

Building a QA system (IID SQuAD track)

Stanford CS224N Default Project

Daniel Xia

Institute of Computational and Mathematical Engineering
Stanford University
jx643@stanford.edu

Abstract

Question Answering is a interesting machine learning task which shows how machine can understand the relationship and the meaning of the words. There are lots of existing models built to solve this task. This paper draws inspiration from the paper Bidirectional Attention Flow for Machine Comprehension [1] and dive deeper into the effect of character level embedding on the performance of the model. Through experimenting on different CNN model for character level embedding, we have concluded that a more complex CNN model does not result in a better performance metrics. However, through manually evaluate the model's prediction, we have found that a more complex model does perform better in certain cases.

1 Introduction

With the task of question answering, the model will be given a paragraph and a question related to the paragraph as the input, and the goal is to answer the question correctly with some sub-part of the paragraph (no external knowledge needed to answer the question). Question answering is a interesting machine learning task since it shows how good a model can dissect and understand a paragraph as well as the question. As one of the most frequently used data set to train and validate question answering model, SQuAD data set collects paragraph from Wikipedia and crowd source questions and answers using Amazon Mechanical Turk. There are numerous model built based on this data set and showed some promising result. This project is built based on one of the existing Bi-directional Attention Flow model and draw inspiration from the paper Bidirectional Attention Flow for Machine Comprehension [1]. In this project, we will dive deeper into the character level embedding and its effect on the model performance. We will experiment different kinds of character level embedding and compare their model result and perform an analysis on the results.

In general, adding a character level embedding layer to the existing Bi-directional Attention Flow model improve the performance of the model. While adding a such layer might result in more computation, the evaluation matrix shows that the new model has a better performance than the model with word level embedding only. Although we expect the embedding would work better with a more complex structure, the result of our experiments shows that the performance does not improve much from just viewing the evaluation matrix. However, with a more qualitative evaluation of the output, we can still find that with a more complex character level embedding, the model can perform a little better in certain cases. There is no quantitative result that which model might be better suit for the task, but the result has proven the point that there are still room for improvement in developing a better character level embedding for the question answering task.

2 Related Work

The project is inspired by the paper Bidirectional Attention Flow for Machine Comprehension [1]. In the paper, they have included a character level embedding layer into the model. Such layer maps each word to vector space using a character level CNN. In the paper, they have mentioned that the the CNN is inspired by the paper by Kim [2]. The model will take the vector representation of the word in

character level as a 1D input to the CNN and the output of the CNN is then max pooled over the size to get a constant length embedding output. Then the character level embedding is concatenate with the word level embedding and construct the complete embedding for the rest of the model. The result from their experiment has shown that character level embedding also contribute to the performance of the model. The paper has mentioned that character level embedding can handle out of vocabulary word or rare frequency word better. This paper has inspire me to dive deeper into how the character level embedding affect the model as a whole. In the paper, they have used the CNN described in Kim's paper [2] and is a simple one layer CNN and we would like to see if a more complex CNN will result in a better performance.

3 Approach

For this project, we will adapt the general structure of the model from the Bidirectional Attention Flow for Machine Comprehension[1] paper and alter the character level CNN to experiment. So the model architecture will follow the original model in general. Figure 1 is a figure adapted from the paper [1] and it also represent the general structure of our model.

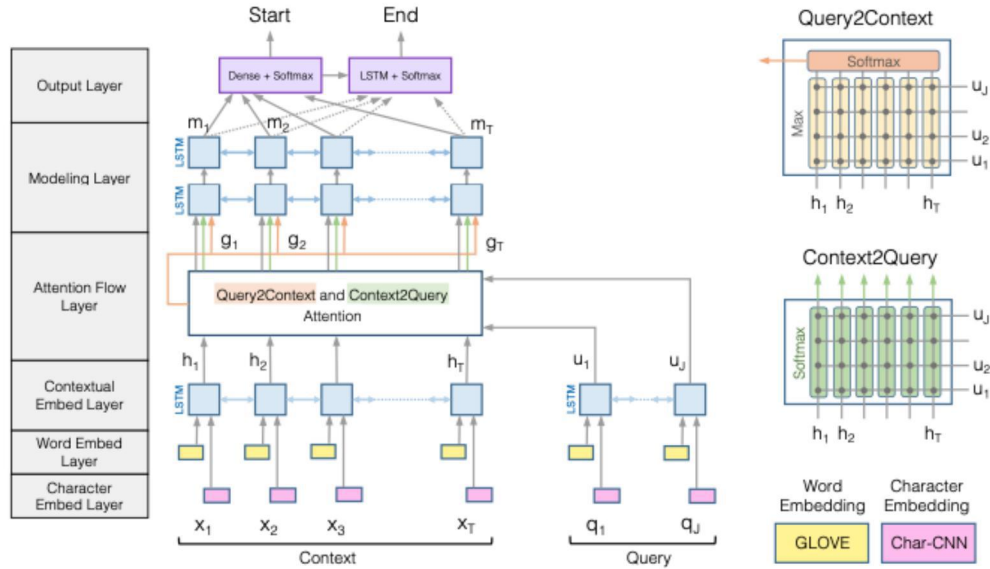


Figure 1: Model Architecture from Bidirectional Attention Flow for Machine Comprehension[1]

From the figure, we can see that there are several important layers that we need to take notes on. We will only roughly introduced these layers in this paper since the details of the layers can be found in the original paper[1] but we will go over the part what might be different from the original model.

- **Embedding Layers:** There are two part of the embedding layers, the first is a pre-trained GloVe word embedding and the second is the character-level CNN embedding. The embeddings from each part are then concatenated to form the complete embedding. The embedding are further refined by a Highway Network [3]. We use a two-layer highway network to transform each embedding. In paper[1], the Char-CNN layer follows the general structure of the CNN from Kim's paper [2]. The network is formed by a Conv1d followed with a ReLU activation function and a max-pooling layer. We will experiment with different structure of CNN layer to test the performance.
- **Contextual Layer:** We use a Long Short-Term Memory Network (LSTM)[4] on top of the embeddings from previous layers to model the temporal interactions between words. LSTM is applied to both direction and the outputs are then concatenated.

- **Attention Layer:** We use a bidirectional attention flow layer which allows the attention to flow from context to question and from the question to context. Details of the attention layer is already covered in paper [1], so we will not go over in this paper. We will use the same computation as the attention layer described in the paper [1].
- **Modeling Layer:** The output from the previous attention layer encodes the query aware representations of the context words and we use a two layer bi-directional LSTM. Each column of the output matrix should contains the information about the word with respect to both context and query.
- **Output Layer:** The output layer will produce a vector of probabilities corresponding to each position in the context. The layer will take the output from attention layer and the modeling layer . The output from the modeling layer will go through a bidirectional LSTM layer and the output will from the LSTM will then be used with the output from the attention layer to compute the probabilities in log-space for stability purpose.
- **Loss Function:** The loss function is the cross-entropy loss for the start and the end location. Given the correct answer has the start location i and end location j . Then the loss of a single example is:

$$loss = -\log p_{start}(i) - \log p_{end}(j)$$

During training, Adadelta optimizer [5] is used to minimize the loss.

- **Predicting no-answer:** The data contains questions that does not have exact answer available in the context, so there are questions with answer "N/A". OOV(Out of Vocabulary) token are placed to the beginning of each context and if the model predict that the pair of the position $(0, 0)$ has the greatest probability, the model will predict no-answer.

4 Experiments

4.1 Data

The data we are using is from the official SQuAD 2.0 data. The data contains paragraphs from Wikipedia and questions and answers crowd-sourced using Amazon Mechanical Turk. Half of the examples have questions that cannot be answered with the provided paragraph. The rest of the questions can be answered with a chunk of text directly from the paragraph.

The data is split in to three parts. Training set contains 129,941 examples all taken from the official SQuAD 2.0 training set. Dev set contains 6078 examples, which are randomly selected from the official dev set. The remaining examples from the official dev set , plus some had-labeled examples are the test set (5915 examples). The model will be trained on the training set and will be evaluated and compared regarding their performance on dev set and test set.

4.2 Evaluation method

Performance is measured via two metrics: **Exact Match (EM)** score and **F1** score.

- **Exact Match** is a strict binary measure of whether the output of the model matches the ground truth answer exactly. Score of 1 means exact match and 0 means the answer and the output are different.
- **F1** is the harmonic mean of precision and recall and it is a less strict metric. A higher F1 score means a greater similarity between the answer and the output.

We will also generate the answer for some of the test example to evaluate the output of the model manually to analyze the performance of the models.

4.3 Experimental details

The models are trained with a constant learning rate of 0.5 and are trained over 30 epoch. We will train three different models. The baseline model is the model without any character level embedding. The single-layer model is the model with a single layer CNN character level embedding and the two-layer model is the model with a two layer CNN character level embedding. All the convolution layer in the CNN are one dimensional and the activation function used are ReLU. The models are expected to be trained for around 12 hours.

4.4 Results

During the training process, we have also plot out the corresponding loss function of the model on the dev set. As shown in Figure1, all the three model have their loss function reduced during the training and the model tends to over fit to the training set at the end of the training process.

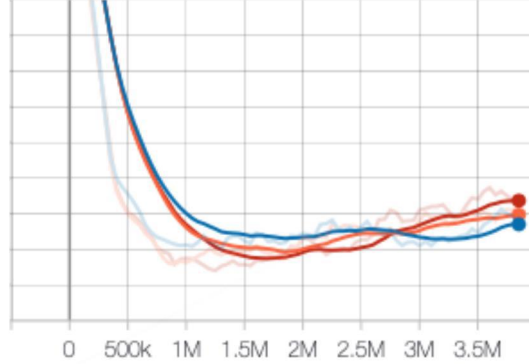


Figure 2: Loss on dev set against the number of examples trained

With the help of the validation set, we select the best model parameters and the following table 2 shows the result for the model on the dev set. It is as expected that with a character level CNN layer,

Model	Training Time	EM score	F1 score
Baseline	11h40m	58.057	61.286
One-layer CNN model	14h36m	59.234	62.592
Two-layer CNN model	18h14m	58.646	61.839

Table 1: model metrics on dev set

the training time will increase and the training time increases with the complexity of the CNN model as more parameters are computed and learned. We can see that with the addition of the character level CNN layer, the model has a better performance in terms of the evaluation metrics. Although it is not as expected that a more complex character-level embedding layer result in a worse performance in terms of the evaluation metrics.

The models are also evaluated in with the test set and the resulting metrics are as follow: The result is

Model	EM score	F1 score
One-layer CNN model	59.915	63.822
Two-layer CNN model	58.478	62.068

Table 2: model metrics on test set

similar to the the ones from the dev set where a simpler character level embedding result in a better performance. The cause of it might be the fact that the question answering task is mainly based on the relationship between the words and a over complicated character level embedding might result in a deviation from purpose of the model. However, as we have look deeper in the result of the model in the next section, we can see that a two-layer CNN model actually does a better job in some circumstances.

5 Analysis

In this section, we will analyze the output of the model manually and try to understand the performance of the model in a more detailed ways.

Questions: What kingdom annexed Duchy in 1796?

Context: Warsaw remained the capital of the Polish–Lithuanian Commonwealth until 1796, when it was annexed by the Kingdom of Prussia to become the capital of the province of South Prussia. Liberated by Napoleon's army in 1806, Warsaw was made the capital of the newly created Duchy of Warsaw. Following the Congress of Vienna of 1815, Warsaw became the centre of the Congress Poland, a constitutional monarchy under a personal union with Imperial Russia. The Royal University of Warsaw was established in 1816.

Answer: N/A

Baseline Prediction: N/A

One-Layer Prediction: N/A

Two-Layer Prediction: Kingdom of Prussia

Analysis: This is an example showing that a complex character level model predicts wrongly. In this example we can see that the model incorrectly predict because "Kingdom of Prussia" does have the action of "annex" but the model does not take account the subject of the action so that the model incorrectly predict "Kingdom of Prussia" as the answer.

Questions: What president eliminated the Christian position in the curriculum?

Context: Charles W. Eliot, president 1869–1909, eliminated the favored position of Christianity from the curriculum while opening it to student self-direction. While Eliot was the most crucial figure in the secularization of American higher education, he was motivated not by a desire to secularize education, but by Transcendentalist Unitarian convictions. Derived from William Ellery Channing and Ralph Waldo Emerson, these convictions were focused on the dignity and worth of human nature, the right and ability of each person to perceive truth, and the indwelling God in each person.

Answer: Charles W. Eliot

Baseline Prediction: N/A

One-Layer Prediction: Charles W. Eliot

Two-Layer Prediction: Charles W. Eliot

Analysis: This is same as expected as character level embedding works better for out of vocabulary words or rare words. For predictions involving specific subjects and places, the model with character level embedding will perform better in prediction. Predicting the full name also shows that the model understand the relationship between different part of the name well.

Questions: When had the Brotherhood renounced violence as a means of achieving its goals?

Context: While Qutb's ideas became increasingly radical during his imprisonment prior to his execution in 1966, the leadership of the Brotherhood, led by Hasan al-Hudaybi, remained moderate and interested in political negotiation and activism. Fringe or splinter movements inspired by the final writings of Qutb in the mid-1960s (particularly the manifesto Milestones, a.k.a. Ma'alim fi-l-Tariq) did, however, develop and they pursued a more radical direction. By the 1970s, the Brotherhood had renounced violence as a means of achieving its goals.

Answer: By the 1970s

Baseline Prediction: 1970s

One-Layer Prediction: 1970s

Two-Layer Prediction: By the 1970s

Analysis: In this example, the two layer model actually understand the test better than the one layer model. Showing that it has a deeper understanding of the word such as "by" which has an important effect indicating the time if before a certain point. This is a good example showing that a more complex character level embedding actually help the model to understand the relationship between the word better.

Questions: Economy, Energy and Tourism is one of the what?

Context: Subject Committees are established at the beginning of each parliamentary session, and again the members on each committee reflect the balance of parties across Parliament. Typically each committee corresponds with one (or more) of the departments (or ministries) of the Scottish Government. The current Subject Committees in the fourth Session are: Economy, Energy and Tourism; Education and Culture; Health and Sport; Justice; Local Government and Regeneration; Rural Affairs, Climate Change and Environment; Welfare Reform; and Infrastructure and Capital Investment.

Answer: current Subject Committees

Baseline Prediction: N/A

One-Layer Prediction: current Subject Committees

Two-Layer Prediction: current Subject Committees in the fourth Session

Analysis: This is an example showing the flaws of the model as well as the evaluation metrics. If we read the context we can see that the exact answer to the question should be "current Subject Committees in the fourth Session" where the second part "in the fourth Session" is an important part of the answer. From the prediction we can see that the two layer prediction model actually predict the answer correctly but the standard answer actually misses the important part of the answer. This shows that the examples actually have their flaws since sometimes one single answer might not be a perfect answer and multiple answers should be provided in order to have a better evaluation. Also, this is also an example showing that a more complex character-level embedding gives the model a better understanding of the text than a single layer model.

6 Conclusion

Through experiment, we have found out that character level embedding help the machine to understand the meaning of the word and the relationship between the words better, especially for words with lower frequency. However, not as we expected, a more complex model actually does not result in a better performance metrics. Through manually evaluate the prediction result, we can see that a more complex embedding actually learn things that are not captured by a simple model, so it is worth to dive deeper into a better character level embedding structure.

References

- [1] Ali Farhadi Minjoon Seo, Aniruddha Kembhavi and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.
- [2] Yoon Kim. Convolutional neural networks for sentence classification. *EMNLP*, 2014.
- [3] Klaus Greff Rupesh Kumar Srivastava and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.
- [4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [5] Matthew D Zeiler. Adadelata: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.