MSCI641 Project Milestone

Han Weng (20737611), Hao Dong (20757585)

June 2019

1 Introduction

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 totally identical so they can be combined into one.

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 over 400 thousand entries with labels, each containing two questions marked by ids. The test set (for competition) has 2.35 million entries, each having two questions. For this milestone, we only use the training data and split it into train, validation and test sets as we need to evaluate the baseline with know results.

3 Baseline

We build our preliminary model on some popular works and kernels on Kaggle. The idea of distinguishing duplicated questions is having two input layers accepting tokenized input data of both questions, then process them with an embedding layer with weights adopted from pre-trained GloVe word vectors. The vectors then go through several hidden layers including an LSTM layer, some dropout layers and some gaussian noise before they reach an output layer where the model gives a prediction of "duplicated" or not. These are the preprocessing we have done:

- Eliminated some too short questions
- Replaced some abbreviations with their expanded words
- Tokenize and vectorize words

And the baseline model looks like this:

```
__________________________________________________________________________
Layer (type)                 Output Shape         Param #     Connected to
==========================================================================
input_2 (InputLayer)         (None, 15)           0
__________________________________________________________________________
input_3 (InputLayer)         (None, 15)           0
__________________________________________________________________________
embedding_layer (Embedding)  (None, 15, 300)      26109600    input_2[0][0]
                                                             input_3[0][0]
__________________________________________________________________________
lstm_1 (LSTM)                (None, 75)           112800      embedding_layer[0][0]
```

```
                                                                  embedding_layer[1][0]
__________________________________________________________________________________
q1_q2_concat (Concatenate)      (None, 150)          0           lstm_1[0][0]
                                                                 lstm_1[1][0]
__________________________________________________________________________________
output_layer (Dense)            (None, 1)            151         q1_q2_concat[0][0]
==================================================================================
Total params: 26,222,551
Trainable params: 112,951
Non-trainable params: 26,109,600
__________________________________________________________________________________
```

4 Evaluation Methodology

As this is a binary classification problem, the evaluation method is quite straight forward. We simply use the loss and accuracy of the model calculated on known results. At this stage, as we are not ready to submit a full solution to Kaggle, we only stick to the given training set and perform all training, validating and test on that.

5 Results

Based on a very rudimentary baseline, we get about 73% accuracy. This is high because most of the question pairs are not duplicated. We will promote the accuracy by performing refined pre-processing on the input data, better tuning and stacking layers, or introducing some more architectures to the model. For the preprocessing part, we may truncate sentences more reasonably, eliminate rare words to avoid overfitting, and exclude very short/meaningless questions (e.g. "What?", "How long?") to achieve better generalization. For the model improvement, now we see that we can use k-folds to make k predictions, then ensemble the answers and postprocess to form answer, as declared effective by some other competitors.