Proposal for the Quora Question Pair project

Han Weng[*], Hao Dong[†]

May 2019

1 Problem Description

This project is aimed at detecting duplicated questions on Quora. For a given input question pair, we would like to know if these two questions are asking the same an if they can be combined to one. For example, these two questions mean the same:

- What are the daily life examples of shear stress?
- What is shear stress? Also give any real life example of it.

However, these two, although look similar, are not the same:

- What should be a daily routine for anyone?
- What is your daily routine?

2 Data

The data we use will be from a Kaggle competition: https://www.kaggle.com/c/quora-question-pairs/data. The training set has 404k entries with labels, each containing two questions marked by ids. The test set (for competition) has 2.35 million entries, each having two questions.

3 Methodology

1. We plan to take several different features implied in the text into consideration, including embedding features, classical text mining features and structural features. For example:
   Embedding features: word embedding, sentence embedding.
   Classical text mining features: length similarity, character n-grams similarity(TF-IDF reweighted), percentage of common tokens when the end and start of both question are the same.

2. Several models, including Siamese LSTM and Attention Neural Networks may be involved in solving this problem. Until now we don't have a clear plan on how to apply these to the problem but as in Related work 1 and 2, these tools should be much helpful. We still need more study on this and will elaborate it in the future work.

[*] 20737611

[†] 20757585

3. Some work on balancing with the difference of target distribution between train and test maybe needed. Initially, we guess there will be biases on perimeters about the token counts of two questions in terms of size, relationship and extreme value.

4 Related Work

We will most likely focus on Siamese LSTM and Attention Mechanism. For the neural networks we may use, Siamese LSTM has pre-trained models available from Stanford (GloVe, https://nlp.stanford.edu/projects/
Two papers [1] and [2] have introduced the integration of Attention Neural Networks and such problem well, and some other participants in the competition achieved accuracy of 0.3 and 0.27 (cross-validation) respectively using this. However, it is mentioned that applying ANN to this could be very slow [3], so we may need to figure out some tricks to balance the performance and accuracy.

5 Evaluation plan

The test does not come with a answer sheet for the test data, therefore we will have to split the training set into two parts to perform a evaluation locally. However, the competition does accept late submission, which means we can still take advantage of the competition system to test our model. Since the label for the data is Boolean (0 for non-duplication and 1 for duplication) and we will report the probability of duplication (e.g. 0.8), we will use the simple log loss function to characterize the accuracy of model. As a result, we will evaluate our model in both ways (training set and competition mark) and report the loss values in the report.

References

[1] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference, 2016.

[2] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference, 2016.

[3] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension, 2016.