Yale

Tree-to-Tree Neural Semantic Parsing

Jack Koch

[1]Department of Applied Mathematics, Yale College, New Haven, CT

LILY Lab

Introduction

Recent work in deep learning has led to neural semantic parsing which aims to directly translate natural language to logical forms (Dong and Lapata, 2016), (Jia and Liang, 2016). Jia and Liang (2016) introduce data recombination for semantic parsing, where a high-precision synchronous context-free grammar is induced from the training data and subsequently sampled to generate new training examples. Thus, prior knowledge is able to be injected into the neural architecture. Dong and Lapata (2016) use a similar neural network architecture, but propose a tree-based decoder to deal with the nested nature of logical forms. This project is an attempt to extend some of this work in neural semantic parsing by testing the effects of encoding a natural language input as a tree instead of a sequence.

Materials and Methods

My sequence-to-sequence RNN model is based on on the attention-based sequence-to-sequence models presented by Jia and Liang (2016) and Dong and Lapata (2016), which were themselves based on very similar models from neural machine translation (Bahdanau et al., 2014), (Luong et al., 2015). However, I did not employ Jia and Liang's (2016) copy mechanism, and unlike these other models, I used GRU cells instead of LSTM cells (Chung et al., 2014). I made this choice because GRU cells have fewer parameters than LSTM cells and thus I hoped they would overfit the small dataset less.
My sequence-to-tree model is much the same as my sequence-to-sequence model, but with the addition of the tree decoder as presented by Dong and Lapata (2016). However, I again use GRU cells instead of their LSTM cells.
For my tree-to-tree model, I used the same tree decoder, but instead of encoding the input query as a sequence with an RNN, I generated a dependency parse tree of the natural language input using the Stanford CoreNLP parser (Manning et al., 2014), which I then feed into a Child-Sum Tree LSTM (Tai et al., 2015),

	Hidden Dimension		
Dropout rate	150	200	250
0.2	22.8	29.2	30.8
0.3	27.7	28.3	**32.2**
0.4	27.5	28.8	28.3
0.5	27.2	26.0	32.2

Table 1. Seq2Seq Average 3-fold CV Accuracy

	Hidden Dimension		
Dropout rate	150	200	250
0.2	33.5	28.5	33.0
0.3	29.5	34.2	33.3
0.4	30.8	**36.2**	31.8
0.5	31.3	30.7	31.3

Table 2. Seq2Tree Average 3–fold CV Accuracy

	Hidden Dimension		
Dropout rate	150	200	250
0.2	26.3	32.0	27.3
0.3	26.5	26.0	29.2
0.4	27.8	31.3	27.8
0.5	25.5	32.3	**33.0**

Table 3. Tree2Tree Average 3-fold CV Accuracy

Method	Accuracy
Seq2Seq	44.3
Seq2Tree	43.9
Tree2Tree	45.0

Table 4. Test results on the GEO dataset

Results

Based on Table 4, it would appear that the tree-to-tree model is superior to the sequence-to-tree and tree-to-tree models. However, none of my accuracies came close to the accuracies reported by Jia and Liang (2016) or Dong and Lapata (2016). Jia and Liang (2016) report denotation accuracy, which is a less strict measure of accuracy than sequence-level accuracy. They also use beam search to make predictions and pick the top logical form on the beam that does not yield an executor error when the corresponding denotation is computed, so I suspected their results to be incomparably high to those of my implementation.
The difference might simply be due to inadequate hyperparameter tuning, but the results from Dong and Lapata (2016) make that seem unlikely. They, like me, used sequence-level accuracy as their accuracy metric and used greedy search instead of beam search, but somehow got better sequence-to-sequence results than Jia and Liang (2016) did, even with the copy mechanism. Additionally, I did the same basic hyperparameter search Dong and Lapata (2016) report in their settings, as I mentioned in 5.2, but I still did not come close to their sequence-level accuracy on the test set.

Conclusion

In this project, I have presented a novel encoder-decoder neural network model for semantic parsing. Although I could not yet replicate the accuracies reported by others (Jia and Liang 2016, Dong and Lapata 2016) on the same dataset in my re-implementations of their sequence-to-sequence and sequence-to-tree models, a comparison between my own models indicates that the tree encoder performs slightly better than my own, possibly faulty implementation of the sequence encoder. Unfortunately, these results are not definitive, but it seems quite possible that if the sequence-based models were tuned or debugged to replicate those other accuracies, the tree-to-tree model could very similarly be modified to maintain its superiority over the others.