have focused on the in-domain setting, there are a number of studies of particular relevance to this

[1] In the experiments we use the provided canonical utterances of the OVERNIGHT dataset.

work. In the recent efforts of scaling semantic parsing to large knowledge bases like Freebase (Bollacker et al., 2008), researchers have explored several ways to infer the semantics of knowledge base relations unseen in training, which are often based on at least one (often both) of the following assumptions: (1) *Distant supervision*. Freebase entities can be linked to external text corpora, and serve as anchors for seeking semantics of Freebase relations from text. For example, Cai and Alexander (2013), among others (Berant et al., 2013; Xu et al., 2016), use sentences from Wikipedia that contain any entity pair of a Freebase relation as the support set of the relation. (2) *Self-explaining predicate symbols*. Most Freebase relations are described using a carefully chosen symbol (surface form), e.g., `place_of_birth`, which provides strong cues for their semantics. For example, Yih et al. (2015) directly compute the similarity of input utterance and the surface form of Freebase relations via a convolutional neural network. Kwiatkowski et al. (2013) also extract lexical features from input utterance and the surface form of entities and relations. They have actually evaluated their model on Freebase subdomains not covered in training, and have shown impressive results. However, in the more general setting of cross-domain semantic parsing, we may have neither of these luxuries. Distant supervision may not be available (e.g., IoT devices involving no entities but actions), and predicate symbols may not provide enough cues (e.g., `predicate1`). In this case, seeking additional inputs from domain administrators is probably necessary.

In parallel of this work, Herzig and Berant (2017) have explored another direction of semantic parsing with multiple domains, where they use all the domains to train a single semantic parser, and attach a domain-specific encoding to the training data of each domain to help the semantic parser differentiate between domains. We pursue a different direction: we train a semantic parser on some source domains and adapt it to the target domain. Another difference is that their work directly maps utterances to logical forms, while ours is based on paraphrasing.

Cross-domain semantic parsing can be seen as a way to reduce the cost of training data collection, which resonates with the recent trend in semantic parsing. Berant et al. (2013) propose to learn from utterance-denotation pairs instead of utterance-logical form pairs, while Wang et al. (2015) and Su et al. (2016) manage to employ crowd workers with no linguistic expertise for data collection. Jia and Liang (2016) propose an interesting form of data augmentation. They learn a grammar from existing training data, and generate new examples from the grammar by recombining segments from different examples.

We use natural language as an intermediate representation to transfer knowledge across domains, and assume the mapping from the intermediate representation (canonical utterance) to logical form can be done deterministically. Several other intermediate representations have also been used, such as Combinatory Categorial Grammars (Kwiatkowski et al., 2013; Reddy et al., 2014), dependency tree (Reddy et al., 2016, 2017), and semantic role structure (Goldwasser and Roth, 2013). But their main aim is to better represent input utterances with a richer structure. A separate ontology matching step is needed to map the intermediate representation to logical form, which requires domain-dependent training.

A number of other related studies have also used paraphrasing. For example, Fader et al. (2013) leverage question paraphrases to for question answering, while Narayan et al. (2016) generate paraphrases as a way of data augmentation.

Cross-domain semantic parsing can greatly benefit from the rich literature of domain adaptation and transfer learning (Daumé III and Marcu, 2006; Blitzer et al., 2006; Pan and Yang, 2010; Glorot et al., 2011). For example, Chelba and Acero (2004) use parameters trained in the source domain as prior to regularize parameters in the target domain. The feature augmentation technique from Daumé III (2009) can be very helpful when there are multiple source domains. We expect to see many of these ideas to be applied in the future.

3 Paraphrase Model

In this section we propose a paraphrase model based on the Seq2Seq model (Sutskever et al., 2014). Similar models have been used in semantic parsing (Jia and Liang, 2016; Dong and Lapata, 2016) but for directly mapping utterances to logical forms. We demonstrate that it can also be used as a paraphrase model for semantic parsing. Several other neural models have been proposed for paraphrasing (Socher et al., 2011; Hu et al., 2014; Yin and Schütze, 2015), but it is not the focus of this work to compare all the alternatives.

For an input utterance $u = (u_1, u_2, \ldots, u_m)$ and an output canonical utterance $c = (c_1, c_2, \ldots, c_n)$, the model estimates the conditional probability $p(c|u) = \prod_{j=1}^{n} p(c_j|u, c_{1:j-1})$. The tokens are first converted into vectors via a word embedding layer ϕ. The initialization of the word embedding layer is critical for domain adaptation, which we will further discuss in Section 4.

The *encoder*, which is implemented as a bi-directional recurrent neural network (RNN), first encodes u into a sequence of state vectors $(h_1, h_2, \ldots, h_m)$. The state vectors of the forward RNN and the backward RNN are respectively computed as:

$$\overrightarrow{h}_i = GRU_{fw}(\phi(u_i), \overrightarrow{h}_{i-1})$$
$$\overleftarrow{h}_i = GRU_{bw}(\phi(u_i), \overleftarrow{h}_{i+1})$$

where gated recurrent unit (GRU) as defined in (Cho et al., 2014) is used as the recurrence. We then concatenate the forward and backward state vectors, $h_i = [\overrightarrow{h}_i, \overleftarrow{h}_i], i = 1, \ldots, m$.

We use an attentive RNN as the *decoder*, which will generate the output tokens one at a time. We denote the state vectors of the decoder RNN as $(d_1, d_2, \ldots, d_n)$. The attention takes a form similar to (Vinyals et al., 2015). For the decoding step j, the decoder is defined as follows:

$$\begin{aligned}
d_0 &= \tanh(W_0[\overrightarrow{h}_m, \overleftarrow{h}_1]) \\
u_{ji} &= v^T \tanh(W_1 h_i + W_2 d_j) \\
\alpha_{ji} &= \frac{u_{ji}}{\sum_{i'=1}^{m} u_{ji'}} \\
h'_j &= \sum_{i=1}^{m} \alpha_{ji} h_i \\
d_{j+1} &= GRU([\phi(c_j), h'_j], d_j) \\
p(c_j|u, c_{1:j-1}) &\propto \exp(U[d_j, h'_j])
\end{aligned}$$

where W_0, W_1, W_2, v and U are model parameters. The decoder first calculate normalized attention weights α_{ji} over encoder states, and get a summary state h'_j. The summary state is then used to calculate the next decoder state d_{j+1} and the output probability distribution $p(c_j|u, c_{1:j-1})$.

Training. Given a set of training examples $\{(u_i, c_i)\}_{i=1}^{N}$, we minimize the cross-entropy loss $-\frac{1}{N}\sum_{i=1}^{N} \log p(c_i|u_i)$, which maximizes the log probability of the correct canonical utterances. We apply dropout (Hinton et al., 2012) on both input and output of the GRU cells to prevent overfitting.

Testing. Given a domain $\{\mathcal{Z}, \mathcal{C}\}$, there are two ways to use a trained model. One is to use it to *generate* the most likely output utterance u' given an input utterance u (Sutskever et al., 2014),

$$u' = \arg\max_{u' \in \mathcal{U}} p(u'|u).$$

In this case u' can be any utterance permissable by the output vocabulary, and may not necessarily be a legitimate canonical utterance in $\mathcal{C}$. This is more suitable for large domains with a lot of logical forms, like Freebase. An alternative way is to use the model to *rank* the legitimate canonical utterances (Kannan et al., 2016):

$$c = \arg\max_{c \in \mathcal{C}} p(c|u),$$

which is more suitable for small domains having a limited number of logical forms, like the ones in the OVERNIGHT dataset. We will adopt the second strategy. It is also very challenging; random guessing will give almost zero accuracy. It is also possible to first find a smaller set of candidates to rank via beam search (Berant et al., 2013; Wang et al., 2015).

4 Pre-trained Word Embeddings for Domain Adaptation

Pre-trained word embeddings like WORD2VEC have a great potential to combat the vocabulary variety across domains. For example, we can use WORD2VEC embeddings to initialize the word embedding layer of the source domain, with the hope that the other parameters in the model will co-adapt with the word embeddings during training in the source domain, and generalize better to the out-of-vocabulary words (but covered by WORD2VEC) in the target domain.

However, a statistical analysis of the WORD2VEC embeddings shows that it might not be a good idea to directly use them in a deep neural network, and proper *standardization* is needed. Our analysis will be based on the 300-dimensional WORD2VEC embeddings trained on the 100B-word Google News corpus[2]. It contains 3 million words, leading to a 3M-by-300 word embedding matrix.

Neural networks are very sensitive to initialization (Erhan et al., 2010). The "rule of thumb" to randomly initialize word embeddings in neural networks is to sample from a uniform or Gaussian distribution with *unit variance*, which works

[2]https://code.google.com/archive/p/word2vec/

Initialization	L2 norm	Micro Variance	Cosine Sim.
Random	17.3 ± 0.45	1.00 ± 0.05	0.00 ± 0.06
WORD2VEC	2.04 ± 1.08	0.02 ± 0.02	0.13 ± 0.11
WORD2VEC + ES	17.3 ± 0.05	1.00 ± 0.00	0.13 ± 0.11
WORD2VEC + FS	16.0 ± 8.47	1.09 ± 1.31	0.12 ± 0.10
WORD2VEC + EN	1.00 ± 0.00	0.01 ± 0.00	0.13 ± 0.11

Table 1: Word embedding initializations. Random: random sampling from $U(-\sqrt{3}, \sqrt{3})$, thus unit variance. WORD2VEC: raw WORD2VEC embeddings. ES: per-example standardization. FS: per-feature standardization. EN: per-example normalization. Cosine similarity is computed on a random (but fixed) set of 1M word pairs.

well for a wide range of neural network models in general. We therefore use it as a reference to compare different word embedding initializations. We report the L2 norm and the per-example (per-row) variance of different initializations in Table 1. The statistics show why using the raw WORD2VEC embeddings may hurt model performance. Compared with random initialization, both the L2 norm and the per-example variance (denoted as *micro variance*) of the WORD2VEC embeddings are much smaller, while the variance of the embedding of different words (denoted as *macro variance*) is much larger (the maximum and the minimum L2 norm are 21.1 and 0.015, respectively). Small micro variance can make the variance of neuron activations starts off too small[3], implying a poor starting point in the parameter space. On the other hand, large macro variance may make a model hard to generalize to words unseen in training.

Based on the above analysis, we propose to do *unit variance standardization* (standardization for short) on the pre-trained word embeddings. There are two possible ways, *per-example standardization*, which standardizes each row of the embedding matrix to unit variance, and *per-feature standardization*, which standardizes each column instead. We do not make the rows or columns zero mean. Per-example standardization enjoys the goodness of both random initialization and pre-trained word embeddings: it fixes the small micro variance problem as well as the large macro variance problem of pre-trained word embeddings, while still preserving cosine similarity, i.e., word similarity. Per-feature standardization does not preserve cosine similarity, nor does it fix the large macro variance problem. However, it enjoys the benefit of *global* statistics, in contrast to the *local* statistics of each word embedding used in per-example standardization. Therefore, in problems where the testing and training vocabularies are similar, per-feature standardization may be more advantageous. Both standardizations lose vector length information. Levy et al. (2015) have suggested *per-example normalization*[4] of pre-trained word embeddings for lexical tasks like word similarity and analogy, which do no involve neural networks. Making the word embeddings unit length alleviates the large macro variance problem, but the small micro variance problem remains.

This is indeed a pretty simple trick, and per-feature standardization (with zero mean) is also a standard data preprocessing method. However, it is not self-evident that this kind of standardization shall be applied on pre-trained word embeddings before using them in neural networks, especially with the obvious downside of rendering the word embedding algorithm's loss function sub-optimal.

We expect this to be less of a issue for large-scale problems with a large vocabulary and abundant training examples. For example, Vinyals et al. (2015) have found that directly using raw WORD2VEC embeddings for initialization can bring a consistent, though small, improvement in neural constituency parsing. However, for smaller-scale problems (e.g., an application domain of semantic parsing can have a vocabulary size of only a few hundreds), this issue becomes more critical. Initialized with the raw pre-trained embeddings, a model may quickly fall into a poor local optimum and may not have enough signal to escape. Because of the large macro variance of the raw word embeddings, standardization is critical for domain adaptation, which needs to generalize to many words unseen in training.

5 Evaluation

5.1 Data Analysis

The OVERNIGHT dataset (Wang et al., 2015) contains 8 different domains. Each domain is based on a separate knowledge base, with logical forms written in λ-DCS (Liang, 2013). Logical forms are converted into canonical utterances via a simple grammar, and the input utterances are collected by asking crowd workers to paraphrase the canon-

[3]Under some conditions, including using Xavier initialization (also introduced in that paper and now widely used) for weights, Glorot and Bengio (2010) have shown that the activation variances in a feedforward neural network will be roughly the same as the input variances (word embeddings here) at the beginning of training.

[4]It can also be found in the implementation of Glove (Pennington et al., 2014): https://github.com/stanfordnlp/GloVe

Metric	CALENDAR	BLOCKS	HOUSING	RESTAURANTS	PUBLICATIONS	RECIPES	SOCIAL	BASKETBALL
# of example (N)	837	1995	941	1657	801	1080	4419	1952
# of logical form ($\|\mathcal{Z}\|$, $\|\mathcal{C}\|$)	196	469	231	339	149	124	624	252
vocab. size ($\|\mathcal{V}\|$)	228	227	318	342	203	256	533	360
% $\in$ other domains	71.1	61.7	60.7	55.8	65.6	71.9	46.0	45.6
% $\in$ WORD2VEC	91.2	91.6	88.4	88.6	91.1	93.8	86.9	86.9
% $\in$ other domains + WORD2VEC	**93.9**	**93.8**	**90.9**	**90.4**	**95.6**	**97.3**	**89.3**	**89.4**

Table 2: Statistics of the domains in the OVERNIGHT dataset. Pre-trained word embeddings cover most of the words in each domain, paving a way for domain adaptation.

Method	CALENDAR	BLOCKS	HOUSING	RESTAURANTS	PUBLICATIONS	RECIPES	SOCIAL	BASKETBALL	Avg.
Previous Methods									
Wang et al. (2015)	74.4	41.9	54.0	75.9	59.0	70.8	48.2	46.3	58.8
Xiao et al. (2016)	75.0	55.6	61.9	80.1	75.8	–	80.0	80.5	72.7
Jia and Liang (2016)	78.0	58.1	71.4	76.2	76.4	79.6	81.4	85.2	75.8
Herzig and Berant (2017)	82.1	**62.7**	78.3	82.2	**80.7**	82.9	81.7	86.2	79.6
Our Methods									
Random + I	75.6	60.2	67.2	77.7	77.6	80.1	80.7	86.5	75.7
Random + X	79.2	54.9	74.1	76.2	78.5	82.4	82.5	86.7	76.9
WORD2VEC + I	67.9	59.4	52.4	75.0	64.0	73.2	77.0	87.5	69.5
WORD2VEC + X	78.0	54.4	63.0	81.3	74.5	83.3	81.5	83.1	74.9
WORD2VEC + EN + I	63.1	56.1	60.3	75.3	65.2	69.0	76.4	81.8	68.4
WORD2VEC + EN + X	78.0	52.6	63.5	74.7	65.2	80.6	79.9	80.8	71.2
WORD2VEC + FS + I	78.6	62.2	67.7	78.6	75.8	85.7	81.3	86.7	77.1
WORD2VEC + FS + X	**82.7**	59.4	75.1	80.4	78.9	85.2	81.8	87.2	78.9
WORD2VEC + ES + I	79.8	60.2	71.4	81.6	78.9	84.7	82.9	86.2	78.2
WORD2VEC + ES + X	82.1	62.2	**78.8**	**83.7**	80.1	**86.1**	**83.1**	**88.2**	**80.6**

Table 3: Main experiment results. I: in-domain, X: cross-domain, EN: per-example normalization, FS: per-feature standardization, ES: per-example standardization.

ical utterances. Different domains are designed to stress different types of linguistic phenomena. For example, the CALENDAR domain requires a semantic parser to handle temporal language like "*meetings that start after 10 am*", while the BLOCKS domain features spatial language like "*which block is above block 1*".

Vocabularies vary remarkably across domains (Table 2). For each domain, only 45% to 70% of the words are covered by any of the other 7 domains. A model has to learn the out-of-vocabulary words from scratch using in-domain training data. Pre-trained word embeddings cover most of the words of each domain, and thus can connect the domains to facilitate domain adaptation. Words that are still missing are mainly stop words and typos, e.g., "*ealiest*".

5.2 Experiment Setup

We compare our model with all the previous methods evaluated on the OVERNIGHT dataset. In the original OVERNIGHT paper, Wang et al. (2015) use a log-linear model with a rich set of features, including paraphrase features derived from PPDB (Ganitkevitch et al., 2013), to rank logical forms. Xiao et al. (2016) use a multi-layer perceptron to encode the unigrams and bigrams of the input utterance, and then use a RNN to predict the derivation sequence of a logical form under a grammar. Jia and Liang (2016) also use a Seq2Seq model with bi-directional RNN encoder and attentive decoder, but are to predict linearized logical forms. They also propose a data augmentation technique, which further improves the average accuracy to 77.5%. But it is orthogonal to this work and can be incorporated in any model including ours, therefore not included.

The above methods are all based on the in-domain setting, where a separate parser is trained for each domain. In parallel of this work, Herzig and Berant (2017) have explored another direction of cross-domain training: they use all of the domains to train a single parser, with a special domain encoding to help differentiate between domains. We instead model it as a domain adaptation problem, where the source and the target domain training are separate. Their model is the same as Jia and Liang (2016). It is the current best-performing method on the OVERNIGHT dataset.

We use the standard 80%/20% split of training and testing, and randomly hold out 20% of training for validation. Hyper-parameters are selected based on the validation set. In cross-domain experiments, for each target domain, all the other domains are combined as the source domain. See Appendix A for hyper-parameter settings. The evaluation metric is accuracy, i.e., the proportion of testing examples for which the top prediction yields the correct denotation. Our model is implemented in Tensorflow (Abadi et al., 2016), and the code will be released at `https://github.com/ysu1989/CrossSemparse`.

5.3 Experiment Results

5.3.1 Comparison with Previous Methods

The main experiment results are shown in Table 3. Our base model (Random + I) achieves an accuracy comparable to the previous best in-domain model (Jia and Liang, 2016). With our main novelties, cross-domain training and word embedding standardization, our full model is able to outperform the previous best model, and achieve the best accuracy on 6 out of the 8 domains. Next we examine the novelties separately.

5.3.2 Word Embedding Initialization

The in-domain results clearly show the sensitivity of model performance to word embedding initialization. Directly using the raw WORD2VEC embeddings or with per-example normalization, the performance is significantly worse than random initialization (6.2% and 7.3%, respectively). Based on the previous analyses, however, one should not be too surprised. The small micro variance problem hurts optimization. In sharp contrast, both of the proposed standardization techniques lead to better in-domain performance than random initialization (1.4% and 2.5%, respectively), setting a new best in-domain accuracy on OVERNIGHT.

5.3.3 Cross-domain Training

A consistent improvement from cross-domain training is observed across all word embedding initialization strategies. Even for raw WORD2VEC embeddings or per-example normalization, cross-domain training is able to alleviate the poor initialization. The best results are again obtained with standardization. As we have discussed before, per-feature standardization does not resolve the large macro variance problem, which may make a model harder to generalize to words unseen in training. The results provide an empirical evidence for this hypothesis. The gain of cross-domain training is more significant with per-example standardization (2.4%) than with per-feature standardization (1.8%). The seemingly radical per-example standardization works best for domain adaptation.

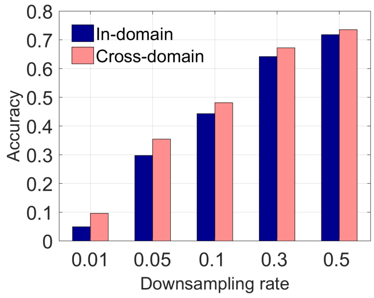

Figure 2: Results with downsampled in-domain training data. The experiment with each downsampling rate is repeated for 3 times and average results are reported. For simplicity, we only report the average accuracy over all domains. Pre-trained word embeddings with per-example standardization is used for both methods.

5.3.4 Using Downsampled Training Data

Considering the number of logical forms, the in-domain training data in the OVERNIGHT dataset is indeed abundant. In cross-domain semantic parsing, we are more interested in the scenario where there is insufficient training data for the target domain. To emulate this scenario, we downsample the in-domain training data of each target domain, but still use all training data from the source domain (thus $N_t \ll N_s$). The results are shown in Figure 2. The gain of cross-domain training is most significant when in-domain training data is scarce. As we collect more in-domain training data, the gain becomes smaller, which is expected.

6 Conclusion

We proposed a paraphrasing based framework for cross-domain semantic parsing. With a sequence-to-sequence paraphrase model, we showed that cross-domain training of semantic parsing can be quite effective. We also studied how to properly standardize pre-trained word embeddings in neural networks, especially for domain adaptation.

This work opens up a number of future directions. As discussed in Section 2.3, many conventional domain adaptation and representation learning ideas can find application in cross-domain semantic parsing. In addition to pre-trained word embeddings, other language resources like paraphrase corpora (Ganitkevitch et al., 2013) can be incorporated into the paraphrase model to further facilitate domain adaptation. We have restrained ourselves to the case where domains are defined using the same formal language, and we look forward to evaluating the framework on domains of different formal languages when such datasets with canonical utterances become available.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv:1603.04467 [cs.DC]* .

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics* 1:49–62.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Yonatan Bisk, Deniz Yuret, and Daniel Marcu. 2016. Natural language communication with robots. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International conference on Management of data*.

Qingqing Cai and Alexander Yates. 2013. Semantic parsing freebase: Towards open-domain semantic parsing. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*.

Giovanni Campagna, Rakesh Ramesh, Silei Xu, Michael Fischer, and Monica S Lam. 2017. Almond: The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant. In *Proceedings of the International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pages 341–350.

Ciprian Chelba and Alex Acero. 2004. Adaptation of maximum entropy capitalizer: Little data can help a lot. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

David L Chen and Raymond J Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the International Conference on Machine Learning*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv:1406.1078 [cs.CL]* .

Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv:0907.1815 [cs.LG]* .

Hal Daumé III and Jagadeesh Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Hal Daumé III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research* 26:101–126.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11(Feb):625–660.

Anthony Fader, Luke S Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the International Conference on Machine Learning*.

Dan Goldwasser and Dan Roth. 2013. Leveraging domain-independent information in semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Jonathan Herzig and Jonathan Berant. 2017. Neural semantic parsing over multiple knowledge-bases. *arXiv:1702.01569 [cs.CL]* .

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580 [cs.NE]* .

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Proceedings of the Annual Conference on Neural Information Processing Systems*.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, László Lukács, Marina Ganea, Peter Young, et al. 2016. Smart reply: Automated response suggestion for email. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics* 3:211–225.

Percy Liang. 2013. Lambda dependency-based compositional semantics. *arXiv:1309.4408 [cs.AI]* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Annual Conference on Neural Information Processing Systems*.

Shashi Narayan, Siva Reddy, and Shay B Cohen. 2016. Paraphrase generation from latent-variable PCFGs for semantic parsing. *arXiv:1601.06068 [cs.CL]* .

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10):1345–1359.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics* 2:377–392.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics* 4:127–140.

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. *arXiv:1702.03196 [cs.CL]* .

Richard Socher, Eric H Huang, Jeffrey Pennington, Andrew Y Ng, and Christopher D Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proceedings of the Annual Conference on Neural Information Processing Systems*.

Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gür, Zenghui Yan, and Xifeng Yan. 2016. On generating characteristic-rich question sets for QA evaluation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Proceedings of the Annual Conference on Neural Information Processing Systems* .

Stefanie A Tellex, Thomas Fleming Kollar, Steven R Dickerson, Matthew R Walter, Ashis Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Proceedings of the Annual Conference on Neural Information Processing Systems*.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

William A Woods. 1973. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the American Federation of Information Processing Societies Conference*.

Chunyang Xiao, Marc Dymetman, and Claire Gardent. 2016. Sequence-based structured prediction for semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. *Proceedings of the Annual Meeting of the Association for Computational Linguistics* .

Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Wenpeng Yin and Hinrich Schütze. 2015. Multi-GranCNN: An architecture for general matching of text chunks on multiple levels of granularity. In *ACL*.

John M Zelle and Raymon J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Hyper-parameter Setting

We do model selection based on the validation set. State size of both the encoder and the decoder are set to 100, and word embedding size is set to 300. Input dropout rate of the GRU cells is 0.7, and output dropout rate is 0.5. Mini-batch size is 512. We use Adam for optimization, which we find works slightly but consistently better than other popular optimizers like RMSprop and Adadelta. We use the default parameters for Adam as suggested in the paper. We use gradient clipping with a cap for global norm at 5.0 to alleviate the exploding gradients problem of recurrent neural networks. Early stopping based on the validation set is used to decide when to stop training. For each experiment, we do the training for 3 times and then test the model with the best validation performance.