Quora Duplicate Questions

Palash Goyal palashgoyal1@gmail.com
https://github.com/palashgoyal1

Data Available : Pair of questions with flag of duplication or not

Feature Engineering : Both the set of questions were processed with Punctuation removal, conversion to lower, removal of stop words, stemming and stripping the Space, and the corpus was created.

1. Set of **Basic features** were created for both the given questions :
    a. Total word count in the original set of questions
    b. Total word count in the reformed set of questions, after above processing
    c. Total unique word count in the original set of questions
    d. Total unique word count in the reformed set of questions
    e. Total character count in the original set of questions
    f. Total character count in the reformed set of questions
    g. Total count of the unique common words in both the questions
    h. Percent of the unique common words in both the questions

The above set of features were created in R (File-1) and were put in the XGBoost model and a thorough grid search was done with optimizing on the AUC metric (File-2)and final predictions were made on the test data(File-3), after computing the same set of features on test data.

2. Set of **Fuzzywuzzy features** :
    a. Partial Ratio
    b. Partial Token Set Ratio
    c. Partial Token Sort Ratio
    d. QRatio
    e. WRatio
    f. Token Set Ratio
    g. Token Sort Ratio

The above set of features were created in Python on train dataset (File-4)and on test dataset (File-6), and were put in the XGBoost model, along with the Set-1 Basic features set, and a thorough grid search was done with optimizing on the :

i) AUC metric (File-5) : final predictions made on test data in (File-7)
ii) LogLoss metric (File-8) : final predictions made on test data in (File-9)

3. Set of **word2vec features** : (on sentence vectors formed by using tokens, created after again processed/cleaned pair of questions)
    a. General features :
        i. Frequency of questions in ‘question1’ column within the column
        ii. Frequency of questions in ‘question2’ column within the column
        iii. Frequency of questions in ‘question1’ column across both the columns
        iv. Frequency of questions in ‘question2’ column across both the columns
    b. Word2vec features :
        i. Cosine Similarity
        ii. Cityblock Similarity
        iii. Jaccard Similarity
        iv. Canberra Similarity

v. Euclidean Similarity
vi. Minkowski Similarity
vii. Braycurtis Similarity
viii. Skewness on 'question1' column's sentence vectors
ix. Skewness on 'question2' column's sentence vectors
x. Kurtosis on 'question1' column's sentence vectors
xi. Kurtosis on 'question2' column's sentence vectors

The above set of features were created in Python (File-10) for both train and test datasets, and were put in the XGBoost model and a thorough grid search was done with optimizing on the LogLoss metric (File-11)and final predictions were made on the test data(File-12).

Model Selection :

1. **XGBoost** model has been selected for the current purpose. Further detailed breakup of steps :

a) Multiple features were created in an iterative manner, and the successive model results on test dataset were saved to be used later for the ensemble purpose.

b) Three different ensemble methods (Average, Geometric mean, Rank Average) of multiple combinations (around 8) of the submission files (around 16) were used and submitted as solution.

c) Dataset was split between train (80%) and validation (20%), and the trained model was validated. The final hyper-parameters from the model trained on train part (80%) were used to train new model on the full dataset. This resulted in at least 3 models for each of the metric selected above (roc_auc/log_loss) :

i) Model with default hyper-parameters on the train part (80%)
ii) Model with final hyper-parameters after grid search on the train part (80%)
iii) Model with final hyper-parameters after grid search on the full dataset (100%)
iv) **(Only for Till now Final model)** Further exhaustive grid search was done, and model was trained on the full dataset (100%)

d) Ensemble model with geometric mean of results from the XGBoost with log_loss on all features with highly exhaustive grid search was selected as final submission.

2. **Word2vec** model was trained on the words generated from both the pairs of columns, and was further used to represent the sentences as vectors, which were used to create Set-3(b) features (File-10).

Hyper-parameter Tuning : A default set of parameters has been taken into account in the first iteration, and the hyper parameters have been further tuned by following a Grid Search based approach. The parameters tuning have been done on the basis of the priority and significance of each of the parameters in the model. The grid values chosen by the model have been done with the metric of roc-auc or log-loss in different model iterations with different set of features. The ranges for the hyper-parameters were selected on the basis of previous experience and general recommendations.

Additional Steps for better accuracy :

1. Additional features : TF-IDF set of features
2. Pre-trained Google word2vec model (https://github.com/mmihaltz/word2vec-GoogleNews-vectors)
3. Other models : Deep Siamese LSTM networks

* Both of the above steps are in-progress and will be updated on github profile
* I will be releasing a detailed blog on http://madoverdata.com/ in June, and would be happy to receive comments/feedback.