Heinz 95-845: Compare Different Models Working on Duplicated Question Recognition Applied Analytics: the Machine Learning Pipeline

Ao Luo aol/aol@andrew.cmu.edu
Department of Engineering
Porter Hall A107
Pittsburgh, PA, United States

Dingze Wang dingzew/dingzew@andrew.cmu.edu
Engineering Technology Innovation Management
Heinz A206
Pittsburgh, PA, United States

Abstract

Quora recently released a dataset for duplicated questions detection. In this project, we developed two models in different way to solve this problem: a traditional gradient boosting tree model and a neural network model. Both of them could bring reliable predictions and neural network is apparently superior. To further compare the generality of the two models, we generated another dataset to verified the outcomes without changing the training schema. The results shows that our models have good generality and still neural network model is better.

1. Introduction

As the Internet grow up at a speed everyone cannot imagine, people can get information from different sources, including Search Engines, Forum and etc. Among all the question answering website, Quora is famous because of several unique features or function: Real Name Policy, Answer Recommendations, Content Moderation, Top Writers Program, Quora World Meetup[1]. We can see most of them are base on Machine Learning algorithm. There are lots of different tasks for Quora to deal with. For example, Recommend potential interesting questions based on each users cookie, Merge duplicate questions in order to let user find the good answers quickly. Currently Quora is using Random Forest Model to deal with duplicate question problem. In this paper, we explore two more different models which are Neural Network Model and Boosting Tree Model. For the remaining parts, in section 2, we provide the background of the underlying concept for our models. In section 3 we discuss two datasets for test. In section 4, we provide some of the metrics we use for this project. In section 5, we give the implementation details and the pipeline of the models. In section 6, there are results and the discussion. In section 7, we provide our conclusions and potential points that can be optimized.

1. TechCrunch.com, https://techcrunch.com/2011/01/05/quora-surge/

2. Background

For most of the NLP problem. The result can base on many features. The feature sometimes can be learnt by Machine learning models such as Neural Networks. Among neural networks, there are lots of different kinds of mutations such as CNN, RNN and etc. The network structure is subject to change, such as hidden layer number, normalization function and etc. Below is a simple neural network structure with one hidden layers. By using this structure, the network will interpret the relationship between information and text itself. For NLP problems, because the word is strong related to the context close to itself, LSTM strategy is used to forget the faraway context and emphasize the context close to the word. The pattern learn by the neural network is vague for human beings and its all encoded in the trained parameters.

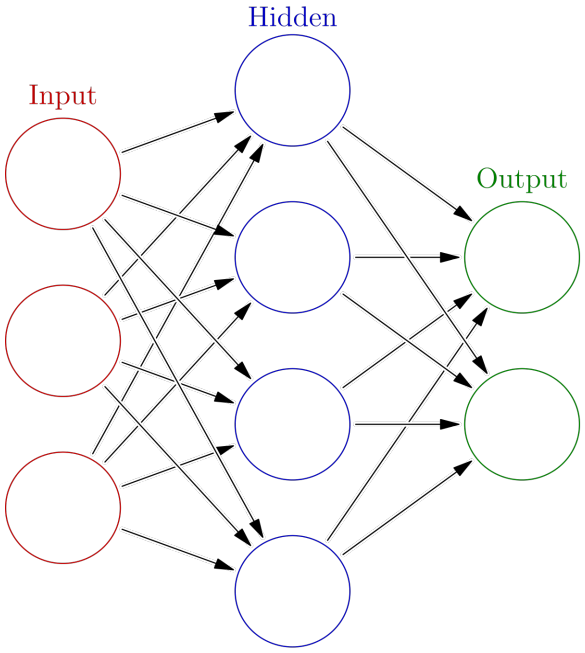

Figure 1: Simple Multilayer Neural Network

We can also use the existing knowledge in Natural Language Processing to select features that high relevant to our problem. The key part of whether two questions are similar is semantic (Ref, Natural Language Processing Version 3, Stanford University). Semantic Vectors such as Word2Vec are important for us to compare two texts. Figure 2 below shows how to get Word2Vec. There are also lots of important features that have already been proved that are very useful to our current problem. That reminds us to use mechanism like vote.

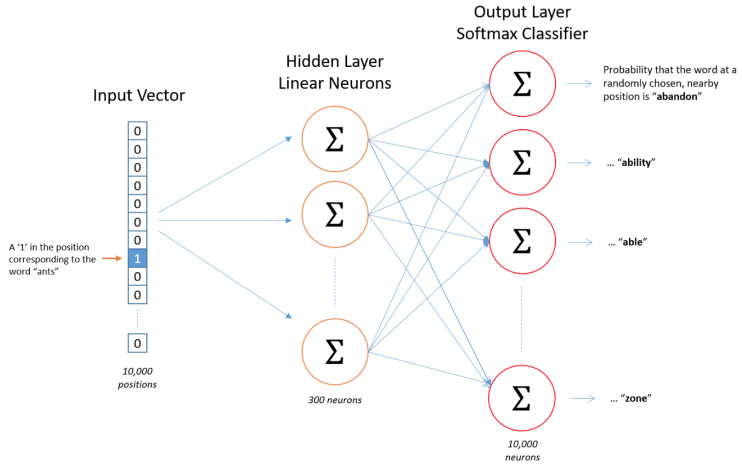

Figure 2: Word2Vec

Using Boosting Tree, multiple features can be synthesized. Compared to Neural Network, It is a more controllable way because we can always add and delete features. Even if different features have different effect on the final result, after combining them together, they can achieve a better performance than any single of them. Figure 3 shows the mechanism of Boosting Tree.

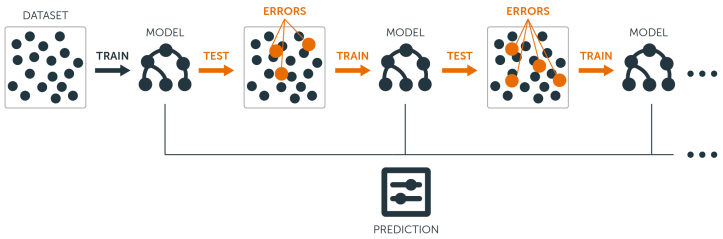

Figure 3: Boosting Tree

3. Dataset description

In this project, we used Quora dataset and wiki dataset. The Quora dataset is released by Quora in 2017 [2] to accelerate the research in identifying duplicate questions. In the Quora dataset, questions exist in pairs and a label is given to justify whether the two questions are duplicate or not. We build our models based on the Quora dataset.

The wiki dataset is a contrast dataset that we generated from WikiAnswers corpus[3]. The WikiAnswers corpus contains clusters of questions tagged by WikiAnswers users as paraphrases. Each cluster optionally contains an answer provided by WikiAnswers users.

2. Quora, https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs
3. QokiAnswers Corpus, https://github.com/wasiahmad/paraphraseidentification/tree/master/dataset

We use this corpus to generate similar and dissimilar question paris. In order to test the models we built on the Quora Question Pairs dataset, we generate the wiki dataset in a way such that the two datasets have different label distributions. Table 1 is the comparison between two datasets. Notice that in Quora question paris dataset, the pairs with high word sharing rate might still be very different, while in our generated Wiki question pairs dataset, the paris with low word sharing rate might, however, be considered as similar.

	Quora dataset	Wiki dataset
Total samples	404290	799137
Duplicate question pair rate	0.369	0.099
Duplicate question rate	0.335	0.200
Ratio of unique words	2.17e-5	8.27e-6
Mean of question lengths	11.06	10.37
Std of question lengths	5.89	25.4

Table 1: Comparison between Quora and Wiki dataset.

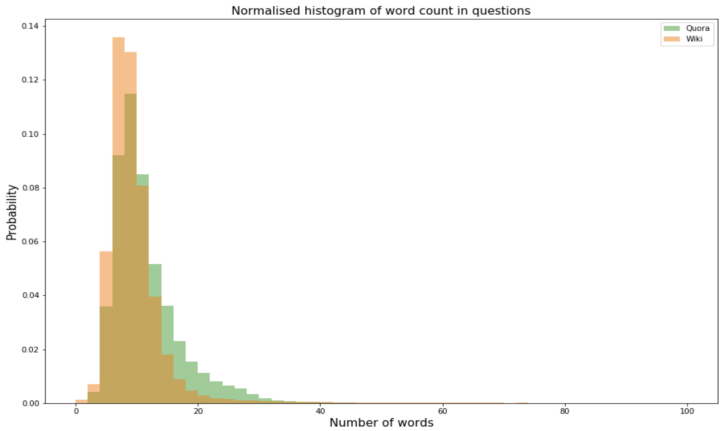

Figure 4: Normalized Histogram of Word Count in Questions

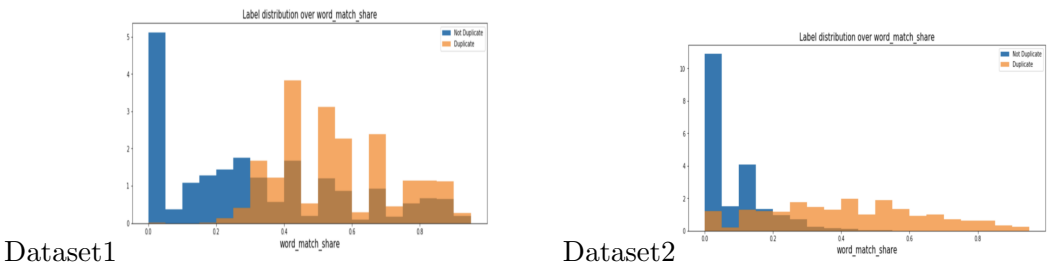

Figure 5: Label Distribution Over Word Match Share

4. Metrics

Whereas the AUC is computed with regards to binary classification with a varying decision threshold, logloss actually takes "certainty" of classification into account. For this project, we compare the AUC score during hyperparameter tuning, but in the final report, we only use logloss since AUC has minor difference.

Log loss. Log-loss takes into account the uncertainty of the prediction based on how much it varies from the actual label. In this binary classification task,

$$Log - loss = -(y \log p + (1 - y) \log(1 - p))$$

ROC-AUC. The probabilistic interpretation of ROC-AUC score is that if you randomly choose a positive case and a negative case, the probability that the positive case outranks the negative case according to the classifier is given by the AUC. Mathematically, it is calculated by area under curve of sensitivity (TPR) vs. FPR(1-specificity).

5. Model Implementation

5.1 Neural Model

In this neural network model, we manage to reduce the feature engineering as much as possible and push the network to figure out the interrelations between question pair by itself. Preprocessing is designed to unify the different writing styles of same content, and several tricks are designed to eliminate the order difference and improve generalization. Figure 2 is the structure of the our model, inspired from siamese network. The model take question sequence pairs and graph features. In the process of extracting graph features, we treat each question as an node, and use the connection between question pair to summarize the inter-reactions between all the questions.

Preprocessed question paris go through same embedding and the following same LSTM layer, the last output sequence is extracted as the representation of each question, and then the addition and difference of the representation is treated as the rnn features. On the other end, the graph features go through a fully connected layer, extracted as dense representation. The concatenation of rnn features and dense graph features are considered as the whole representation of question pair. The final prediction is made on the fully connected results of that.

An important aspect of the model is that it totally eliminates the order difference of the two questions by processing two questions using same LSTM layer and by the addition and difference square tricks, instead of just concatenating the two question pairs. This is good for the generalization of the model. In addition, batch normalization is introduced to speed up learning and provide some regularization. Dropout is applied multiple times during the network to prevent units from over-correlating. Both of them bring about additional generalization.

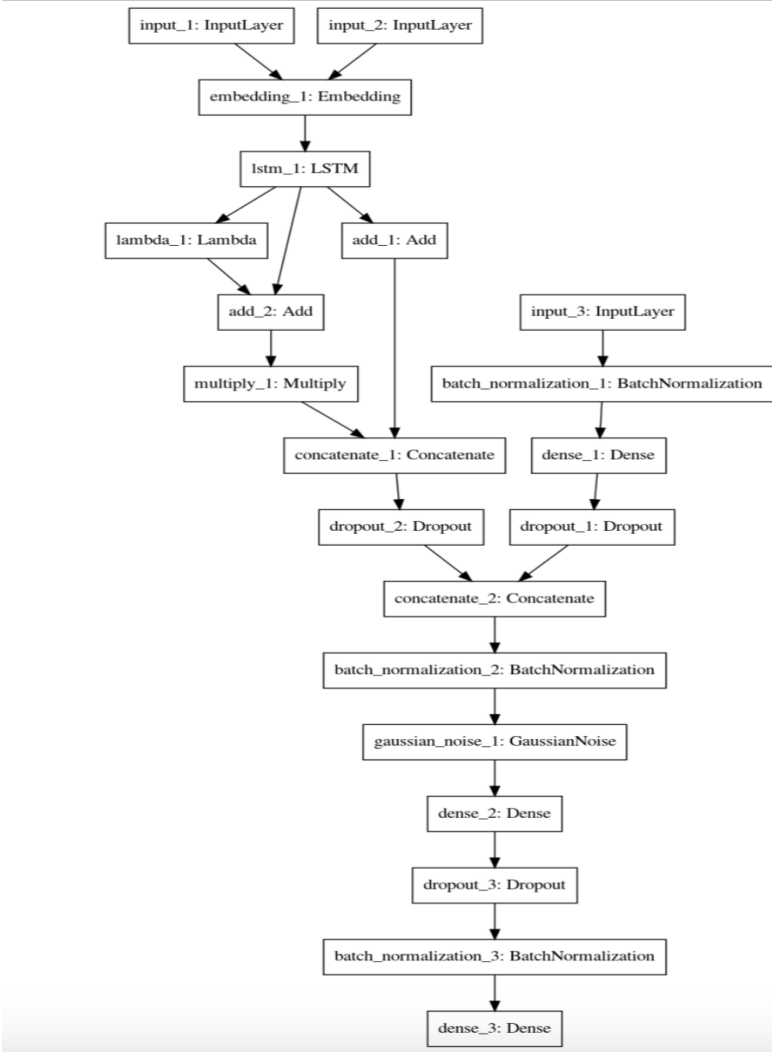

Figure 6: Neural Network Structure

5.2 Gradient Boosting Tree Model

In our gradient boosting tree model, we solved the problem in a different way. Instead of relying on our model to do the most feature engineerings work by itself, we managed to fully utilize the traditional nlp tools, and extract as many features as we can by ourselves from intuition and experiments. Here, we used lightgbm package to build our gradient boosting trees model.

Light GBM is a gradient boosting framework that uses tree based learning algorithm. Light GBM grows tree vertically, aka leaf-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm. [4]

4. Lightgbm Website, https://lightgbm.readthedocs.io/en/latest/

In order to use GBM model, we select several features below.

1) Length and length difference The length of questions on char and words levels, also the absolute length difference of two questions. A question pair with large length difference gap might more likely be not duplicate.

2) Common words, fuzz features and Levenshtein Distance The common words measure the shared words between question pairs. The fuzz features and Levenshtein distance provide different evaluation on string matching that reflect the difference or similarity between two strings concerned about per character difference. For example, Levenshtein distance is shortest times of modification to make one question to be another question. If the edit distance is 0, two questions are exactly the same question

3) Cosine Similarity: This bag of word approach is the most reasonable feature. However, since each question only have a few words, this approach sometime will not work well when synonym appears very frequently. The vector of the sentence is easy to construct so we will construct vector by ourselves.

4) Word2Vec The word2Vec train the vector for each word. By treating the context around the word as positive example and randomly sampling other words in the lexicon to train a logistic regression classifier. This method works very well when the dataset is large. However, we should notice that the window size should be modified properly since the questions are usually very short. In this project, we get pre-trained model from library. In this project, we use doc2vec in genism models

5) More distance in Scipy library Cosine similarity is a common feature to compare similarity. For the question similarity problem, we dont know if any other distance metric works better. So, we add more distance features such as Cityblocks, Jacquard, Canberra, Euclidean and etc.

5.3 Out of Folder Predictions

To make the model more stable and thus let the results more convincing, we split the training set into 10 subsets in stratified way. In each time, we select one fold out as validation set, and use the remaining 9 folds as training set to train a boosting model. Finally, we could get 10 best model for each training subsets. We use the 10 models to predict on the testing set, the final prediction is based on the average of these 10 predictions.

6. Results

6.1 Result on Dataset 1

	NN Solution	LGB Solution
Loss	0.1495	0.24
Preprocessing Time	2.5min	11min
Training time (10 oof)	3h(on k80 GPU, batchsize=256)	5h

Table 2: Performance comparison 2 models on Quora dataset

Analysis: 1. Both the neural network model and gradient boosting tree models provide reliable predictions and solve the problem. While, neural network model apparently per-

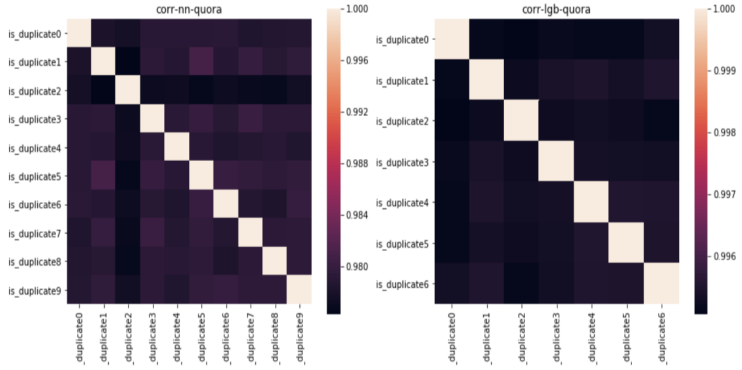

Figure 7: Correlation Map of 10 Folds Predictions

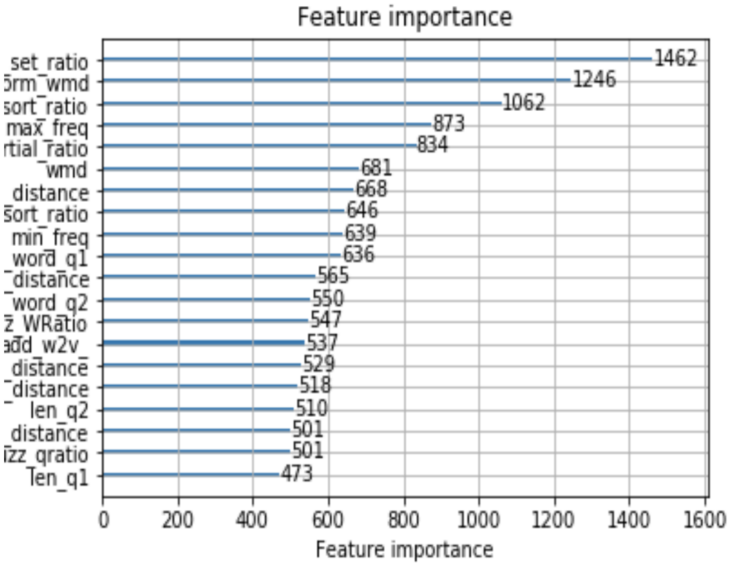

Figure 8: Feature Importance for LGBM model on wiki dataset

forms better than gradient boosting tree model. 2. Neural network save the time for feature engineering. With the acceleration of GPU, neural network model could probably reduce the project time, and still provide better results. 3. Gradient boosting trees could provide the feature importance plots that make the model much interpretable. 4. The correlation map shows that both neural network models and gradient boosting tree models that are trained on different subset of the data have highly correlated results. Neural network models has higher fluctuations on the predictions of 10 each fold, 10 folds out-of-fold prediction help reduce variance, and reach better generality.

6.2 Result on Dataset 2

Analysis: The models that fine-tuned on the Quora dataset could still yield reliable results without changing the training schema. The neural network model still have better

	NN Solution	LGB Solution
Loss	0.01425	0.02580
Preprocessing Time	5min	20min
Training time (10 oof)	4h(on k80 GPU, batchsize=256)	5h

Table 3: Performance comparison 2 models on Quora dataset

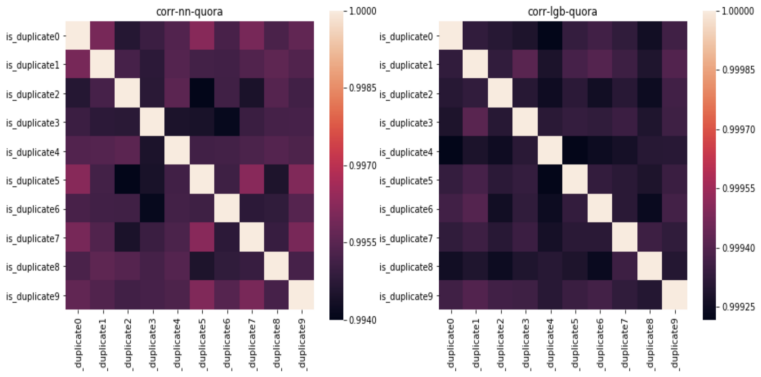

Figure 9: Correlation Map of 10 Folds Predictions

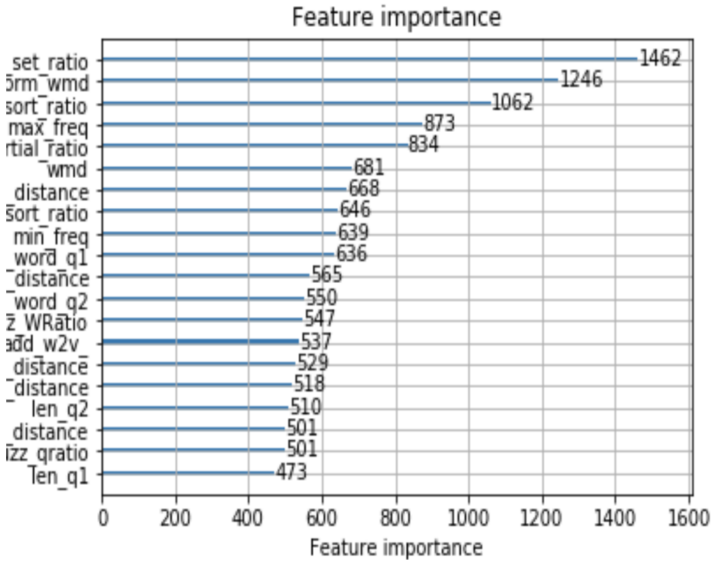

Figure 10: Feature Importance for LGBM model on wiki dataset

performance, less training time with GPU accelerating and higher fluctuations compared to gradient boosting tree models. Overall, these two models have good generality and performance, and neural network is superior than gradient boosting tree models in such duplicated question pair detection task.

7. Conclusion

In this project, we implemented two different models to solve duplicate question detection problem. One model is using Neural Network Model while another is using Gradient Boosting Tree with carefully selected features. The Neural Network model achieves a better performance both in training time and loss. This probably imply Neural Network is more suitable for this kind of text comparison problem. However, the reason may also because the features we select are not good enough. If using more advanced Natural Language Processing technique, Boosting Tree Model may be able to achieve a better performance. We also use another dataset to explore if theres any difference of two models working on different dataset. Generally, our models work reasonably good on both of the datasets, showing good generality. The models can still be modified in order to achieve better performance by modifying the structure of NN or add more relevant features to Boosting Tree Model. The ideas of this project is not limit in this topic but can also be used in some other circumstance such as plagiarism detection.

References

Dan Jurafsky and James H. Martin. Natural Language Processing. `https://web.stanford.edu/~jurafsky/slp3/`.

Wikipedia. Figure 1,2,3. `https://en.wikipedia.org/wiki/Artificial_neural_network`.