Tree-Based Semantic Parsing

Alexander Fabbri,[1] Dragomir Radev PhD[1]

[1]Department of Computer Science, Yale University

LILY Lab

Introduction

Semantic parsing is the process of mapping natural language sentences to formal meaning representations. Semantic parsing techniques can be performed on various natural languages as well as task-specific representations of meaning. A semantic parser can be learned in a supervised or semi-supervised manner in which the natural language sentence is paired with either a logical form or its denotation. The logical forms may be database queries, dependency graphs, lambda-calculus terms, among others. A common approach has been to induce semantic parsers from data with the use of a probabilistic grammar such as a combinatory categorical grammar. However, recent advances in deep learning have led to neural semantic parsing in the form of encoder-decoder networks which have been used for tasks such as machine translation. Such sequence to sequence models are usually combined with an attention mechanism which weighs the input according to its importance at the current decoding process timestamp. A recent work enhanced this architecture by constraining the decoding process to output a tree rather than a sequence. The motivation behind this choice is the compositional nature of the logical forms the model is supposed to output. With this in mind, we aim to test the influence of encoding the input as a tree. Recursive Neural Networks take as input a constituency or dependency tree of an input sentence and recursively build a representation of the sentence. These models have been used for tasks such as a fine-grained sentiment analysis and textual entailment and more recently to machine translation, with improved results on a Chinese-English corpus.

Materials and Methods

The main data throughout out experiments was the GeoQuery dataset developed by Ray Mooney's group at the University of Austin. This dataset contains natural language queries about U.S. geography and their associated Prolog queries. The dataset consists of 880 total examples, usually split into a training and tests sets of size 680 and 280. We follow a common preprocessing technique for the dataset and use De Brujin index notation for variable-name standardization.

```
Input Query:
How many states border the state that
borders the most states ?

Logical Form:
_answer ( NV , _count ( NV , ( _state
( V0 ) , _next_to ( V0 , NV ) , _most
( V0 , NV , ( _state ( V1 ) , _next_to ( V1
, V0 ) , _state ( V0 ) ) ) ) , V3 ) )

Constituency Parse Tree:
(ROOT
  (SBARQ
    (WHNP
      (WHADJP (WRB How) (JJ many))
      (NNS states))
    (SQ
      (VP (VB border)
        (NP
          (NP (DT the) (NN state))
          (SBAR
            (WHNP (WDT that))
            (S
              (VP (VBZ borders)
                (NP (DT the) (JJS most)
(NNS states))))))))
    (. ?)))
```

Figure 1. Example query-logical form pair from the GeoQuery dataset as well as its corresponding constituency parser provided by Stanford CoreNLP which demonstrate the compositional nature of the dataset.

We adapt a syntax aware encoder-decoder model used in machine translation for our task of semantic parsing. The input query is parsed to a constituency tree and then binarized. We make use of Stanford CoreNLP tools for parsing. The encoder decoder model uses Tree-GRU units, analogous to the Tree-LSTM units originally used for Recursive Neural Networks. Tree-GRU units build upon vanilla GRU units by combining input from the children nodes. In addition to the bottom-up method which builds a representation of the subtrees at the root node, our model uses a bidirectional encoding which combines the bottom-up approach with a top-down encoding. The motivation for this addition is that the leaf nodes are only encoded using sequential information and do not receive any syntactic information from higher up in the tree. This draws on sequence to sequence models which often employ a bidirectional encoding for improved results. An attention mechanism is used over the encoding. As the non-leaf nodes encode more information than leaf nodes, the model prefers to attend over non-leaf nodes. This can lead to repetition in the decoding process. To alleviate this problem, a coverage mechanism prevents the model from consistently attending to the same encoding part by keeping track of the nodes attended to previously.

$$z^i = \sigma\left(U^{z^i} h_1 + V^{z^i} h_2\right), \qquad i = 1, 2$$

$$r = \sigma\left(U^{r} h_1 + V^{r} h_2\right)$$

$$\bar{h} = \tanh\left(U^{h}(h_1 \odot r) + V^{h}(h_2 \odot r)\right)$$

$$h = (\mathbf{1} - \sum_{i=1}^{2} z^i) \odot \bar{h} + \sum_{i=1}^{2} z^i \odot h_i$$

Figure 2. Binary Tree-GRU - equations for the Tree-GRU used in Recursive Neural Networks on constituency parses

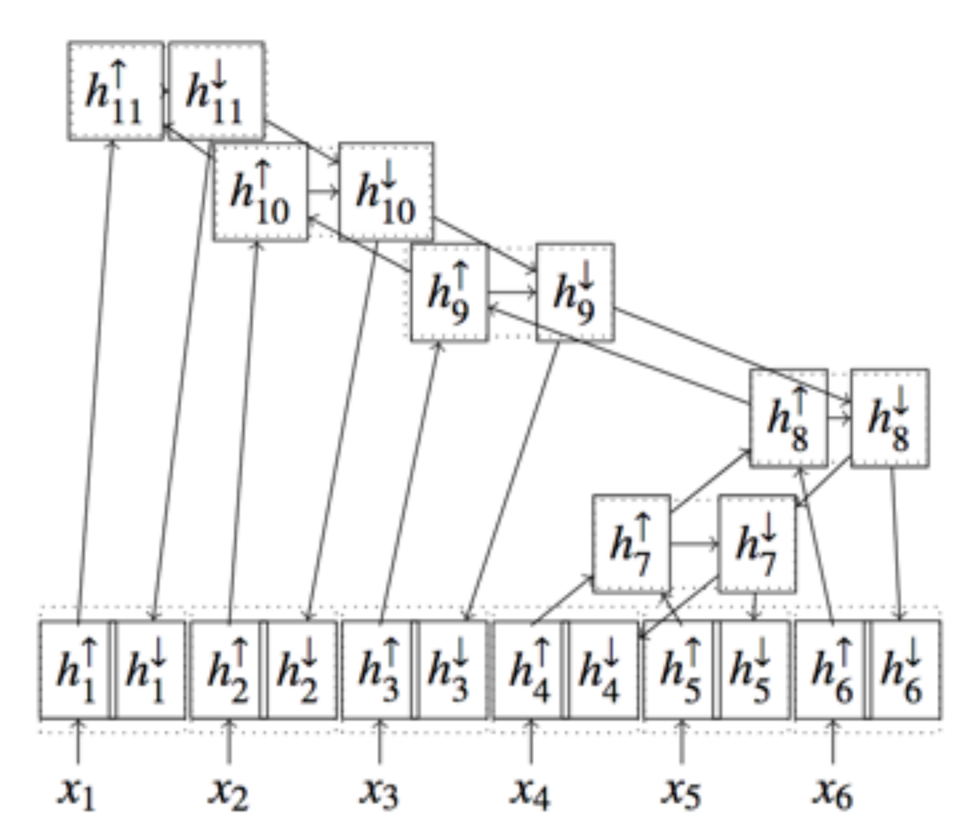

Figure 3. Bidirectional Tree Encoder - the encoding for a sentence is calculated in a combined bottom-up and top-down manner

Experiments

We perform a hyper-parameter search over the size of the word embeddings {100,150,200,250,300}, the recurrent layer size (same as word embeddings) as well as the dropout probability used {.2, .3, .4, .5}. This was done via 5-fold cross validation on the 680 training examples. We do not use a cutoff for word frequency or maximum length of the query. All out-of-vocabulary words are mapped to a special token UNK. We use RMSProp for optimization, a batch size of 20 and clip gradients at 5. We evaluate the models based on sequence accuracy, token accuracy and denotation accuracy. Initial results have not been positive. The parameters which give the best results (word embeddings of size 300 and dropout .5) output a token accuracy of ~68%, while state-of-the-art methods perform ~90%.

Conclusion and Future Work

We attempt to adapt a model that gives improved results in the task of machine translation to the task of semantic parsing. Although initial results have not been promising, improvements can be made through a more extensive hyper-parameter tuning as well as refinement of the architecture. The original paper on machine translation on which the experiments were based uses much larger word embeddings (512 dimensions), hidden layer size (1024 dimensions) and vocabulary size (30,000 most frequent words). Also, while the attention mechanism includes coverage, it does not include a copying or pointer-network mechanism, which has been shown to improve results in such tasks. Often where the model predicts the the majority of tokens correctly, the model incorrectly outputts named-entities which can be copied from the source input. Additionally, tests should be performed on other semantic parsing datasets, particularly ones of larger size such as OVERNIGHT and WikiSQL which contain 26,098 and 80,654 examples respectively.

Acknowledgement

Thank you to Professor Radev for his insightful advising and for the fellow LILY Lab students whose discussion and advice helped me throughout the semester. Images from: https://link.springer.com/content/pdf/10.1007/978-3-319-64206-2_9.pdf, https://arxiv.org/pdf/1707.05436.pdf