Domain Adaptation for Low-Resource Neural Semantic Parsing

Alvin Kennardi[1], **Gabriela Ferraro**[1,2], and **Qing Wang**[1]

[1]Research School of Computer Science, Australian National University
[2]Data61, CSIRO
{alvin.kennardi, qing.wang}@anu.edu.au
gabriela.ferraro@data61.csiro.au

Abstract

One key challenge for building a semantic parser in new domains is the difficulty to annotate new datasets. In this paper, we propose a sequential transfer learning method as a domain adaptation method to tackle this issue. We show that we can obtain a model with better generalisation on a small dataset by transferring network parameters from a model trained with a bigger dataset with similar meaning representations. We evaluate our model with different datasets as well as versions of the datasets with different difficulty levels.

1 Introduction

Semantic parsing maps natural language sentences into meaning representations, for example, logical formulae, SQL queries, or executable codes. The successful implementation of the encoder-decoder architecture in the machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014) has driven researchers to apply this model into semantic parsing task (Dong and Lapata, 2016; Jia and Liang, 2016; Ling et al., 2016; Dong and Lapata, 2018). These neural semantic parsing models have achieved impressive results.

Semantic parsing datasets are usually domain and meaning representation dependent, thus making it difficult to re-use existing datasets for building general semantic parsers or semantic parsers in new domains. The process of annotating sentences with their meaning representations for modeling new domains or augmenting the existing datasets is expensive. Prior works proposed several strategies to tackle this issue, such as paraphrasing (Su and Yan, 2017), decoupling structure and lexicon (Herzig and Berant, 2018), and multi-task learning (Susanto and Lu, 2017; Herzig and Berant, 2017).

Our method aims to provide an alternative to the previous work. We perform transfer learning by training a model for one task using a dataset and fine-tuning the model using another related dataset. The idea of transfer learning is to utilize features, weights, or other knowledge acquired for one task to solve another related task. It has been extensively used for domain adaptation and building models to solve problems where only limited data is available (Pan and Yang, 2010). The fine-tuning transfer learning procedure has been successfully implemented in the encoder-decoder architecture for Neural Machine Translation Task (NMT) (Luong and Manning, 2015; Sennrich et al., 2016; Servan et al., 2016). In contrast with the multi-task learning, which jointly trains several tasks together, we perform transfer learning by training the first and second tasks in sequence.

Compared to models without transfer learning, our experiments shows that transfer learning gives a good prior for models trained with small datasets, hence improving model performance when only limited amounts of data are available.

Neural semantic parsing models are usually trained and tested using datasets in which variables are identified and anonymised before hand, thus considerably reducing the difficulty of the semantic parsing task (Finegan-Dollak et al., 2018). In this work, we use the un-anonymised versions of two semantic parsing datasets, as well as different data splits to provide extensive evaluation of our model.

To summarise, the contributions of this paper are as follows:

- Evaluation of transfer learning as domain adaptation for low-resource neural semantic parsing with different datasets and difficulty levels.

- Compilation and release of un-anonymised versions of ATIS and GeoQuery datasets for

semantic parsing in lambda calculus formulae.[1]

2 Related Work

Encoder-decoder architectures based on neural networks have been successfully applied to semantic parsing (Dong and Lapata, 2016; Jia and Liang, 2016; Ling et al., 2016; Dong and Lapata, 2018). Since then, several ideas such as including attention mechanism (Dong and Lapata, 2016), multi-task learning (Susanto and Lu, 2017; Herzig and Berant, 2017; Fan et al., 2017), data augmentation (Jia and Liang, 2016; Kočiský et al., 2016) and two-steps (coarse-to-fine) decoder (Dong and Lapata, 2018) have been applied to semantic parsing models with the aim of boosting performance.

Similar to our work, others tried to overcome the lack of annotated data by leveraging existing datasets from related domains. Previous works from Herzig and Berant (2017) and Fan et al. (2017) used a multi-task framework to jointly learn the neural semantic parsing model and encourage parameter sharing between different datasets. The model proposed by Herzig and Berant (2017) used multiple knowledge bases in different domains to enhance the model performance. On the other hand, the work from Fan et al. (2017) leveraged access to a very large labeled dataset to help a small one. However, their models are trained using proprietary datasets, which are not publicly available, thus making model comparison unfeasible. The work proposed by Damonte et al. (2019) investigates the possibility of transfer learning to tackle the issue of lacking annotated data on neural semantic parsing. They used more complex model and data sets compared to our work.

Our work focuses on training a model using a larger dataset and fine-tune using another related low-resource dataset rather than multi-task learning. We also evaluate how additional training examples impact transfer learning models.

3 Methodology

3.1 Transfer Learning as Domain Adaptation

We adapt the formal definition of transfer learning from Pan and Yang (2010) to the neural semantic parsing problem involving a question q and a meaning representation f. A domain $\mathcal{D}$ consists of input space $\mathcal{Q}$ and marginal probability $P(Q)$, where $Q = \{q_1, q_2, ..., q_n\} \subseteq \mathcal{Q}$. A domain can be denoted by $\mathcal{D} = \{\mathcal{Q}, P(Q)\}$. Given a domain $\mathcal{D} = \{\mathcal{Q}, P(Q)\}$, a task $\mathcal{T}$ consists of output space $\mathcal{F}$ and conditional probability $P(F|Q)$. A task can be denoted as $\mathcal{T} = \{\mathcal{F}, P(F|Q)\}$. In the semantic parsing problem, we want to learn conditional probability $P(F|Q)$ from the training set with training data (q_i, f_i), where $q_i \in \mathcal{Q}$ and $f_i \in \mathcal{F}$.

Suppose we have a *source domain* $\mathcal{D}_S$, with *source task* $\mathcal{T}_S$ and a *target domain* $\mathcal{D}_T$ with *target task* $\mathcal{T}_T$ where $0 < n_T << n_S$. Transfer learning uses the knowledge from D_S and T_S to improve the performance of T_T, where $D_S \neq D_T$, or $T_S \neq T_T$ (Pan and Yang, 2010).

Our transfer learning method starts by training a model in the *source domain* $\mathcal{D}_S$ to solve a *source task* T_S. Subsequently, we transfer the knowledge (i.e network parameters) to the model aimed to solve *target task* $\mathcal{T}_T$ and fine-tune the model using the *target domain* $\mathcal{D}_T$.

3.2 Model

In this work, we adopt the sequence-to-sequence with neural attention method from Dong and Lapata (2016). The model aims to map a question input $q = \langle x_1, x_2, ..., x_{|q|} \rangle$ to a meaning representation $f = \langle y_1, y_2, ..., y_{|f|} \rangle$. We want to compute the conditional probability of generating the meaning representation f given a question q as follows:

$$p(f|q) = \prod_{t=1}^{|f|} p(y_t|y_{<t}, q) \quad (1)$$

The question input q is encoded using an encoder, and then a meaning representation f is generated using an attention decoder. The encoder hidden state $\mathbf{h}_t$ and cell state $\mathbf{c}_t$ at time step t can be computed as follows:

$$\mathbf{h}_t, \mathbf{c}_t = LSTM(\mathbf{h}_{t-1}, \mathbf{c}_{t-1}, \mathrm{E}(x_t)) \quad (2)$$

where LSTM refers to a LSTM function described by Zaremba et al. (2014) and $E(.)$ is an embedding layer that returns a word vector representation of x_t. The hidden and cell state of the last encoder step are used to initialize the LSTM cell on the first decoder step, hence giving the context to the decoder. The LSTM encoder and decoder have different parameters.

The attention layers aim to include the encoder information to a meaning representation at each

[1]The code and datasets are available from https://github.com/akennardi/Semantic-Parsing

decoder step (Bahdanau et al., 2015; Luong et al., 2015). In an attention layer, we compute an attention score $s_{k,t}$ between the k-th encoder hidden state h_k and a decoder hidden state $\mathbf{h}_t$. The context vector c_t is a weighted sum of all encoder hidden vectors. We use the context vector $\mathbf{c}_t$ and the decoder hidden state $\mathbf{h}_t$, to obtain an attention hidden state vector $\mathbf{h}_t^{att}$ using equations as follows:

$$
\begin{aligned}
s_{k,t} &= \frac{\exp\{\mathbf{h}_k \cdot \mathbf{h}_t\}}{\sum_{j=1}^{|q|} \exp\{\mathbf{h}_j \cdot \mathbf{h}_t\}} \\
\mathbf{c}_t &= \sum_{k=1}^{|q|} s_{k,t}\mathbf{h}_k \\
\mathbf{h}_t^{att} &= \tanh(\mathbf{W}_1\mathbf{h}_t + \mathbf{W}_2\mathbf{c}_t)
\end{aligned}
\tag{3}
$$

The conditional probability of generating token y_t at time step t can be expressed as:

$$
p(y_t|y_{<t}, q) = (softmax(\mathbf{W}_o\mathbf{h}_t^{att}))^T\mathbf{e}(y_t) \tag{4}
$$

where $\mathbf{e}(y_t)$ is a one-hot vector with value 1 in the element of index y_t in the embedding layer and 0 otherwise.

We train our model to minimise the negative log-likelihood function over questions and formulae in the training set $\mathcal{T}$. The optimisation problem can be written as follows:

$$
\text{minimise} \quad -\sum_{(q,f)\in\mathcal{T}} \log(p(f|q)) \tag{5}
$$

Given a question q, we used the model to generate the most probable sequence $\tilde{f}$ as follows:

$$
\tilde{f} = arg\,max_{f'}\, p(f'|q) \tag{6}
$$

The model performs a greedy search to generate one token at a time to construct a sequence in lambda calculus.

4 Experiments

4.1 Datasets

For evaluation we used two semantic parsing datasets, namely ATIS and GeoQuery. The meaning representation of the datasets is lambda calculus. There are two types of dataset splits: question-split and query-split. In question-split, the training and test examples are divided based on the questions (Finegan-Dollak et al., 2018), thus based on the input sequence. Meanwhile, in query-split, the training and test examples are divided according to the similarity of their meaning representations (Finegan-Dollak et al., 2018), thus based on the output sequences. Therefore, the query-split is more appropriate to evaluate the model's capability of composing output sequences, in this case, lambda calculus expressions.

ATIS
Question : cheapest fare from ci0 to ci1 Formula : (min $0 (exists $1 (and (from $1 ci0) (to $1 ci1) (= (fare $1) $0))))
ATIS Un-anonymised
Question : cheapest fare from Indianapolis to Seattle Formula : (min $0 (exists $1 (and (from $1 indianapolis) (to $1 seattle) (= (fare $1) $0))))
GeoQuery
Question : what is the capital of s0 Formula : (capital:c s0)
GeoQuery Un-anonymised
Question : what is the capital of Georgia Formula : (capital: georgia)

Table 1: Example of natural language questions and their meaning representation in lambda calculus.

Data Set	Train	Dev.	Test
ATIS	4,434	491	448
ATIS un-anonymised	4,029	504	504
GeoQuery	600	0	280
GeoQuery un-anonymised	583	15	279
GeoQuery un-anonymised + query-split	543	148	186

Table 2: Number of training (Train), development (Dev.), and testing (Test) instances for each dataset.

The ATIS dataset (Price, 1990; Dahl et al., 1994; Zettlemoyer and Collins, 2007) consists of queries from a flight booking system. We obtained the un-anonymised version of ATIS by preprocessing the non-SQL ATIS dataset (Finegan-Dollak et al., 2018). Question variables in this dataset are not anonymised, but the formulae have variable identifiers. We removed the variable identifiers in logical formulae. The ATIS dataset split is question-split.

The GeoQuery dataset (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005) consists of queries about US geographical information. We annotated the un-anonymised version of GeoQuery based on non-SQL GeoQuery dataset (Finegan-Dollak et al., 2018), which has different meaning representations. We compared the question with the anonymised version and annotated lambda calculus formulae on the non-SQL GeoQuery dataset. We ran a script to put the variable back into the questions-formulae pairs, and split them into training, development and test sets based

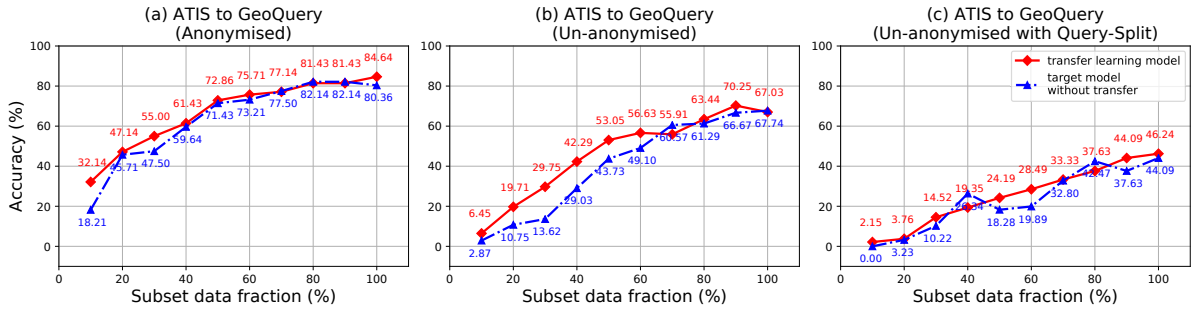

Figure 1: Learning curves from different transfer learning setups.

Source Domain	Target Domain
ATIS	GeoQuery
ATIS un-anonymised	GeoQuery un-anonymised
ATIS un-anonymised	GeoQuery un-anonymised with query-split

Table 3: Transfer learning experiments with ATIS and GeoQuery datasets.

on Finegan-Dollak et al. (2018). We also divided the GeoQuery un-anoymised dataset using query-split as proposed by Finegan-Dollak et al. (2018). Table 2 shows the details of each dataset.

4.2 Setup

We considered ATIS as a *Source Domain* dataset and GeoQuery as a *Target Domain* dataset. We believe that ATIS training samples are less similar, since it could only achieve a good model performance using more training samples. Thus it is more beneficial to use ATIS as *Source Domain*. We evenly divided the GeoQuery into 10 subsets of {10%, 20%,...,100%} fraction of the training set. With this setup, we simulate the situation where we have limited data in the target domain. This setup also allowed us to evaluate the effectiveness of transfer learning with sufficient training data. Details about the experiments setups are depicted in Table 3.

We set the model hyper-parameters following Dong and Lapata (2016) for GeoQuery. We optimised the objective function in Equation 5 using RMSProp algorithm (Tieleman and Hinton, 2012) with a decay rate of 0.95. The batch size was 20. We randomly initialised parameter from the uniform distribution $\mathcal{U}(-0.08, 008)$. The hidden unit size was 150, and the dropout rate was 0.5. We used 15 epoch to obtain a model from ATIS. We increased the number of epochs after transferring all network parameters to 150 and 180 for anonymised and un-anonymised GeoQuery, respectively. Source and target models were trained with their own vocabularies to handle differences of vocabularies between two datasets. The evaluation metric was accuracy. We evaluated each model with inference described in Equation 6 on the full GeoQuery test set for every bucket. We reported exact match accuracy computed using equation as follows:

$$\text{Accuracy} = \frac{\text{\# of correct formulae}}{\text{\# test examples in the test set}} \quad (7)$$

4.3 Evaluation on Transfer Learning

We compared our transfer learning framework with the original target model (i.e. without transfer learning) in three different setups described in Section 4.2. Figure 1 shows the learning curves of those setups. The results from small GeoQuery subsets confirmed our hypothesis that the source model gives a stronger prior to the target model. The model obtained from transfer learning has 13.93%, 3.58%, and 2.15% accuracy improvement on the 10% fraction of GeoQuery, GeoQuery Un-anonymised, and GeoQuery Un-anonymised with Query-Split datasets respectively. Figure 1(a) and (b) clearly shows how the transfer learning improves the performance of the target models trained with small subsets. In Figure 1(c), the performance of the model with transfer learning are comparable to the original target model. However, the performance of original target model drops with additional training examples from 40% to 50% subset. On the other hand, the model with transfer learning does not have a sudden drop. A possible explanation to this result may be due to the difficulty of the original target model to learn from difficult training samples. The learning curves of the transfer learning models show smoother changes with additional training data as

No.	Question	Transfer Learning	Original Target Model
1	river in s0	(lambda $0 e (and (river:t $0) (loc:t $0 s0)))	(lambda $0 e (and (river:t $0) (loc:t $0 s0)) (size:i $0))
2	what is the capital of the smallest state	(capital:c (argmin $1 (state:t $1) (size:i $1)))	(capital:c (argmax $1 (state:t $1) (size:i $1)))
3	how many rivers does colorado have	(count $0 (and (river $0) (loc $0 colorado)))	(count $0 (and (state $0) (loc $0 usa)))
4	how large is texas	(size texas)	(argmax $0 (river $0) (density $0))
5	how many states does missouri border	(count $0 (and (state $0) (next_to $0 missouri)))	(count $0 (and (state $0) (next_to $0 delaware)))
6	how many states does the missouri river run through	(count $0 (and (state $0) (loc $0 missouri)))	(lambda $0 e (and (state $0) (loc $0 missouri)))

Table 4: Examples of Meaning Representations generated by the model trained with transfer learning and original target model using 10% fraction of various GeoQuery datasets.

compared to the original target model, indicating better model generalisation when the training data is small. With bigger subset (i.e 70% and more), the results from transfer learning models are comparable to the original models, indicating that the out-of-domain data does not impair the model performance. We show that our transfer learning method helps the target model to have a better performance when the training data is very small.

4.4 Error Analysis on Transfer Learning

We also looked into samples generated from the transfer learning models and original target models. Table 4 presents six samples from three different setups described in Table 3 with the target model trained with 10% subset of training examples. The first two samples are obtained from the models trained with GeoQuery. In the first example, the model trained with transfer learning can identify correct meaning representation, while the original target model generates wrong meaning representation due to the generation of extra tokens. The second example shows the model trained with transfer learning correctly identified the token "smallest" to generate "argmin" instead of "argmax".

The third and fourth samples show examples of meaning representations generated by the model trained with un-anonymised GeoQuery. In the third examples, the model with transfer learning correctly identified the entity "river". On the other hand, the model without transfer learning generates "state", which is more common in the training set. On the fourth example, the original target model generates an irrelevant meaning representation.

The last two samples are obtained from models trained with un-anonymised GeoQuery with query-split. The fifth example shows how the original target generates a wrong entity name "delaware" instead of "missouri". Similarly, the sixth example shows original target model produces a token "lambda" instead of "count". This error may be due to the fact that the original target model tends to generate the token they are familiar with in the training set. Examples described above shows how the model trained with transfer learning has a better ability to generate tokens that are different with training examples, thus improve the performance of the model.

5 Conclusion and Future Work

We proposed a transfer learning method by training a model using a larger dataset and fine-tuning with another related low-resource dataset. With this method, we can use a bigger dataset with a similar composition to improve the performance of a model trained with a smaller dataset.

For future work, it would be interesting to combine transfer learning and data selection methods so that the source model is trained only with the most similar instances in respect with the target domain. Another direction would be to explore transfer learning on a more complex model such as sequence-to-tree, which has a better performance than sequence-to-sequence models when trained with large datasets.

Acknowledgement

We would like to thank Xiang Li for his insight throughout the project. We would also like to thank the three anonymous reviewers for their valuable comments and insights. This work is a part of Individual Computing Project Course at the Australian National University taken by the first author with the same title.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the atis task: The atis-3 corpus. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Marco Damonte, Rahul Goel, and Tagyoung Chung. 2019. Practical semantic parsing for spoken language understanding. *CoRR*, abs/1903.04521.

Li Dong and Mirella Lapata. 2016. Language to Logical Form with Neural Attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2016, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2018. Coarse-to-Fine Decoding for Neural Semantic Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2018, pages 731–742, Melbourne, Australia. Association for Computational Linguistics.

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 2017, pages 48–56, Vancouver, Canada. Association for Computational Linguistics.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

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 (Volume 2: Short Papers)*, ACL 2017, pages 623–628, Vancouver, Canada. Association for Computational Linguistics.

Jonathan Herzig and Jonathan Berant. 2018. Decoupling structure and lexicon for zero-shot semantic parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1619–1629, Brussels, Belgium. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2016. Data Recombination for Neural Semantic Parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2016, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2013, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Tomáš Kočiský, Gábor Melis, Edward Grefenstette, Chris Dyer, Wang Ling, Phil Blunsom, and Karl Moritz Hermann. 2016. Semantic Parsing with Semi-Supervised Sequential Autoencoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2016, pages 1078–1087, Austin, TX, USA. Association for Computational Linguistics.

Wang Ling, Phil Blunsom, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Fumin Wang, and Andrew Senior. 2016. Latent Predictor Networks for Code Generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2016, pages 599–609, Berlin, Germany. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *International Workshop on Spoken Language Translation*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *Trans. on Knowledge and Data Eng.*, 22(10):1345–1359.

P. J. Price. 1990. Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Christophe Servan, Josep Maria Crego, and Jean Senellart. 2016. Domain specialization: a post-training domain adaptation for neural machine translation. *ArXiv*, abs/1612.06141.

Yu Su and Xifeng Yan. 2017. Cross-domain semantic parsing via paraphrasing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1235–1246, Copenhagen, Denmark. Association for Computational Linguistics.

Raymond Hendy Susanto and Wei Lu. 2017. Neural Architectures for Multilingual Semantic Parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL 2017, pages 38–44, Vancouver, Canada. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

T. Tieleman and G. Hinton. 2012. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *CoRR*, abs/1409.2329.

John M. Zelle and Raymond J. Mooney. 1996. Learning to Parse Database Queries Using Inductive Logic Programming. In *Proceedings of the 13th National Conference on Artificial Intelligence*, volume 2, pages 1050–1055, Portland, Oregon, USA. AAAI Press / The MIT Press.

Luke Zettlemoyer and Michael Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 678–687, Prague, Czech Republic. Association for Computational Linguistics.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, pages 658–666, Arlington, Virginia, United States. AUAI Press.