Constituency-Tree Recursive Neural Network for Quiz Bowl Answering

Chuan Tian
Department of Energy Resources Engineering
Stanford University
Stanford, CA 94305
chuant@stanford.edu

Wenyue Sun
Department of Energy Resources Engineering
Stanford University
Stanford, CA 94305
wenyue@stanford.edu

Abstract

Quiz bowl answering represents a challenging task in natural language processing where few indication entities of answers are given in the questions. Traditional approaches such as manual feature engineering and bag of words representations can fail to reason the correct answers due to such lack of direct indications. Recently recursive neural network (RNN) model shows promising performance in quiz bowl answering by outperforming multiple baselines. In this work, we introduce a constituency-tree recursive neural network (CT-RNN) and evaluate its performance on a public dataset.

1 Introduction

Factoid question answering is a type of problem of identifying an semantic entity from given question. There is one kind of factoid question answering called *quiz bowl*, where the participants make guesses on the entity based on given descriptions. The descriptions normally consist of several sentences, which contain more and more specific information about the target entity. However, the answers should be never mentioned explicitly in the descriptions. Below is an example of the question about the German composer Johann Sebastian Bach:

This composer was given the task of improvising on a "Royal Theme" provided to him by a monarch, which eventually resulted in his piece The Musical Offering. The fifth part of another work by this composer contains a solo for virtuoso harpsichord. That work was written for the margrave of a certain city. Name this German composer whose works include the organ piece Toccata and Fugue in D minor, The Well-Tempered Clavier, and the Brandenburg Concertos.

Quiz bowl represents a challenging task in Natural Language Processing (NLP), because of the rich compositional information in the descriptions. Traditional approaches are focused on using a bag of words (Boyd-Graber et al., 2012), which may fail due to the limit window size. More recently, Iyyer et al. (2014) proposed to use a dependency-tree recursive neural network (DT-RNN) approach for quiz bowl answering. Their method was shown to learn the representations across sentences and outperformed multiple baselines. Although DT-RNN was shown promissing, there have been no other neural network methods applied to quiz bowl answering so far.

In this work, we apply CT-RNN on a public quiz bowl dataset. CT-RNN is another RNN approach based on the constituency tree and provides an alternate parse tree structure for sentence representation. Unlike DT-RNN where internal nodes are associated with words, CT-RNN builds up the tree by combining leaf words to higher level internal nodes. Such characteristics of CT-RNN make it a suitable choice for quiz bowl answering. An public dataset from Iyyer et al. (2014) is used to thoroughly test the performance of CT-RNN. Finally we compare the performance of CT-RNN with baselines including BOW and DT-RNN, and analyze the results by visualizing the predictions and embedding space.

2 Related Work

The traditional approach for quiz bowl answering, or more broadly, factoid question answering, is manual feature engineering and hand-crafted pattern matching (Bilotti et al., 2010; Shen, 2007; Wang, 2006). A key drawback of such approach is that it heavily relies on the linguistics knowledge beforehand and can be rarely generalized well to other languages. Another NLP technique that has been applied for quiz bowl answering is the bag of words (BOW) method (Boyd-Graber et al., 2012). Although BOW is able to learn some co-occurrence structure among words, it can hardly reason the meaning of long text due to the limit of window size, and can easily fail to reply the correct answers if the indication entities associated with the answer are fuzzy.

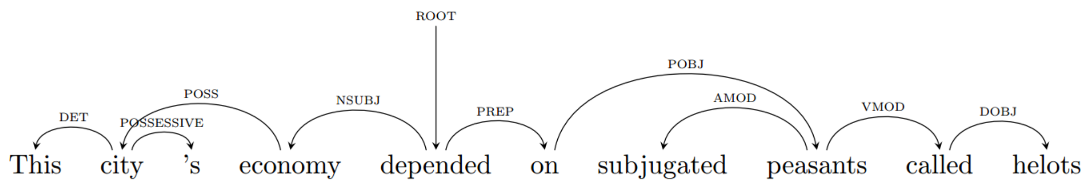

Figure 1: Dependency tree of a sentence from a question about Sparta (Iyyer et al., 2014).

The successful application of RNN in NLP draws the attention from researchers to use RNN for quiz bowl answering. The work from Iyyer et al. (2014) is among the pioneer explorations in this direction. They use dependency parse tree to structure the sentence level representation. An example of the dependency tree structure of a sentence from quiz bowl question is shown in Figure 1. By learning the word, phrase and sentence level representations to reason about entities, their method outperforms multiple strong baselines. Although DT-RNN seems promising, it requires the same answers to appear multiple times to training the model (twelve times on average, from Iyyer et al. 2014), and the prediction accuracy is still below half based on a single sentence description, which are both areas that worth further investigations. Furthermore, there has been no RNN models other than DT-RNN examined on quiz bowl answering. One of the RNN models that has been widely used in NLP, constituency-tree RNN, uses the semantic structure of the sentence as its meaning representation. It would be very interesting to investigate how the CT-RNN will help in quiz bowl answering, and to obtain a deeper understanding about the performance of various RNN models by comparing the results from CT-RNN and DT-RNN.

3 Technical Approach

CT-RNN model takes the constituency parse tree from the Stanford Parser (`http://nlp.stanford.edu/software/lex-parser.shtml`). Figure 2 shows the constituency parse on the same sentence as in Figure 1. All the words sit on the leaf nodes, and higher level nodes are created by combining the nodes below. Such process continues up to the root to generate semantic representations of the whole sentence.

Each word w in a sentence is represented by a word vector $x_w \in \mathbb{R}^d$, which is stored in a word embedding matrix $W_e \in \mathbb{R}^{d\times|V|}$. An internal node $h \in \mathbb{R}^d$ is given by

$$h = max(\bar{h_c}W + b_1, 0),$$

where $\bar{h_c}$ is the average vector representation of all the subtrees, W is a weight matrix of dimension $d \times d$, and b_1 is a bias term of dimension d. Such calculation is repeated up to the root.

With the vector representation h, a probability distribution $\hat{y} \in \mathbb{R}^{|V|}$ can be calculated using

$$\hat{y} = softmax(hU + b_2),$$

where $U \in \mathbb{R}^{d\times d}$ is a weight matrix, and $b_2 \in \mathbb{R}^d$ is a bias term. Given a one-hot vector $y \in \mathbb{R}^{|V|}$ represented the true label, we calculate cross entropy loss as

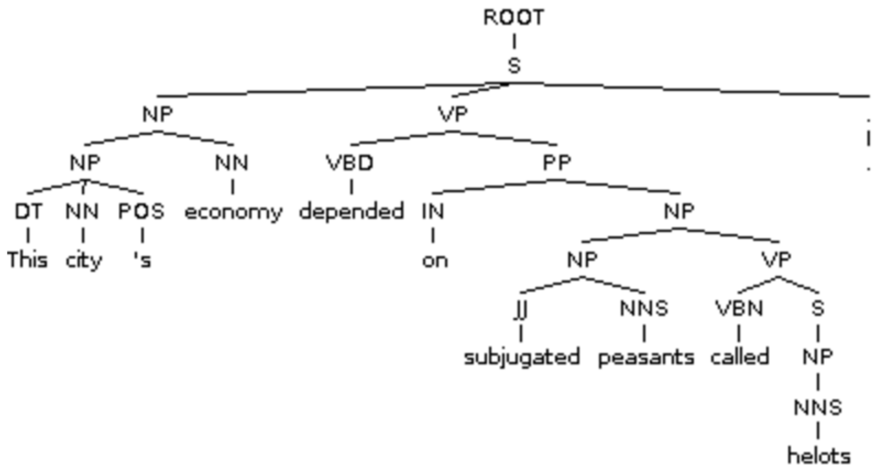

Figure 2: Constituency tree of the same sentence as shown in Figure 1.

$$CE(y, \hat{y}) = -\sum_{i=1}^{|V|} y_i log(\hat{y_i}).$$

That finishes the loss calculation for a single node. Because there is only one true answer corresponding to a sentence in the question, all the nodes in that sentence share the same label y. We define $s \in S$ to be a node in the constituency tree, where each node is associated with the predicted probabilities y^S. Thus the loss for a sentence can be calculated by summing up the cross entropy loss for all the nodes in that sentence

$$C(S, \theta) = \sum_{s \in S} CE(y^S, \hat{y^s}).$$

Then the objective function is given by the sum of errors over all sentences T

$$J(\theta) = \sum_{t \in T} C(t, \theta),$$

where parameters $\theta = (W_e, W, U, b_1, b_2)$. The gradient of the objective is computed using auto-differentiation in *TensorFlow*.

4 Experiments

4.1 Dataset

The experiments are tested on tested on a public dataset from Iyyer et al. (`http://cs.umd.edu/~miyyer/qblearn/`). The dataset contains 20407 questions about 2347 answers, where each question is a paragraph of descriptions about the answer entity. That is a different version of the dataset used by Iyyer et al. (2014), which contains 10145 question-answer pairs on history and literature knowledge. We preprocess the data to exclude the categories with less answers and the answers that appear less than six times, which gives us 7184 questions on four categories: fine art, history, literature and social science. Quality check is also performed on the data to make them ready for parsing, for instance, adding period to the end of a question if there is none, so that the parser won't mix the current question with the next one.

4.2 Results

We looked at the prediction with CT-RNN for a particular example with the correct answer as 'jefferson davis'. The first sentence is:

he resigned from the senate in 1851 to run for governor but was defeated by henry foote.

The method returns the first three guesses as 'coxey's army', 'jefferson davis', and 'monroedoctrine'. We can see that the first guess is wrong in this case, and the guesses 'coxey's army' and 'monroedoctrine' are even not name entity. However, with the section following sentence:

five years earlier he had resigned from the house to command the mississippi rifles in the mexican-american war.

The three high rank results became: 'jefferson davis', 'coxey's army' and 'george armstrong custer'. The method provide the right guess and have new guess 'george armstrong custer' as a person entity that is intuitively closer to be the right answer compared with 'monroedoctrine'. This updated results indicate that our method CT-RNN is able to show improved prediction with more sentences in this case.

Figure 3 shows the loss history for our CT-RNN method. We see that the loss display a reduction at early iterations (around 600); however, the loss functions remain similar afterwards. We expected our implemented CT-RNN method to provide reasonable level of accuracy of the prediction over the training set; however, the final training and test accuracy were found to be relatively low and were not comparable with those with Qanta. We carefully looked into the codes to remove all potential coding bugs and tried the selection of different hyperparameters and even loss functions, while the final results were still unsatisfactory. Some suspects are summarized below:

1. We generated all constituency trees with Stanford parser on raw quiz bowl data. However, the raw text data were not organized very clean. Lots of the punctuations are used and inserted inappropriately, which severely impacts the parsing quality. Though we preprocess the raw text data, some of the resulting constituency trees generated might still be problematic.
2. There are more layers in the constituency tree compared to that for the dependency tree. The benefit of having more layers is that constituency tree can better represent the sentence. The disadvantage is that the training objective function becomes highly nonlinear to the wording embedding matrix that are used to represent the root node, thus, the optimizer (we use the gradient descent algorithm in Tensorflow) is not able to find a good minimum.

5 Conclusion

1. We build a CT-RNN model for quiz bowl answering. Based on our literature review, this is the very first work that CT-RNN is used for quiz bowl answering. The model is tested thoroughly on a public dataset.
2. CT-RNN functions reasonably well with decreasing loss function and generating predictions based on given questions. By examining the predicted answers, we find that CT-RNN learns the underlying meaning of the question content and predicts entities related to the questions. In addiction, CT-RNN shows higher prediction accuracy with as it sees more sentences in the questions.
3. Current implementation of the CT-RNN method yield inferior results compared with those from Qanta; however, we are still very interested in further exploring some potential adjustments and improvements for the CT-RNN method.

References

[1] Matthew W. Bilotti, Jonathan Elsas, Jaime Carbonell, and Eric Nyberg. 2010. Rank learning for factoid question answering with linguistic and semantic constraints. In *CIKM*.

[2] Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daume III. 2012. Besting the Quiz Master: Crowd-sourcing Incremental Classification Games. In *EMNLP*.

[3] Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daum III. 2014. A Neural Network for Factoid Question Answering over Paragraphs. In *EMNLP*.

[4] Dan Shen. 2007. Using semantic role to improve question answering. In *EMNLP*.

[5] Mengqiu Wang. 2006. A survey of answer extraction techniques in factoid question answering. *Computational Linguistics*, 1(1).

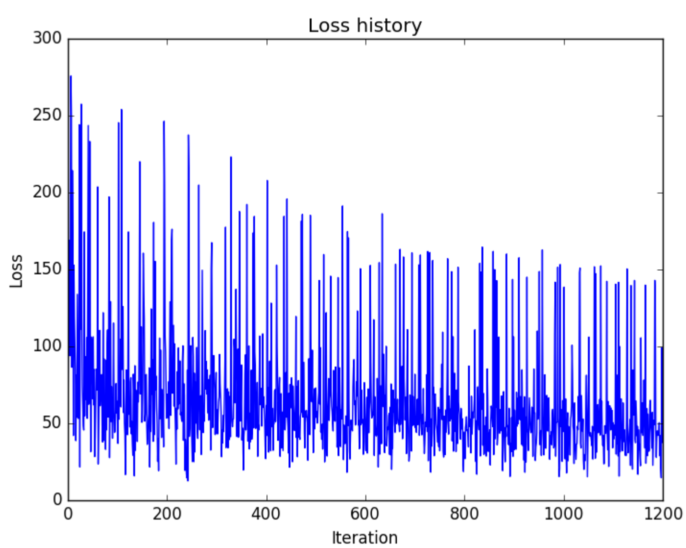

Figure 3: Loss history of CT-RNN