A Transformer-based Semantic Parser for NLPCC-2019 Shared Task 2*

Donglai Ge[1], Junhui Li[1], and Muhua Zhu[2]

[1] School of Computer Science and Technology, Soochow University, Suzhou, China
20175227014@stu.suda.edu.cn, lijunhui@suda.edu.cn
[2] Alibaba Group, Hangzhou, China
muhua.zmh@alibaba-inc.com

Abstract. Sequence-to-Sequence (seq2seq) approaches formalize semantic parsing as a translation task from a source sentence to its corresponding logical form. However, in the absence of large-scale annotated dataset, even the state-of-the-art seq2seq model, i.e., the *Transformer* may suffer from the data sparsity issue. In order to address this issue, this paper explores three techniques which are widely used in neural machine translation to better adapt seq2seq models for semantic parsing. First, we use byte pair encoding (BPE) to segment words into subwords to transfer rare words into frequent subwords. Second, we share word vocabulary on both the source and the target sides. Finally, we define heuristic rules to generate synthetic instances to increase the coverage of training dataset. Experimental results on the NLPCC 2019 shared task 2 show that our approach achieves state-of-the-art performance and gets the first place in the task from the current rankings.

Keywords: Semantic parsing · Sequence-to-sequence · Synthetic instances

1 Introduction

The task of semantic parsing, which aims to map natural language utterances into their corresponding meaning representations, has received a significant amount of attention with various approaches over the past few years. The languages of meaning representation mainly fall into two categories: logic based formalisms and graph-based formalisms. The former includes *first order logic*, *lambda calculus*, and *lambda dependency based compositional semantics*, while the latter includes *abstract meaning representation* and *universal conceptual cognitive annotation.* Traditional approaches are mostly based on the principle of compositional semantics, which compose the semantics of utterances from lexical semantics by using a set of predefined grammars. The widely used grammars include SCFG [16, 11], CCG [10, 4], DCS [12, 3], etc. One of the main shortcomings

* Supported by the National Natural Science Foundation of China (Grant No., 61876120)

of grammar-based approaches is that they rely on high-quality lexicons, handcrafted features, and manually-built grammars. In recent years, one promising direction in semantic parsing is to represent the semantics of texts as graphs. This way, semantic parsing can be formalized as a process of graph generation. In this direction, Ge et al [6] propose to obtain semantic graphs through transformation from syntactic trees. Reddy et al [13] use Freebase-based semantic graph representation and convert sentences into semantic graphs by using CCGs or dependency trees. Bast et al [2] identify the structure of a semantic query through three predefined patterns. Yih et al [18] generate semantic graphs using a staged heuristic search algorithm. All these approaches are based on manually-designed and heuristic generation process, which may suffer from syntactic parsing errors and structure mismatching, especially the case for complex sentences.

An alternative to the aforementioned approaches to semantic graph generation is to utilize the sequence-to-sequence (seq2seq) framework, which has been adopted for a variety of natural language processing tasks [7, 1], semantic parsing included [17, 5]. Now the task at hand translates to building seq2seq models in order to map word sequences into corresponding sequences that represent semantic graphs. To train such models, it is important to have enough training data of high quality. Generally, the performance of seq2seq models is highly dependent on the quality and quantity of available training data. However, most of the datasets for semantic parsing are curated by human, which is labor intensive and time consuming. Consequently, annotated corpora are generally limited in size and training of seq2seq models tends to suffer from the scarcity of annotated training data.

The NLPCC-2019 shared task 2 is a competition for open domain semantic parsing, which is defined to predict the meaning representation in lambda-calculus for an input question on the base of a given knowledge graph. Each question in the shared task data is annotated with entities, the question type, and the corresponding logical form. The dataset is called Multi-perspective Semantic ParSing (MSParS) and includes more than 80,000 human-generated instances. In MSParS, there is a total of 9 question types, including single-relation, multi-hop, multi-constraint, multi-choice, aggregation, comparison, yes/no, superlative, and multi-turn. Table 1 presents an illustrating example of a question accompanied by its logical form, entities, and question type: the first row is the question that we need to parse, the second row presents the logical form of the question, the third row shows the entities and their positions in the logical form, and the last row gives the question type. Participating systems are evaluated on the prediction of logical forms given input questions

In the competition of shared task 2, we build our semantic parsing system on the base of the Transformer, a state-of-the-art seq2seq model that is originally proposed for neural machine translation and syntactic parsing [15]. Furthermore, to enhance the performance of our system, we apply the following three techniques: using byte pair encoding (BPE) to segment words into subwords, sharing word vocabulary on both the source and target sides, and enlarging training data by automatically generating synthetic training instances. In the NLPCC 2019

Table 1. An example from the dataset for open domain semantic parsing.

<question id=1>	who is film art directors of “ i see you ” from avatar
<logical form id=1>	(lambda ?x (mso:film.film.art_director “_i_see_you_” _from_avatar ?x))
<parameters id=1>	“_i_see_you_”_from_avatar (entity) [6,12]
<question type id=1>	single-relation

Shared Task 2, our system win the first place among all the participating systems and the proposed techniques achieve remarkable improvements over the Transformer baseline.

2 Semantic Parsing as Neural Seq2Seq Learning

In this section, we describe in detail our approach to semantic parsing. The model is built on *Transformer*, a state-of-the-art seq2seq model that is originally proposed for neural machine translation and syntactic parsing [15].

2.1 Preparing Data

Each instance in the MSParS dataset is a tuple of four elements, including a question, its logical form, parameters, and question type. In this paper we only use questions and their corresponding logical forms to train our parsing model but ignore parameters and question type because they are not evaluated. Questions are fed into the encoder as source sequences and their corresponding logical forms are viewed as target sequences.

Note that in the logical form, an entity is presented as a string which consists of multiple words concatenated by ‘_’. In pre-processing, we split an entity string into its corresponding multiple words and symbols of ‘_’. For example, the logical form in Table 1 is processed as:
(lambda ?x (mso:film.film.art_director “ _ i _ see _ you _” _ from _ avatar ?x))

In post-processing, we resume entity strings by simply replacing ‘ _ ’ with ‘_’ in output sequences.

We have also tried to split the strings of entity types into multiple pieces. For example, *mso:film.film.art_director* in Table 1 is split into *mso : film . film . art_ director*. However, our preliminary experiments showed that it slightly hurts the performance.

2.2 Sequence-to-Sequence Modeling

As mentioned, we use Transformer seq2seq model for semantic parsing. The encoder in Transformer consists of a stack of multiple identical layers, each of which has two sub-layers, one for multi-head self-attention mechanism, and the other is a position-wise fully connected feed-forward network. The decoder is also composed of a stack of multiple identical layers. Each layer in the decoder consists of the same sub-layers as in the encoder layers as well as an additional sub-layer that performs multi-head attention to the output of the encoder stack. Experiments on the tasks of machine translation and syntactic parsing show that Transformer outperforms RNN-based seq2seq models [1]. Fig. 1 shows the structure of Transformer seq2seq model.

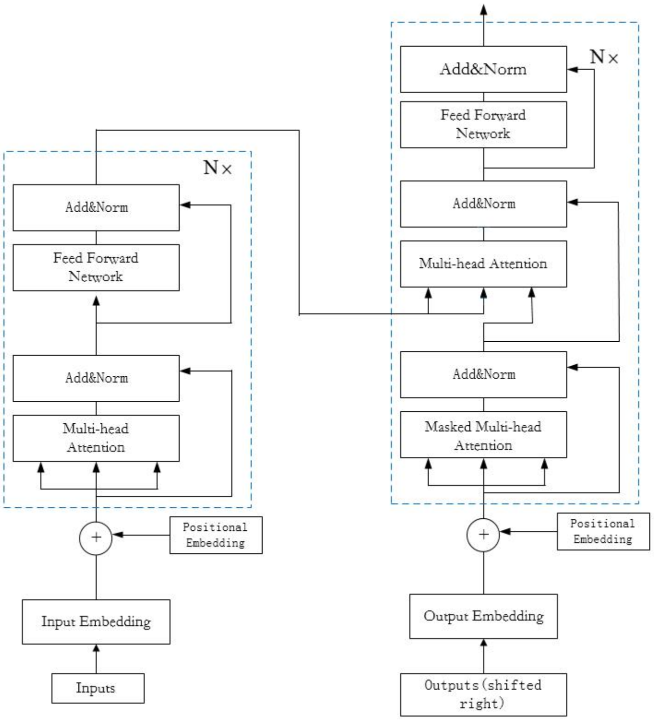


Fig. 1. Transformer seq2seq model.

The self-attention in Transformer uses Scaled Dot-Product Attention which operates on an input sequence, $x = (x_1, \cdots, x_n)$ of n elements where $x_i \in R^{d_x}$ and computes a new sequence $z = (z_1, \cdots, z_n)$ with the same length:

$$z = Attention\,(x) \tag{1}$$

where $z \in R^{n \times d_z}$. Each output element z_i is calculated as a weighted sum of a linearly transformed input elements:

$$z_i = \sum_{j=1}^{n} \alpha_{ij} \left(x_j W^V \right) \tag{2}$$

where $W^V \in R^{d_x \times d_z}$ is a parameter matrix, and

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})} \tag{3}$$

$$e_{ij} = \frac{\left(x_i W^Q\right)\left(x_j W^K\right)^T}{\sqrt{d_z}} \tag{4}$$

where the weight vector $\alpha_i = (\alpha_{i1}, \cdots, \alpha_{in})$ over input vectors is obtained by self-attention model, which captures the correspondences between element x_i and others, and e_{ij} is an alignment model which scores how well the input elements x_i and x_j match. Here $W^Q, W^K \in R^{d_x \times d_z}$ are parameter matrices.

2.3 Generation of Synthetic Training Instances

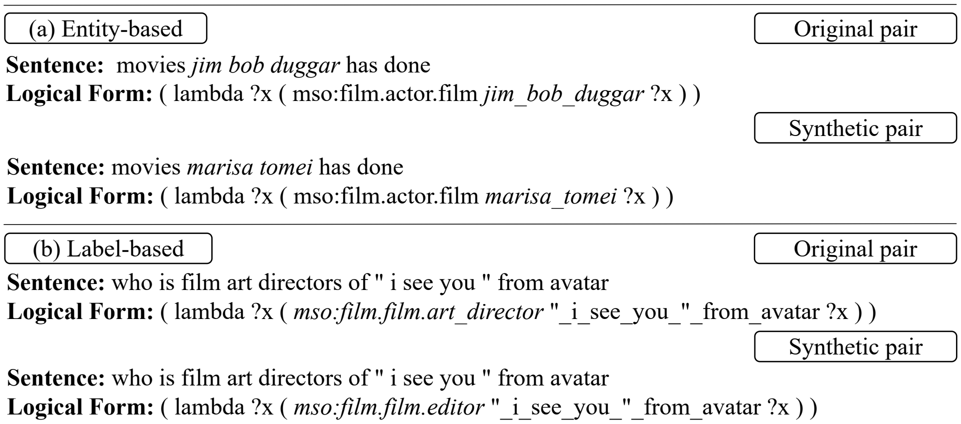
Supervised machine learning algorithms tend to suffer from data imbalance problems. In the dataset of MSParS, we find entity types contain quite skewed distributions. For example, the entity type $mso : film.actor.film$ contains the most entity instances 1832 while the entity type mso : *baseball.batting_statistics.slugging_pct* only has 1 entity instances. Seq2seq models trained on such a kind of dataset may be overwhelmed by training instances of big entity types while parameters for small entity types are not well learned. The resulting models are apt to achieve relatively poor performance on test set due to limited generalization ability.

To attack the data imbalance problem, we generate synthetic training instances from the following two perspectives:

- Entity-based: Given a sentence and its logical form from the original training set, we choose one entity in the sentence and replace it with another random entity which has the same entity type. Figure2(a) shows examples of an original pair and its synthetic pair.
- Label-based: Given a sentence and its logic from the original training set, we choose one entity who has multiple entity types and replace its entity type with another valid type. As shown in Figure2(b), since the entity of *"_i_see_you_"_from_avatar* has multiple entity types, we randomly select another entity type but *film.film.art_director*, and generate a synthetic pair.

3 Experimentation

In this section, we first introduce the dataset we used. Then we describe the settings of our model for the experiments. After that, we present a comparative study on our system and other participating systems.



(a) Entity-based

Original pair

Sentence: movies *jim bob duggar* has done
Logical Form: (lambda ?x (mso:film.actor.film *jim_bob_duggar* ?x))

Synthetic pair

Sentence: movies *marisa tomei* has done
Logical Form: (lambda ?x (mso:film.actor.film *marisa_tomei* ?x))

(b) Label-based

Original pair

Sentence: who is film art directors of " i see you " from avatar
Logical Form: (lambda ?x (*mso:film.film.art_director* "_i_see_you_"_from_avatar ?x))

Synthetic pair

Sentence: who is film art directors of " i see you " from avatar
Logical Form: (lambda ?x (*mso:film.film.editor* "_i_see_you_"_from_avatar ?x))

Fig. 2. Examples of automatically generated synthetic instances

3.1 Experimental Settings

Dataset We take our evaluation on the Multi-perspective Semantic ParSing (or MSParS) released by NLPCC-2019 Shared Task 2. The dataset includes more than 80,000 human-generated questions, where each question is annotated with entities, a question type, and corresponding logical form. The organizers split MSParS into a training set, a development set, and a test set. Both the training and development sets are provided to participating teams, while the test set is not. The training set has 63,826 instances and development set has 9,000 instances. Note that the organizers divide the test set to select a hard subset according to certain criteria, so each team has two final results: full set score and hard subset score.

Evaluation The evaluation uses accuracy (ACC), i.e. the percentage of predicted logical forms which exactly match the golden ones.

Settings We use openNMT [9] as the implementation of the Transformer seq2seq model. In the parameter setting, we set the number of layers in both the encoder and decoder to 6. For optimization we use Adam [8] with $\beta 1 = 0.1$. The number of heads is set to 8. In addition, we set the hidden size to 512 and the batch token-size to 8192. In all experiments, we train the models for 250K steps on a single K40 GPU and save the models at every 5K steps. To overcome the data sparsity issue, in all experiments we follow Ge et al. [5] and share vocabulary for the input and the output. To address the translation of rare words, we segment words into word pieces by byte pair encoding (BPE [14]) with 8K operations. We average the last 20 models' parameters to improve the performance .

Table 2. Ablation results of our baseline system on the development set.

Model	ACC
Baseline	85.93
-BPE	54.90
-Sharing Vocab.	84.00
-Both	52.47

3.2 Experimental Results

We first show the performance of our baseline system. As mentioned earlier, BPE and sharing vocabulary are two techniques we applied to relieving data sparsity. Table 2 presents the results of the ablation test on the development set by either removing BPE, or vocabulary sharing, or both of them from the baseline system. From the results we can see that BPE and vocabulary sharing are critical to building our baseline system (an improvement from 52.47 to 85.93 in accuracy), revealing that they are two effective ways to address the issue of data sparsity for semantic parsing.

Generation of synthetic training instances substantially increases the number of instances in our training set. As shown in Table 3, both the two methods of generating synthetic training instances roughly double the number of training instances and achieve similar improvements over the model trained on the original training set (e.g., 0.85 and 1.01), suggesting that our two methods are effective in increasing the coverage of training instances. However, there exists overlap in coverage of the two methods. In the presence of one method, the other method achieves limited or no improvement.

We also compare our final system with systems from other participants in Table 4. From the results we can see that our final system achieves the highest performance, especially on the hard subset. This illustrates the feasibility and effectiveness of our seq2seq-based semantic parsing.

Table 3. ACC (%) of our semantic parsing models on the development set.

Training set	# Instances	ACC
Original	63,826	85.93
+Entity-based	137,198	86.78
+Label-based	140,485	86.94
+Both (our final model)	213,857	86.96

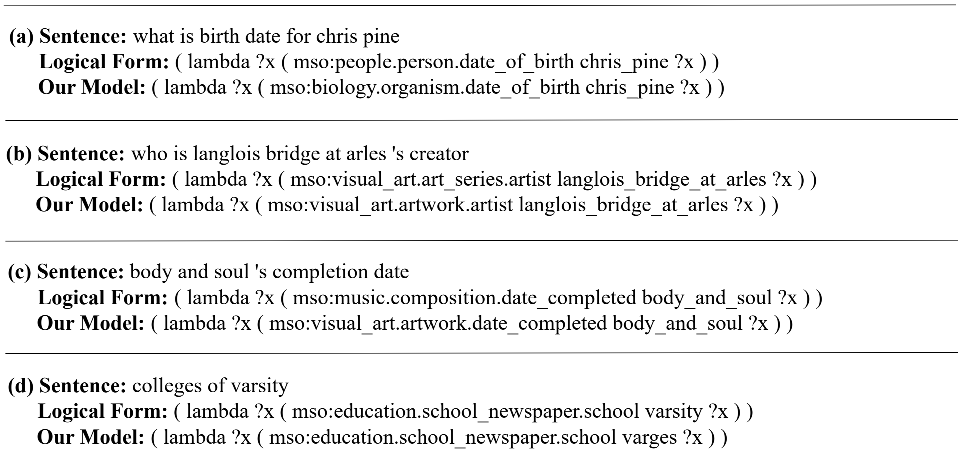
3.3 Error Analysis

To find the reasons for improper parsing, we analyze 50 bad cases selected randomly from the development set. The mistakes mainly fall into four categories. First, entity type is incorrectly predicted when the entity has multiple types. As

Table 4. Comparison of our final parser with other parsers

Model	ACC on full set	ACC on hard subset
Soochow_SP (this paper)	85.68	57.43
NP-Parser	83.73	51.93
WLIS	82.53	47.83
Binbin Deng	68.82	35.41
kg_nlpca_ai_lr	30.79	14.89
TriJ	26.77	14.49

shown in Figure 3 (a), entity *chris pine* has 5 types in the training set and our model incorrectly predict its type as *biology.organism* in this example. 16 wrong cases out of the 50 ones are caused by this error category. Second, it is hard to correctly predict entity type if an entity occurs only once in the training set. As shown in Figure 3 (b), *langlois bridge at arles* occurs once in the training set and our model incorrectly predicts its type as *visual_art.artwork*. 7 wrong cases belong to this error category. Third, for those entities that even do not appear in the training set, our model tends to make incorrect prediction. As shown in Figure 3, though our model successfully recognizes *body and soul* as an entity, it fails to identify its entity type. 18 wrong cases are in this error category. Finally, in few cases our model sometimes fails to recognize entities. As shown in Figure3 (d), our model fails to recognize entity *varsity*.



(a) Sentence: what is birth date for chris pine
Logical Form: (lambda ?x (mso:people.person.date_of_birth chris_pine ?x))
Our Model: (lambda ?x (mso:biology.organism.date_of_birth chris_pine ?x))

(b) Sentence: who is langlois bridge at arles 's creator
Logical Form: (lambda ?x (mso:visual_art.art_series.artist langlois_bridge_at_arles ?x))
Our Model: (lambda ?x (mso:visual_art.artwork.artist langlois_bridge_at_arles ?x))

(c) Sentence: body and soul 's completion date
Logical Form: (lambda ?x (mso:music.composition.date_completed body_and_soul ?x))
Our Model: (lambda ?x (mso:visual_art.artwork.date_completed body_and_soul ?x))

(d) Sentence: colleges of varsity
Logical Form: (lambda ?x (mso:education.school_newspaper.school varsity ?x))
Our Model: (lambda ?x (mso:education.school_newspaper.school varges ?x))

Fig. 3. Examples of error types

4 Conclusion

In this paper, we present our seq2seq model that parses natural language utterances to logical forms. To overcome the data sparsity issue, we use BPE to segment rare words into frequent subwords, and we share vocabulary on the source and the target side, considering the fact that many words are common on both sides. Finally, to increase the coverage of training instances, we use heuristic rules to generate synthetic instances from the original ones. Experiments on the NLPCC-2019 shared task 2 show that our approach achieves state-of-the-art performance and ranks the first among the participating systems.

Detailed analysis shows that misjudgement of entity types is one of the major error sources. In future work, we will focus on joint learning of entity recognition and semantic parsing.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of ICLR (2015)
2. Bast, H., Haussmann, E.: More accurate question answering on freebase. In: Proceedings of CIKM. pp. 1431–1440 (2015)
3. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: Proceedings of EMNLP. pp. 1533–1544 (2013)
4. Cai, Q., Yates, A.: Large-scale semantic parsing via schema matching and lexicon extension. In: Proceedings of ACL. pp. 423–433 (2013)
5. Ge, D., Li, J., Zhu, M., Li, S.: Modeling source syntax and semantics for neural amr parsing. In: Proceedings of IJCAI. pp. 4975–4981 (2019)
6. Ge, R., Mooney, R.J.: Learning a compositional semantic parser using an existing syntactic parser. In: Proceedings of ACL. pp. 611–619 (2009)
7. Ilya, S., Oriol, V., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of NIPS. pp. 3104–3112 (2014)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of ICLR (2015)
9. Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.M.: Opennmt: Open-source toolkit for neural machine translation. In: Proceedings of ACL, System Demonstrations. pp. 67–72 (2017)
10. Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., Steedman, M.: Lexical generalization in ccg grammar induction for semantic parsing. In: Proceedings of EMNLP. pp. 1512–1523 (2011)
11. Li, J., Zhu, M., Lu, W., Zhou, G.: Improving semantic parsing with enriched synchronous context-free grammar. In: Proceedings of EMNLP. pp. 1455–1465 (2015)
12. Liang, P., Jordan, M.I., Klein, D.: Learning dependency-based compositional semantics. In: Proceedings of ACL. pp. 590–599 (2011)
13. Reddy, S., Lapata, M., Steedman, M.: Large-scale semantic parsing without question-answer pairs. Transactions of the Association for Computational Linguistics **2**, 377–392 (2014)
14. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of ACL. pp. 1715–1725 (2016)

15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., N.Gomez, A., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of NIPS. pp. 5998–6008 (2017)
16. Wong, Y.W., Mooney, R.J.: Learning synchronous grammars for semantic parsing with lambda calculus. In: Proceedings of ACL. pp. 960–967 (2007)
17. Xiao, C., Dymetman, M., Gardent, C.: Sequence-based structured prediction for semantic parsing. In: Proceedings of ACL. pp. 1341–1350 (2016)
18. Yih, W.T., Chang, M.W., He, X., Gao, J.: Semantic parsing via staged query graph generation: Question answering with knowledge base. In: Proceedings of ACL. pp. 1321–1331 (2015)