|  |  |  |  |
| --- | --- | --- | --- |
| Capstone Project |  | | Ang Zhen Xuan |
| Machine Learning Engineer Nanodegree | |  | 26th December 2017 |

Quora Question Pairs

**Project Overview**

**Domain Background**

Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly worded questions. Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question. Quora values canonical questions because they provide a better experience to active seekers and writers, and offer more value to both groups in the long term.

**Datasets and Inputs**

Dataset obtained from: <https://www.kaggle.com/c/quora-question-pairs/data>

Input variables of the train.csv include:

Independent variables

* id - the id of a training set question pair
* qid1, qid2 - unique ids of each question (only available in train.csv): question1, question2

Dependent variables

* is\_duplicate - the target variable, set to 1 if question1 and question2 have essentially the same meaning, and 0 otherwise.

The question1 and question2 text will be cleansed on (through stemming and stop words, etc) prior to the training of the machine learning models. Said model is trained on the training dataset to prevent implicit bias infused into the trained model by ‘peeking’ into the test model[[1]](#footnote-1). while testing is done on the testing dataset. The accuracy obtained from comparing the model procured answers and the labels on the testing set can be used as a decent gauge for the performance of the machine learning techniques adopted in training.

**Presence of noise**: Human labelling is also a 'noisy' and inherently subjective process, and reasonable people will disagree. As a result, the ground truth labels on this dataset should be taken to be 'informed' but not 100% accurate, and may include incorrect labelling.

**Imbalanced classes**: The imbalance brought out by labelled classes needs to be addressed as 149,263 examples are labelled with 1, and the majority examples are labelled with 0. This is an issue to be addressed either by oversampling/undersampling methods[[2]](#footnote-2).

I originally planned to use the following pre-trained word embeddings (resources shared on Kaggle[[3]](#footnote-3)):

1. Google's word2vec embedding from [this link](https://code.google.com/archive/p/word2vec/)
2. Glove word vectors from [this link](https://nlp.stanford.edu/projects/glove/)
3. Facebook's fastText embeddings from [this link](https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md)

Only the word2vec embedding from Google was utilized since I was computing with limited resources (on dual-core CPUs).

**Problem Statement**

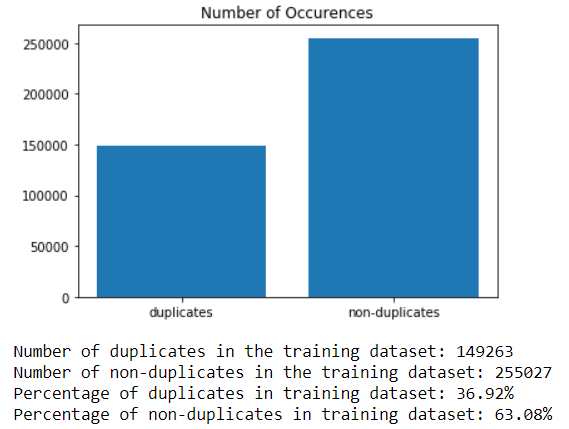
Currently, Quora uses a Random Forest model to identify duplicate questions. By tackling this natural language processing problem by applying advanced techniques to classify whether question pairs are duplicates, will make finding high quality answers to questions easier. This would result in an improved experience for Quora writers, seekers, and readers. I used the sklearn library’s RandomForestsClassifier to replicate a simple benchmark prediction model to be compared against.

The accuracy of the trained model must be able to quantify well enough (in the accuracy sense, to justify the use of the benchmarked Random Forest classifier or any subsequent prediction model to be used after training) as well as be able to generalize well enough to new question pairs.

I adopted the Keras wrapper with Tensorflow as its backend to train the MaLSTM[[4]](#footnote-4) model. The NLTK library was also used on the word vectors of the question pairs to help tokenize word into vectors, facilitating model training.

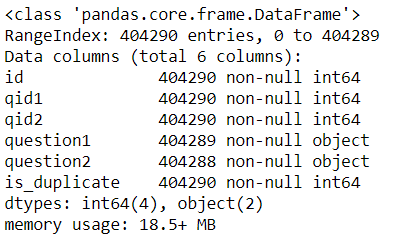
**Data Exploration & Exploratory Visualization**

Referencing Figure 2, the percentage of duplicates to non-duplicates is as shown:



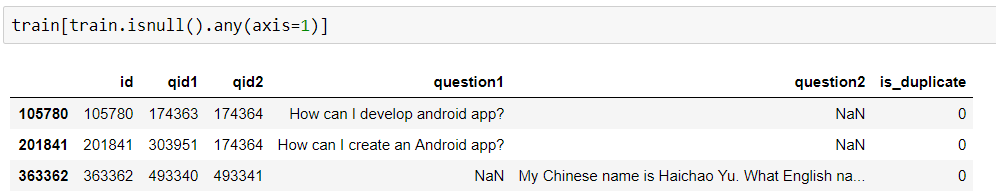
*Figure 1: Amount and Percentage of duplicates to non-duplicates in the training dataset*

This puts me in good stead with regards to the distribution of the training dataset. Next, I looked into the null values contained within the training distribution. This was easily achieved using **info()** method on my **train** DataFrame.



*Figure 3: Checking the null values that may need to be handled*

As shown in the figure above, both the question1 and question2 rows show a count of non-null objects that is less than the total number of question pairs in the training dataset (404289 and 404288 respectively). This proves that there are two null values in the question2 column while there exists one null value in the column of question1.

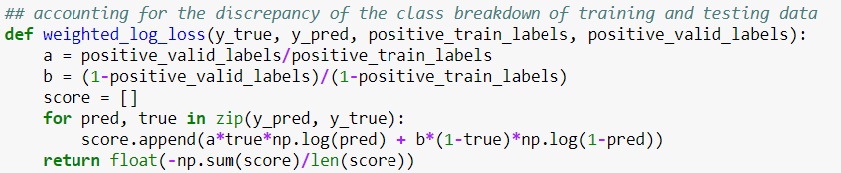


*Figure 2: Making sure the non-null rows are actually NaN values*

I decided to remove it from the training dataset as the null values would not contribute much to the prediction model in my opinion.

**Metrics**

It was written that Quora uses a model based on Random Forests to detect similarities between questions. I simulated a simple Random Forest model, and used it as a benchmark against the MaLSTM. We seek to minimise the error based on the weighted log-loss function that we have defined to be the main metric for comparison.



*Figure 3: Weighted log-loss function used in training the models and evaluation*

The weighted log-loss function was derived from the hold-out test set that I plan to use to evaluate the effectiveness of the prediction models. The test dataset has a label split of 83% non-duplicates and 17% of duplicates compared to the 63-37 split in the training dataset. Accounting for the difference in distributions between both datasets, the weighted log-loss function was derived based on the Bayes’ Theorem[[5]](#footnote-5).

In addition, another evaluation metric to be compared against models would be the classification accuracy of the models (on the testing data set).

Classification Accuracy = Accurately Classified/Number of Test Data Points

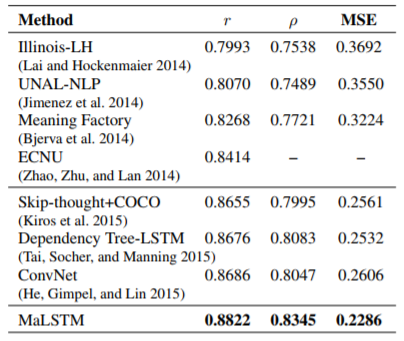
Lastly, I would also generate the confusion matrix to see if we can derive any further insights.

**Algorithms & Techniques**

Neural networks are adaptable, and have been gaining traction in the field for handling word vectors yielding comparable results to classical measures like Cascading Features and Shallow Join Training for Natural Language Processing problems[[6]](#footnote-6).

My original intention of training a model based on Convolutional Neural Networks (CNN) over RNNs was due to the fact that RNNs are slower and fickler to train, and sequencing may not be that important[[7]](#footnote-7) as ‘feelings’ detection (such as happiness or sadness) in corpora may be more essential.

However, after further research, I realised that the Siamese Manhatten LSTM (MaLSTM)[[8]](#footnote-8) provides a rather straightforward approach to the problem of sentence similarity. With reference to Figure 1, the MaLSTM has outperformed its counterparts (similar semantic handling models) by obtaining the lowest mean squared error on the SICK (Sentences Involving Compositional Knowledge) dataset – usually used to gauge performances of models with regards to accounting for syntactic and semantic issues[[9]](#footnote-9). It has also proven to perform well especially in tasks such as semantic similarity, which is essential to Natural Language Processing problems in recent times.



*Figure 4: Referencing the MSE column, it has the lowest error rate compared to the other models based on the SICK semantic similarity task.*

**Data Preprocessing**

* Classical features & word embeddings processing

**Implementation**

* Code chunk to show evaluation of both random forests & MaLSTM

**Benchmark**

* Build Random Forests Classifier on classical features built through the sentences

**Refinement**

* Built a Siamese MaLSTM prediction model trained on pre-trained word embeddings from Google

**Model Evaluation & Validation**

* Benchmarks from Kaggle ranking and error loss rate. (validation accuracy?)
* Briefly talk about the robustness provided by GridSearch and cross validation

**Reflections & Improvements**

In conclusion, after rigorous testing and tuning of the prediction model’s parameters, I believe there are a few points we can conclude for the end model:

1. Firstly, transfer learning could be applied to our initial weights – pre-train MaLSTM on separate sentence-pair data provided in the SemEval 2013 Semantic Textual Similarity paper[[10]](#footnote-10)instead of randomly drawing weights from a Gaussian distribution. This has proven to be a superior starting point compared to random initliazation.
2. Generate possible new data, be it manually or through generative deep learning systems such as the GAN (Generative Adversarial Networks) to be used in the training processes. More data usually provides more robust and reliable prediction models.
3. Possibly replace words with their synonyms to obtain more training data.
4. The word embeddings from Google that I have utilized may not have been the best contextual corpus to derive weights from as Google’s corpus was based on news articles and reports. This corpus may not be providing totally relevant context to my corpus of sentences from a diversity of categories on Quora.
5. Lastly, the classical features could have been expanded on as the number of features were limited – might not have been a representative trained Random Forests classifier.

1. Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H. (2012). *Learning from data: a short course*. S.l.: AMLbook.com. Chapter 5 – Three Learning Principles. [↑](#footnote-ref-1)
2. Dealing with class imbalance while using CNNs. <http://www.academia.edu/8472416/Tackling_Class_Imbalance_with_Deep_Convolutional_Neural_Networks> [↑](#footnote-ref-2)
3. Word embeddings resources shared on Kaggle -

   <https://www.kaggle.com/c/quora-question-pairs/discussion/30286> [↑](#footnote-ref-3)
4. Original paper published for the MaLSTM model - <http://www.mit.edu/~jonasm/info/MuellerThyagarajan_AAAI16.pdf> [↑](#footnote-ref-4)
5. Accounting for differences in training and test dataset’s distributions - <https://swarbrickjones.wordpress.com/2017/03/28/cross-entropy-and-training-test-class-imbalance/> [↑](#footnote-ref-5)
6. Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing. *Proceedings of the 25th international conference on Machine learning - ICML 08*. doi:10.1145/1390156.1390177 [↑](#footnote-ref-6)
7. Reasons relating to picking CNN over RNN. <https://datascience.stackexchange.com/questions/11619/rnn-vs-cnnat-a-high-level> [↑](#footnote-ref-7)
8. How to predict Quora Question Pairs using Siamese Manhattan LSTM - <https://medium.com/mlreview/implementing-malstm-on-kaggles-quora-question-pairs-competition-8b31b0b16a07> [↑](#footnote-ref-8)
9. The SICK dataset - <http://clic.cimec.unitn.it/composes/sick.html> [↑](#footnote-ref-9)
10. SemEval 2013 Semantic Textual Similarity - <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=36B3188663E26B1D311592D8757A11B7?doi=10.1.1.310.7053&rep=rep1&type=pdf> [↑](#footnote-ref-10)