Quora Question Duplication

Harish Karnick
Course Instructor - CS671A

Abhishek Maher (14021) **Jatin Gupta** (14280) **Raushan Joshi** (14537)

Sachin K Salim (14575) **Shivam Yadav** (14655) **Rishav Raj** (150587)

Abstract

Quora is a question-and-answer platform where users ask and answer questions of diverse topics. One of the challenges faced by the Quora community is the presence of multiple questions with the exact or similar meaning. Thus, detecting whether two questions are semantically similar or not helps in organizing the content by merging the pages of those questions.
We tested out three different approaches in order to detect this semantic similarity. First one was Bidirectional LSTM with Attention. Then we tested out a Time Distributed Layer with Concatenation approach and finally, LSTM with Multiplication.

1 Introduction

Understanding relatedness of sentences semantically has the potential of organizing (or in a somewhat vague sense, understanding) large portions of user generated content on the web. And this area of comparing sentence similarity has a particularly high usage when it comes to removing or merging similar content on knowledge bases such as Quora.

So, in this paper, we tried to detect the semantic similarity between sentences, *i.e.*, when provided with a pair of questions, the models discussed in this paper try to detect whether they are semantically similar on not.

Now, there are various challenges to be tackled while trying to solve this problem. Questions might have a very similar structure even if they are semantically different, like the pair (*What are natural numbers?, What is the smallest natural number?*). Negative examples may also have subtle semantic differences like handling different scopes of the same event. An example would be that asking when something happens is not the same as asking whether that thing happens. Another challenge would be recognizing questions formed using different words and phrases but having the same semantic, meaning that the questions would elicit the same answer. An example of such kind would be the pair (*Is watching too much television bad?, Should I not spend excess time in front of tv?*).

We implemented three different models, already mentioned in the abstract above, and tried to compare them while analyzing their results.

2 Related Works

Various methods have been proposed to solve this problem. The Quora Question Duplication paper [3] by Elkhan *et al.* explored this problem using LSTM networks. It used a Siamese architecture where the representation of both sentences were learnt from an LSTM. Another approach mentioned

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

in the paper was a sequence-to-sequence model that uses two LSTMs with separate parameters where the last state of the first LSTM is passed to the second.
Initially, Quora used random forests on extracted features to perform the classification task. However, in a Semantic Question Matching with Deep Learning blog at Engineering at Quora, the author speaks about three different experimental approaches to solve the problem. Two involve LSTM networks to perform classification after either concatenating the embeddings or extracting the Euclidean distance and angle as features from them and the third was an attention based approach.
The attention-based approach combined neural network attention with token alignment, commonly used in machine translation. The most prominent advantage of this approach, relative to other attention-based approaches, was the small number of parameters. This model represents each token from the question with a word embedding. Training was done on example sentences using different scoring functions: additive scoring, bilinear scoring and bilinear scoring adding the original embeddings.

3 Dataset

We used the Quora question pair dataset released by Quora for identifying duplicate questions. There are over 400k lines of potential duplicate pairs. Each line contains ID for each question in the pair, two questions and a value(0 or 1) depending on whether they are semantically same or not.

3.1 Words

There was close to 1 lakh (95,596) unique words in the dataset. We assigned a numeric ID to each of these word for easier identification. A mapping was made between between these IDs and their corresponding GloVe vector.

3.2 Sentences

The questions varied in length ranging from 1 word to 200 words. We transformed each question to a fixed length (25) by padding shorter ones and truncating longer ones.

4 Approach

4.1 Embeddings

GloVe (Global Vector) pre-trained word vectors were uses to initialize the word embeddings. GloVe is a corpus count based vector based on global co-occurrence statistics.
The data was downloaded from the official webpage of GloVe in stanford website. The word vectors were trained on Wikipedia corpus and had 300 dimensions.

4.2 TimeDistributed Layer with Concatenation

First we defined an Embedding layer to apply the GloVe embedding to transform each word in the vocabulary to a 300 dimensional vector.
Then a regular densely-connected Neural Network layer with ReLU activation and with dimensionality of output space 300 is defined. A TimeDistributed wrapper is applied on the dense layers so as to generate a many-to-many transformation.
Representations of both questions are then concatenated and fed to a Deep Neural Network for further processing.

4.3 LSTM with Multiplication

We used the counts of each word in the dataset to embed each word to a 10k sized vector.
An Embedding layer is then used to transform each word to 300 dimensional vector. It is then passed through an LSTM network to generate a 256 dimensional vector.
The above procedure is done on both questions and then they are multiplied element-wise. 2 simple layers of dense networks are then applied to output the prediction.

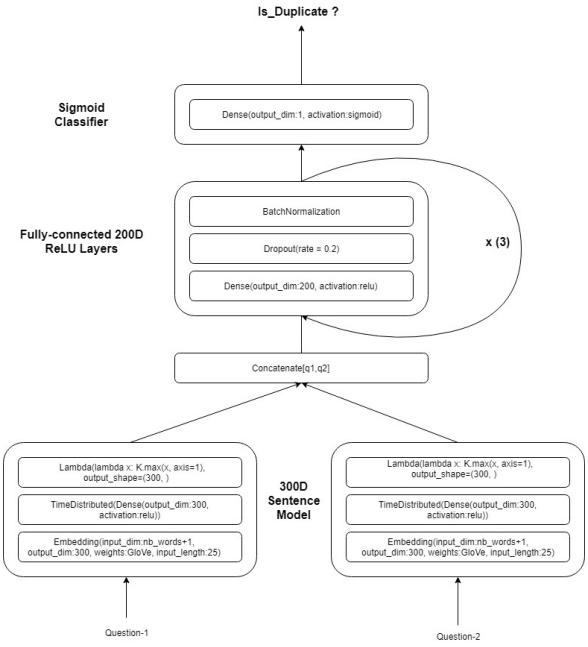

Figure 1: TimeDistributed Layer with Concatenation

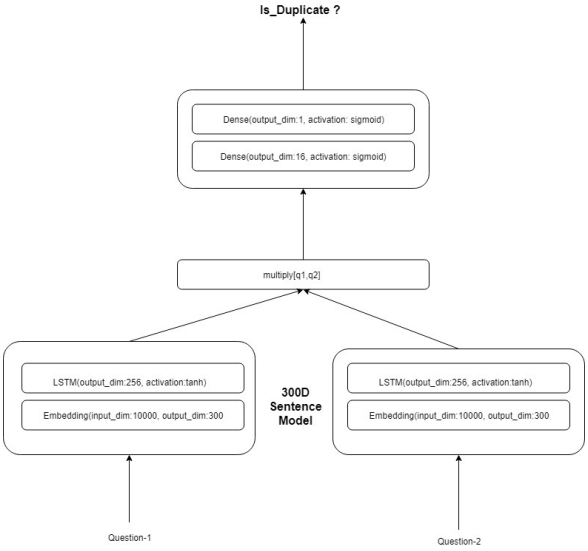

Figure 2: LSTM with Multiplication

4.4 Bidirectional LSTM with Attention

In these system decoder is suppose to generate a translation based on last hidden layer.But for example, in the many language the first word of translation is highly correlated with first word of source language.
To handle this we need to produce output based on many previous hidden layers. Bidirectional LSTM is able to deal with this problem to a great extent but long-range dependencies are still problematic.

Attention Network :
A bidirectional LSTM RNN is basically not only running text from left to right but also in the opposite direction from right to left. Outputs get combined before being passed on to the next layer with 'sum' merger.
Using attention model, we can shorten this complexity by just paying attention on what is more relevant. Model will itself learn which part to give attention based on the input sentence and the output produced so far.

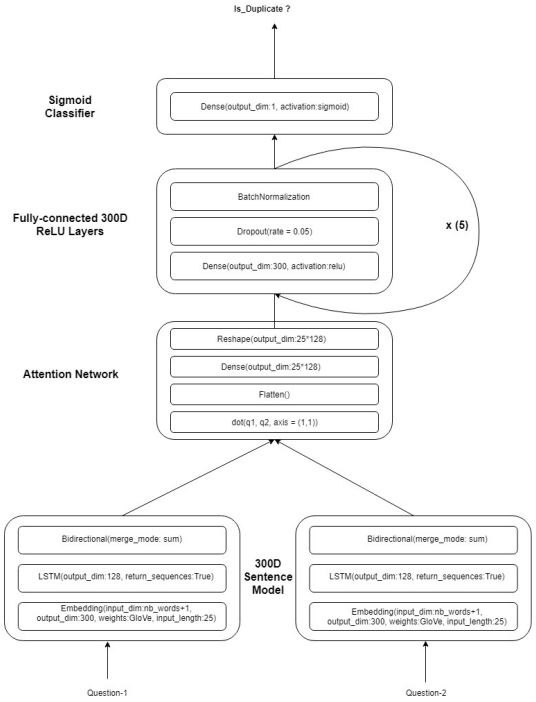

Figure 3: Bidirectional LSTM with attention

5 Experiments and Results

We used 10% data for testing, and out of the remaining percentage, 30% was kept aside for validation and 70% was used for the actual training.

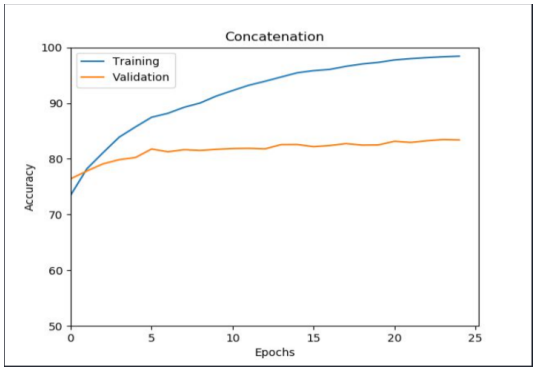

Figure 4: TimeDistributed Layer with Concatenation (Maximum valid. acc: 83.46 at epoch 24)

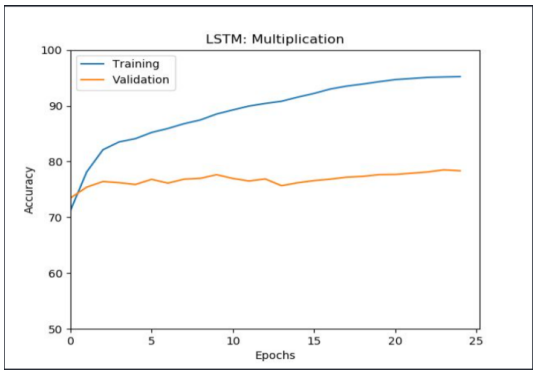

Figure 5: LSTM with Multiplication (Maximum valid. acc: 78.51 at epoch 24)

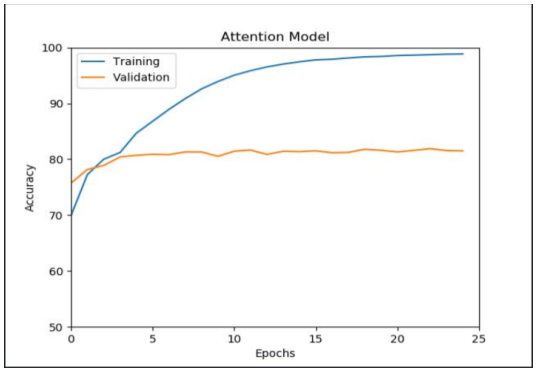

Figure 6: Bidirectional LSTM with attention (Maximum valid. acc: 81.89 at epoch 23)

Model	**Test Accuracy**
Bidirectional LSTM with Attention	81.78%
ATimeDistributed Layer with Concatenation	83.88%
LSTM with Multiplication	79.45%

6 Conclusion

We used three different approaches and varied their architectures until we got the best results. The results show the significance of GloVe embeddings over traditional count-based embeddings. Concatenation model outperformed the LSTM with attention and both of the approaches were better than the model using attention.

7 Future Directions

We can further extend this project for removal of duplicate answers for the same question. Supposedly, if two answers to a particular question are semantically similar above a threshold, then we could remove the smaller one.
Challenges: The main difference between this problem and the problem which we tackled in this paper is of the size of the text to be compared. Size of the answers can comparatively be much larger than the questions, so we'd have to handle much larger sized vector for the comparison.

Contribution

For instructor's use only :-
Every member contributes approximately equal.

References

[1] Jonas Mueller and Aditya Thyagarajan. Siamese Recurrent Architectures for Learning Sentence Similarity. In *AAAI*, 2016

[2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation," in *EMNLP* 2014

[3] E Dadashov, S Sakshuwong, K Yu. Quora Question Duplication

[4]Shuohang Wang and Jing Jiang. "Learning Natural Language Inference with LSTM" in *Proceedings of NAACL*, 2016

[5]Eren Golge. Duplicate Question Detection with Deep Learning on Quora Dataset