Transformer Semantic Parsing

Gabriela Ferraro[1,2] and **Hanna Suominen**[2,1,3]

[1]Commonwealth Scientific and Industrial Research Organisation, Australia
[2]Research School of Computer Science, The Australian National University, Australia
[3]Department of Future Technologies, University of Turku, Finland
gabriela.ferraro@data61.csiro.au
hanna.suominen@anu.au

Abstract

In neural semantic parsing, sentences are mapped to meaning representations using encoder–decoder frameworks. In this paper, we propose to apply the Transformer architecture, instead of recurrent neural networks, to this task. Experiments in two data sets from different domains and with different levels of difficulty show that our model achieved better results than strong baselines in certain settings and competitive results across all our experiments.

1 Introduction

Semantic parsing maps natural language sentences to meaning representations including, but not limited to, logical formulas, Structured Query Language (SQL) queries, or executable codes. In recent years, end-to-end neural semantic parsing has achieved good results (Dong and Lapata, 2016; Jia and Liang, 2016a; Ling et al., 2016; Dong and Lapata, 2018; Finegan-Dollak et al., 2018). The main advantage of these models is that they do not require intermediate representations, lexicons, manually designed templates, or handcrafted features.

Current neural semantic parsing models use encoder–decoder architectures with Recurrent Neural Networks (RNNs). One drawback of RNNs is their inability to capture long distance relationships between input tokens or between input and output tokens (Yu et al., 2019). Hence, to model dependencies disregarding their distance, the best performing models also include some kind of an attention mechanism (Vaswani et al., 2017), which allows to model dependencies more accurately; specifically, the Transformer architecture introduced in this last reference has become the new state-of-the-art for sequence-to-sequence problems.

The Transformer architecture consists of a self-attention mechanism and does not include recurrent layers. Unlike RNNs, in which sequences are processed sequentially — word by word — Transformer models process the entire sentence as a whole. This characteristic is particularly beneficial in capturing long distance dependencies as the self-attention mechanism sees all the words at once. Despite the success of Transformer, to the best of our knowledge, prior to Li (2019) that founded our paper, this framework has never been applied to semantic parsing before. Thus, in this paper, we propose the Transformer architecture for semantic parsing.

A well-known limitation of sequence-to-sequence models is their inability to learn competitive parameter values for words that are rare in a given data set. To alleviate this problem, a common method is to anonymise entities with their respective types. For example, city names such as *Denver* are anonymised as *ci0* and later, as part of post-processing, put back in the output utterance. Neural semantic parsing models are usually trained and tested using data sets where variables are identified and anonymised beforehand — as in the example above — which considerably reduces the difficulty of the semantic parsing task (Finegan-Dollak et al., 2018). As a result, many input sentences of the test set are seen prematurely while training.

Consequently, we have a twofold approach to evaluate our model more extensively in this work and demonstrate its contributions to semantic parsing: First, we use non-anonymised versions of two data sets for semantic parsing from different domains, as well as different data splits. Second, we test the ability of our model to compose new output meaning representations. These experiments give evidence of the Transformer model outperforming strong baselines in certain settings. Its results are competitive on other settings across the data sets.

The rest of the short paper is organised as follows: Section 2 includes the related work in neu-

ral semantic parsing. Section 3 describes our model architecture, data sets, and experimental setup. Section 4 introduces our experimental results. Section 5 concludes our study.

2 Related Work

Encoder–decoder architectures based on neural networks have been applied in the past five years (i.e., up to, and including, 2019) to semantic parsing, but they have typically learnt from sentences paired with meaning representations without using explicit syntactic knowledge. Dong and Lapata (2016) have proposed two models with an attention mechanism as follows: the first model generates sequences and the second one generates trees as logical formulae that are hierarchically structured. Both models include an attention mechanism over a RNN that has the ability of focusing on a subset of input tokens or features. They also provide soft alignments between the input sentences and the logical formulae. Later, Dong and Lapata (2018) have proposed to use a two-step (coarse-to-fine) decoder to better model the compositionality of the logical formulae. Finally, to overcome the limitations of semantic parsing data sets being small and domain-dependent, several methods, such as multi-task learning (Susanto and Lu, 2017; Herzig and Berant, 2017; Fan et al., 2017), transfer learning (Kennardi et al., 2019; Damonte et al., 2019) , and data augmentation (Jia and Liang, 2016a; Kočiský et al., 2016), have been applied.

However, the aforementioned models cannot address the fact that the distance between tokens is not always an indication of a weak relationship. This problem becomes significantly worse for long sentences paired with long logical formulae. Thus, we propose to use a Transformer-based model in which token relations are not affected by the (long) distance.

As mentioned, data sets for semantic parsing are small and neural models do not tend to be good at learning the appropriate parameters for the long tail of rare words. To mitigate this problem, a common method is to apply variable or entity anonymisation as a pre-processing step. Later, the entities are put in the output sequence in a post-processing step. Another strategy is to use Pointer networks (Vinyals et al., 2015) where input tokens are copied to the output sequence at each decoding step. Moreover, attention-based copying (Jia and Liang, 2016b) refers to a mechanism in which the decoder can either choose to copy over a word to the output sequence or to pick from a softmax over the entire vocabulary. In our study, we use two data sets from different domains with and without variable anonymisation, and different splits to reflect diffing complexity levels of the semantic parsing task.

3 Methods

As our model architecture, we have implemented a self-attention neural semantic parsing model with the mechanism of Transformer (see Vaswani et al. (2017, Figure 1) for further information). As in other state-of-the-art sequence-to-sequence architectures, Transformer is essentially an encoder–decoder structure with blocks for encoding and decoding. Similar to other neural generation models with attention, the output of the last layer of the encoder is used as part of the input of each layer of the decoder. The most significant difference between Transformer and other sequence-to-sequence models is that Transformer uses neither *Convolutional Neural Networks* (CNNs) nor RNNs. Instead, it uses self-attention, which reduces path lengths within the network, thus minimising the loss of information due to computations.

Identical encoder blocks consist of multi-head self-attention and fully-connected feed-forward layers. After each layer there is a residual module, followed by a normalisation step. This produces a 512-dimensional output.

Each of identical decoder blocks has a multi-head self-attention layer, fully-connected feed-forward layer, and multi-head attention layer over the output of the encoder stack. Again, a residual module and normalisation step are applied to each output layer. Multi-head self-attention layers in encoding and decoding are similar, except in the latter one adds a mask operation between scaling and the softmax activation function. The reason for adding this look-ahead mask is to avoid that at the timestamp t, the tokens after t are used for predicting token at t .

In most Natural Language Processing tasks, the model should capture the order and position information from the sequential inputs. This is one of the advantages of RNNs and CNNs. Thus, our model includes a position embedding in the source and target inputs as in Vaswani et al.

Table 1: Example sentences and their corresponding logical formulae. Abbreviations: anon — anonymised, ground_transport — GT, quest — question,

Data set	Input	Output
ATIS	ground transport in ci0	(lambda $0 e (and GT) (to_city ci0)))
ATIS non-anon quest-split	ground transport in Denver	(lambda $0 e (and (GT $0) (to_city $0 denver)))
GEO	how many citizen in s0	(population:i s0)
GEO non-anon quest-split	how many citizens in Alabama	(population alabama)
GEO non-anon query-split	how many citizens in Boulder	(population boulder)

Table 2: Number of training (Train), development (Dev), and test (Test) examples

Data set	Train	Dev	Test
ATIS	4,434	491	448
ATIS non-anon	4,029	504	504
GEO	600	0	280
GEO non-anon quest-split	583	15	279
GEO non-anon query-split	543	148	186

(2017). Furthermore, word embeddings are randomly initialised for source and target inputs to treat them equally.

We use the Adam optimiser. The learning rate is set to 3×10^{-4}. The dimension of the self-attention model is $1,024$. From 6 to 8 encoder and decoder blocks are used. The dropout rate is 0.4 and maximum number of epochs 720 with early stopping.

We have used two semantic parsing data sets — namely ATIS with queries from a flight booking system (Price, 1990; Dahl et al., 1994; Zettlemoyer and Collins, 2007) and GEO with queries about US geographical information (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005) — for evaluation. The meaning representation of data sets is lambda calculus.

Table 3: Vocabulary (vocab) size for sentences and logical forms in the ATIS and GEO training sets. Entity anonymisation has a bigger impact in the vocab size of the input (I) than in the vocab size of the output (O).

Data set	I vocab	O vocab
ATIS	166	433
ATIS non-anon	444	887
GEO	51	120
GEO non-anon quest-split	141	243
GEO non-anon query-split	149	254

Table 4: The accuracy [%] on ATIS and GEO on anononymised test sets

Model	ATIS	GEO
Statistical Baselines		
ZC07 (Zettlemoyer and Collins, 2007)	84.6	86.07
TISP (Zhao and Huang, 2015)	84.2	**88.9**
Neural Baselines		
Seq2Seq + Attention (Dong and Lapata, 2016)	84.15	84.6
Seq2Tree + Attention (Dong and Lapata, 2016)	86.9	87.1
ASN (Rabinovich et al., 2017)	85.3	85.7
ASN + Attention (Rabinovich et al., 2017)	85.9	87.1
coarse2fine (Dong and Lapata, 2018)	87.7	88.2
Our Neurals		
Bi-GRU	85.93	86.42
Transformer	**87.95**	86.78

There are two types of data set splits: question-split and query-split. In the former, training and testing examples are divided based on questions, thus based on the input sequence. In the latter, training and test examples are divided according to the similarity of their meaning representations, thus based on output sequences. In other words, training and testing examples in query-split are strictly controlled to have a more diverse set of logical formulae. Therefore, its use is more appropriate when evaluating the model's capability to compose output sequences (i.e., lambda calculus expressions here).

For each split, data sets might contain variables with or without anonymisation (Tables 1 and 2, resulting in two versions of the first data set (i.e., ATIS question-split and ATIS question-split non-anonymised) and three versions of the second data set (i.e., GEO question-split, GEO question-split non-anonymised, and GEO query-spit non-anonymised). Versions of GEO without anonymisation are from Kennardi et al. (2019) and splits originate from Finegan-Dollak et al. (2018).

As output logical formulae cannot be partially correct, we report the exact match by computing

$$\text{Accuracy} = \frac{\text{\# of correct formulae}}{\text{\# test examples in the test set}}.$$

4 Results

Our self-attention neural semantic parsing model became the new state-of-the-art on ATIS with

Table 5: The accuracy [%] on non-anononymised (NA) ATIS and GEO test sets

Model	ATIS NA	GEO NA quest split	GEO NA query split
Seq2Seq + Attention (Dong and Lapata, 2016)	72.02	67.39	41.94
coarse2fine (Dong and Lapata, 2018)	**79.1**	72.4	52.69
Bi-GRU	73.41	72.4	56.45
Transformer	75.99	**75.27**	**63.98**

Table 6: Difficult examples

Data set	Input	Output
ATIS	fare code fb0 what doe that mean	fb0
	what type of plane is a ac0	ac0
GEO	what is the average population per square km in s0	density:i s0
	what is the length of the r0 in s0	len:i r0

its accuracy of 87.95% (Table 4). However, the best result on GEO was by a statistical semantic parser called *Type-Driven Incremental Semantic Parsing* (TISP) (Zhao and Huang, 2015). The result was explained by our model overfitting on GEO that has fewer examples than ATIS, regardless of us using a smaller self-attention model on GEO that on ATIS (i.e., 8 vs. 16 heads). As expected, regardless of the model, results entity with anonymisation were always better than without (Table 5). On ATIS (GEO), this difference was approximately 10% (at least 15%). The GEO query-split task — with more diverse input and output instances — was harder than the GEO question-split task. Results indicated that our model is capable of capturing relationships by learning token attributes as opposed to only one-to-one mappings from a token in a sentence to a token in a logical formula.

Thus, Transformer was powerful in semantic parsing. The model outperformed its baselines on ATIS, GEO question split, and GEO query split with the best accuracy values of 87.95%, 75.27%, 63.98%, respectively. Our implementation of Bi-GRU was also competitive, achieving better results than the baseline model from (Dong and Lapata, 2016) across these data and outperforming all baselines on GEO query-split. We argued that Transformers are better at capturing long distance dependencies a the model process an sentence is process as a whole, instead of word by word. However, the Transformer implemented in this research is known to have an upper limit to the distances over it can easily learn relationships (Dai et al., 2019).

Token generation was an important feature in our comparisons although theoretically the difference between Seq2Seq's Long Short-Term Memory (LSTM) and our basic model's Bi-Directional Gated Recurrent Units (Bi-GRU, which performed substantially worse) should have been minor. Seq2Seq used a greedy search[1] for token generation while all other models beam searched,[2] which tends to be a better choice for sequence-to-sequence models.

Table 6 shows example instances that were difficult for every model. There was a considerable difference between the length of input sentences (Input column) and their corresponding logical forms (Output column). This was explained by sequence-to-sequence models' tendency to not choose the end of sequence (<eos>) when beginning the generation process, because of them having learnt that logical formulae are usually longer than one or two tokens (i.e., the probability of <eos> is low in the beginning of decoding which makes the mapping from long inputs to short outputs inaccurate).

5 Conclusion and Future Work

We evaluated the Transformer architecture for semantic parsing. The model was extensively evaluated with two data sets from different domains — with and without anonymisation — across a range of complexity levels. Experiments shows Transformer is competitive with other state-of-the-art models and outperformed strong baselines in some settings.

For future work, it would be interesting to design a tree-structure self-attention model. As logical forms are tree-structures, adding some constraints in the decoder to enforce tree-based decoding would be of particular interest.

[1]Greedy search generates the next token with the highest probability relating to the current output sequence. While this strategy is suitable for the current timestamp, it may be a sub-optimal choice to construct the full output formula.

[2]Beam search has k-best output sequences each time and it considers all options of combining those sequences and all candidates in the vocabulary. Then, it chooses k-best output sequences to generate the end of sequence.

Acknowledgement

We are thankful for our co-supervised student's contribution. Namely, we express our gratitude to Xiang Li for his insight throughout his Bachelor of Advanced Computing (Honours) project (Li, 2019) in the Australian National University in 2019 that founded this study. We also thank the Australasian Language Technology Association and anonymous referees of its 2020 workshop for their helpful comments.

References

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*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Marco Damonte, Rahul Goel, and Tagyoung Chung. 2019. Practical semantic parsing for spoken language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 16–23, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Robin Jia and Percy Liang. 2016a. 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.

Robin Jia and Percy Liang. 2016b. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Alvin Kennardi, Gabriela Ferraro, and Qing Wang. 2019. Domain adaptation for low-resource neural semantic parsing. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 87–93, Sydney, Australia. Australasian Language Technology Association.

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.

Xiang Li. 2019. *Improving Semantic Parsing with Self-Attention Model. Bachelor of Advanced Computing (Honours) Thesis*. The Australian National University, Canberra, ACT, Australia.

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.

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*.

Maxim Rabinovich, Mitchell Stern, and Dan Klein. 2017. Abstract syntax networks for code generation and semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1139–1149, Vancouver, Canada. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, 31(7):1235–1270.

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.

Kai Zhao and Liang Huang. 2015. Type-driven incremental semantic parsing with polymorphism. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1416–1421, Denver, Colorado. Association for Computational Linguistics.