Machine Learning Engineer Nanodegree

Capstone Project

Elvyna Tunggawan
May 13, 2018

I. Definition

Project Overview

Emergence of web 2.0 has increased variety, veracity, velocity and volume of textual data. Demand of text analytics researches also increases, such as sentiment analysis, topic modelling, text summarization, and text duplication detector. One of the challenges on text analytics, especially on text duplication detector, is identifying semantically equivalent sentences. Different phrases could have similar meaning, while a word also could have different meanings. Moreover, there are thousands of language in the world with various grammars and vocabularies, which add the complexity.

As a question-and-answer site, Quora enables people to submit questions, give high quality answers, and learn from each other. Having more than 100 million users, submitted questions could have similar words, and multiple questions might have similar intent. Quora emphasizes on providing canonical questions in order to ease user on finding the best answers of a question and prevent users from answering similar questions. This project uses Quora question pair dataset which is provided on Kaggle[1].

Problem Statement

The main problem on this project is to identify duplicate questions, so that all questions provided on Quora are unique. This could be approached as a binary classification problem, in which features on each question pair could be extracted and represented in numerical features. Given the extracted features, the classification model will identify whether the question pair is a duplicate or not. The main challenge is identifying semantically similar questions, since it couldn't be identified merely from character-level identification such as edit distance. As a solution, besides extracting sentence characteristics from the question, word embedding will also be used.

Metrics

Since the dataset has imbalanced proportion on each class (there are more non-duplicate pairs), model performance will be mainly measured based on its F1 score. F1 score is a harmonic mean of the precision and recall, which is appropriate to use when the sample contains extreme values. I select F1 score instead of other F-measure since it is equally important to have good precision and recall. In addition, accuracy of the prediction will also be reported.

---

[1]Quora Question Pairs. https://www.kaggle.com/c/quora-question-pairs. Accessed on February 20, 2018.

II. Analysis

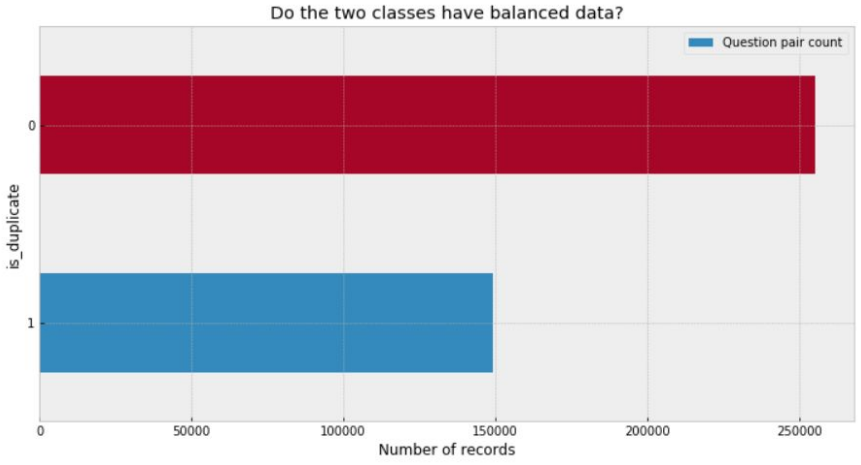
Data Exploration

The dataset contains 404,290 records, where 2 records of non-duplicate class contain empty question pairs. Each record consists of some fields; the sample records can be seen below.

Sample questions

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in share market in india?	What is the step by step guide to invest in share market?	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Diamond?	What would happen if the Indian government stole the Kohinoor (Koh-i-Noor) diamond back?	0
2	2	5	6	How can I increase the speed of my internet connection while using a VPN?	How can Internet speed be increased by hacking through DNS?	0
3	3	7	8	Why am I mentally very lonely? How can I solve it?	Find the remainder when [math]23^{24}[/math] is divided by 24,23?	0
4	4	9	10	Which one dissolve in water quikly sugar, salt, methane and carbon di oxide?	Which fish would survive in salt water?	0

The identifier fields (*id, qid1, qid2*) are not used as inputs to the classification model, since they don't resemble characteristics of the question pairs. The questions pairs aren't distributed equally on the two classes: only 37% of the dataset are duplicate pairs.

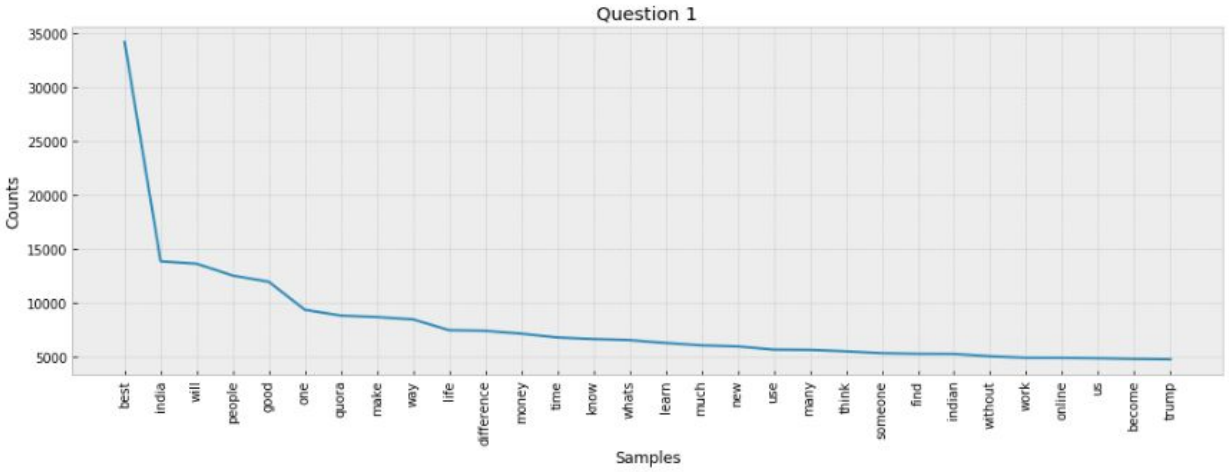


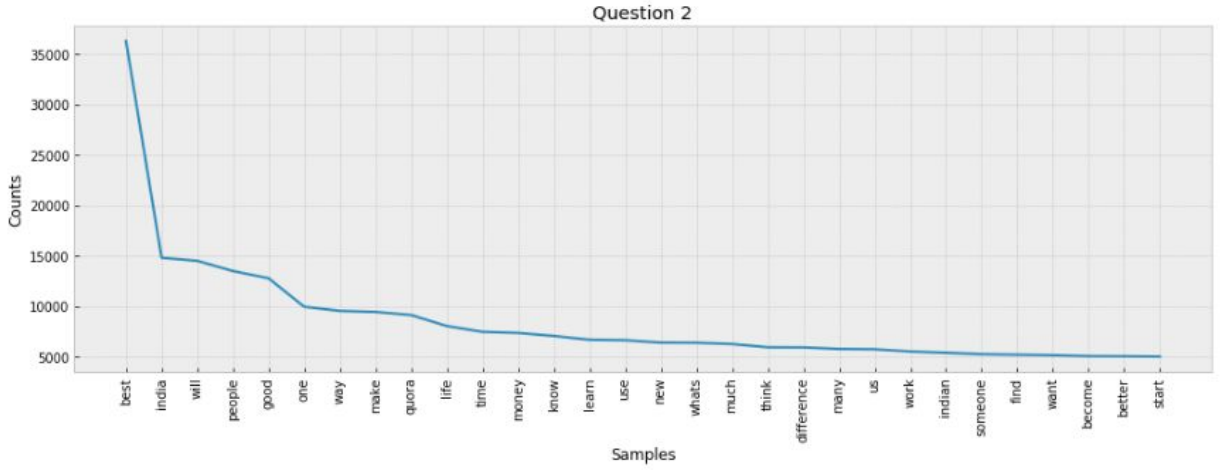
Prior to analyze further, it might be interesting to know the contents of these questions. The most common words for *question1* and *question2* will be visualized using word cloud, since it could give a simple yet understandable visualization of the text; in which bigger font size resembles word with more frequent occurrence.

Question 1

Question 2

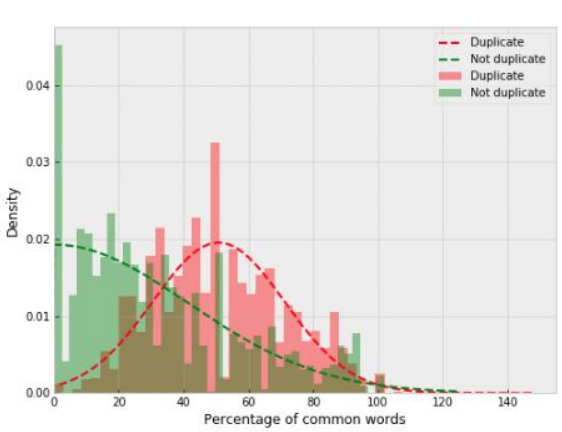
Both questions seem to have a lot of common words, such as difference, India, Donald Trump. Since the common words are quite a lot, we couldn't observe the difference between both questions. The word frequency plot below shows top common words on each question. Although the top words are similar, some words appear more frequent on *question1* compared to *question2*.

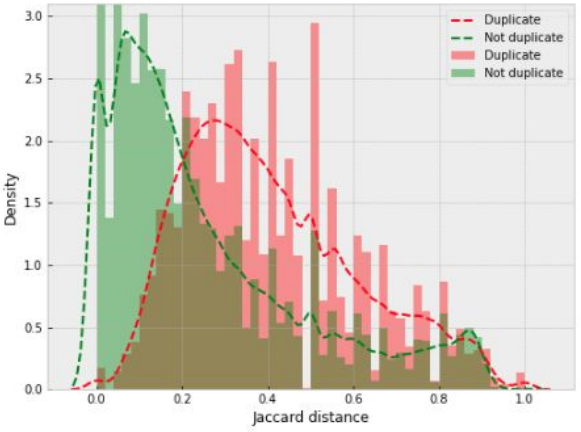




Exploratory Visualization

Compared to non-duplicate, duplicate question pairs tend to have more proportion of common words, and it is nearly normally distributed. Duplicate pairs have 50.78% of common words in average, while the non-duplicate ones only have average of 32.27% common words. Since the distribution of this feature is different, this could be considered as a good input feature on the classification model.





Another features which could differentiate the question pairs are distance metrics. One of the commonly used distance metrics is Jaccard distance, which is measure of similarity between two sets (in this case: set of words on two questions). My initial assumption is duplicate pairs tend to have less Jaccard distance than the non-duplicate ones. However, the plot shows the opposite, which indicates semantic meaning should also be considered while identifying duplicate questions. Duplicate pairs have average Jaccard distance of 0.4, while the distribution shows that more duplicate pairs have more than 0.2 Jaccard distance. This visualization helps on defining threshold value to do simple classification approach, which will be explained on the next section.

Algorithms and Techniques

Prior to diving into deep learning approach, the datasets will be classified using simpler methods. The first approach is identification based on Jaccard distance, i.e. if the question pairs' distance is above certain threshold, it will be marked as duplicate. Jaccard distance is a measure of similarity between two data which is easy to interpret, although it may give erroneous result if the data is too small or have missing observations[2]. The other approach is by training logistic regression model based on some attributes which are correlated to *is_duplicate* field. Since this is a binary classification problem, logistic regression will return probability of the pair being as a duplicate or non-duplicate record. Performance of these two approaches will be used as the baseline metrics to later evaluate performance of deep learning method.

Past researches found Siamese Long-Short Term Memory (LSTM) model as a solution for sentence similarity identification problem[3,4]. My initial deep learning approach will be based on Manhattan Long Short-Term Memory (MaLSTM) architecture[5] which uses word embeddings as the input vector. MaLSTM model is based on Siamese network, in which two inputs are fed on separated network with similar architecture. The resulting outputs of the Siamese network will be compared and get its distance calculated using exponential Manhattan distance - which becomes the measure of similarity between the two inputs. Since sentence similarity features were also proved to be useful[6], I also try incorporating them into the neural network architecture, hoping that it could improve the model performance.

---

[2] Jaccard Index. http://www.statisticshowto.com/jaccard-index/. Accessed on May 10, 2018.
[3] Mueller and Thyagarajan. (2016) Siamese Recurrente Architectures for Learning Sentence Similarity. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16).
[4] Shih et al. (2017) Investigating Siamese LSTM Networks for Text Categorization. Proceedings of APSIPA Annual Summit and Conference 2017.
[5] How to predict Quora Question Pairs using Siamese Manhattan LSTM. https://medium.com/mlreview/implementing-malstm-on-kaggles-quora-question-pairs-competition-8b31b0b16a07. Accessed on March 10, 2018.
[6] Achananuparp et al. (2008) The Evaluation of Sentence Similarity Measures. In DaWaK '08

Benchmark

Using similar datasets, Homma et al. achieved 85% accuracy with 84% F1 score from Siamese Gated Recurrent Unit (GRU) with 2-layer similarity network trained on an augmented dataset[7]. Achieving on par results will be ideal, however since this project is an experimentation on combining sentence-based features and word embeddings, having better performance than the baseline model (Jaccard distance and logistic regressions) would be good enough.

III. Methodology

Data Preprocessing

On Data Exploration section, I mention that there are two question pairs with empty *question2*. Those two questions are actually duplicate questions, I initially let them be and only replace null *question2* values to empty strings. I combined those two questions and marked them as duplicate pairs during model refinement (more details on Refinement section).

Contents of *question1* and *question2* are turned into lowercase and tokenized. The word vector representation of those tokens are retrieved from pre-trained GloVe word vector[8]. Besides, some sentence characteristics measure are extracted from the records, such as Jaccard distance, Cosine distance, word count, unique word count, sentence count, common words between question pair, Levenshtein distance for the first and last word of the question, and part of speech (PoS) tag counts. I also replace tokens from each question to its named entity recognition (NER), and get the common NERs of each question pairs.

Implementation

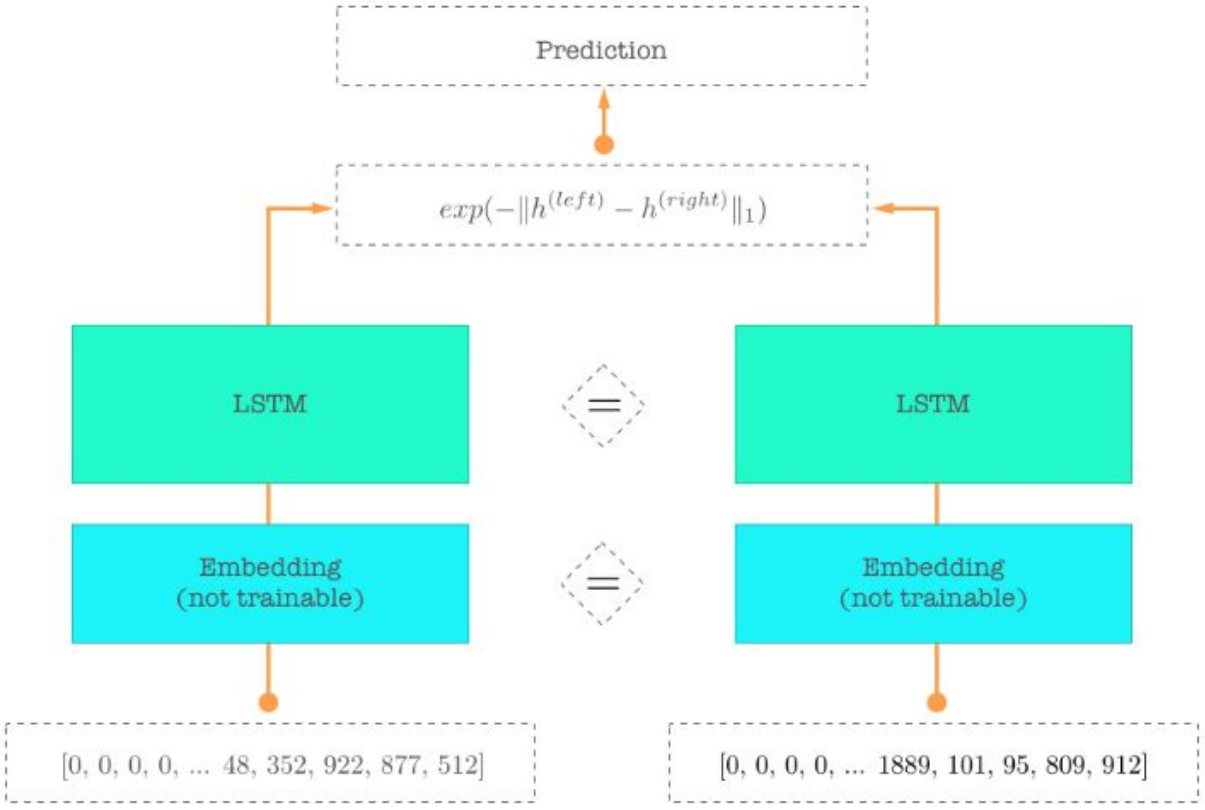
The prepared features are split into training (64%), validation (16%), and test set (20%); stratified based on the class label. The first implemented classification model is based on Jaccard distance. Since the exploratory analysis shows that duplicate pairs mostly have more than 0.2 distance, I set it as the threshold: all records with Jaccard distance above that value are predicted as duplicate.

Before starting with logistic regressions model, I plot correlation matrix of the features, and observe which ones are correlated to *is_duplicate* field. The attributes used on the logistic regressions model are Cosine distance, Jaccard distance, Levenshtein distance on last word, and number of common words.

---

[7] Homma et al. (2016) Detecting Duplicate Questions with Deep Learning. 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.
[8] GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove/. Accessed on March 20, 2018.

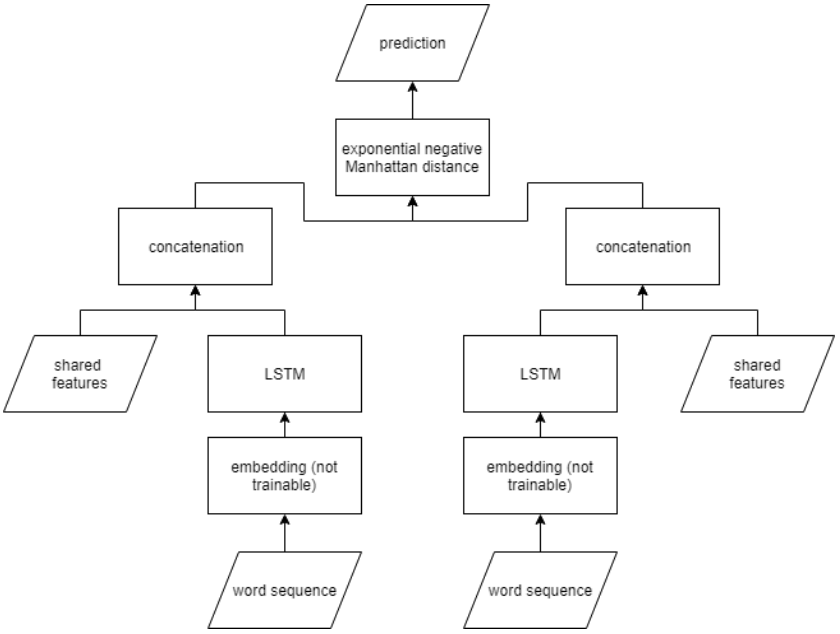
My initial approach on deep learning model uses similar MaLSTM architecture as Cohen[5], which only uses word embeddings as the input features. Since the sequence length of each question is different, it is standardized to the maximum sequence length by adding zero padding on the left of shorter sequences. Each token in the sequence is represented by the word vector, thus result in the word embedding matrix which is used as the input. The architecture is built on top of Keras library, with Adadelta as optimizer and mean squared error as the error function.



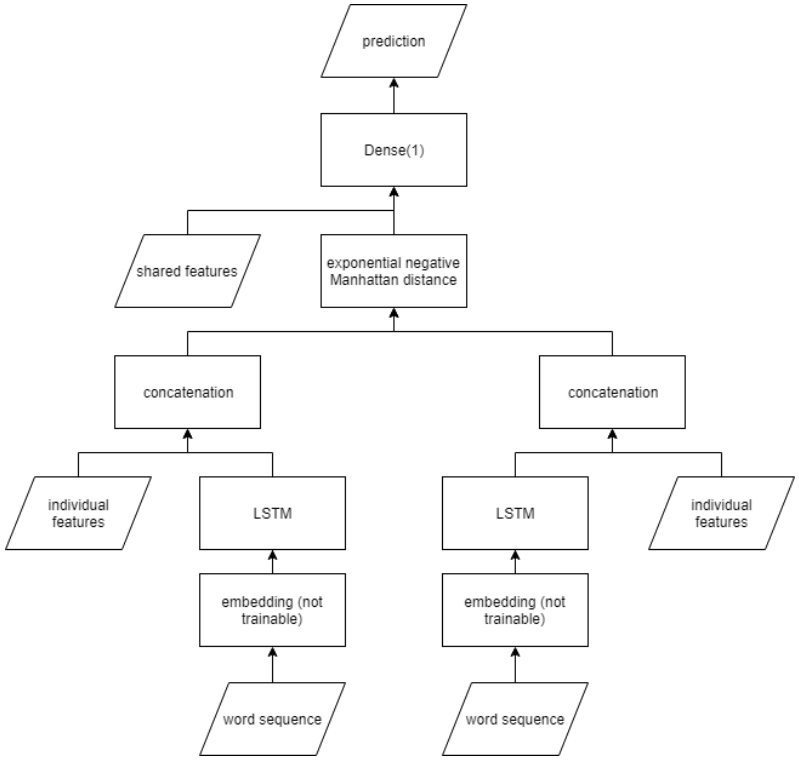
Initial MaLSTM architecture[5]

Refinement

The initial MaLSTM architecture has higher F1 score than Jaccard distance-based model and logistic regressions (Jaccard F1 score: 65%, logistic regressions: 49.35%, initial MaLSTM: 74.85%), but it still has lower performance than the benchmark model. Thus, I try using other optimizers, and it turns out that Adam gives slightly better performance. Since the model performance hasn't achieved on par results with benchmark model, I include sentence characteristics based features to the model and thus create customized model architectures.



Custom Model Architecture #1



Custom Model Architecture #2

On the first customized architecture, LSTM outputs on each question are concatenated with shared features of the question pair: Cosine distance, Jaccard distance, edit distance on the first and last word, number of common words before and after replacement with its NER, percentage of common words before and after replacement with its NER, word count difference and unique word count difference. On the second customized architecture, the LSTM output is concatenated with individual question's features: word count, unique word

count, and sentence count. Another difference is that calculation of exponential Manhattan distance is concatenated with the shared features of the question pair (similar to the first customized architecture's, excluding word count differences).

Since the customized models don't give significantly better performance, I also try to do data augmentation: using 50% of randomly sampled duplicate question pairs, I replace some words on *question1* field which aren't tagged as proper noun, determiner, conjunction, or pronoun from WH-words by Natural Language ToolKit (NLTK) to its synonym. Those tag aren't replaced since their replacements mostly change the semantic meaning of the original question. In addition, I also combine the two question pairs which have empty *question2* field and mark them as duplicate, since they seem to be actually a duplicate pair. The augmented data are then appended to the default training set, and are preprocessed accordingly. Having this additional data, the dataset has roughly balanced class: 53% non-duplicate and 47% duplicate class. Similar classification models are trained using this dataset.

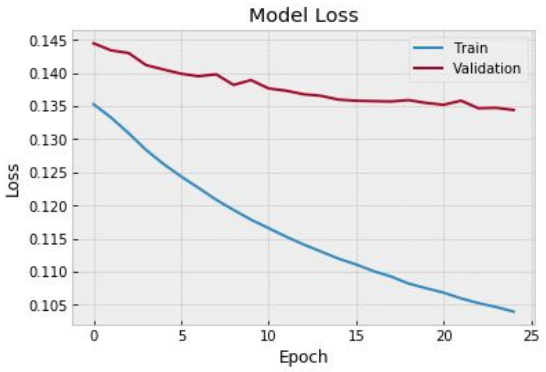
IV. Results

Model Evaluation and Validation

The best model achieves 79.36% F1 score and 81.33% accuracy, which is trained based on the augmented dataset using initial MaLSTM architecture with Adam optimizer. This model is chosen among other alternative models, since it has the highest F1 score and the dataset is not balanced.

Model	Dataset	Test - F1 score	Test - Accuracy
Jaccard distance	Default	65%	66.56%
	Augmented	63.29%	62.34%
Logistic regressions	Default	49.35%	67.24%
	Augmented	56.67%	60.66%
Initial MaLSTM - Adadelta - MSE	Default	82.47%	74.87%
	Augmented	80.60%	78.9%
Initial MaLSTM - Adam - MSE	Default	82.45%	74.92%
	Augmented	81.33%	79.36%
Custom #1 - Adadelta - MSE	Default	82.45%	75.08%
	Augmented	80.38%	78.25%
Custom #2 - Adam - MSE - sigmoid on dense layer	Default	73.27%	-
	Augmented	72.79%	70.47%

The model performance of each model is validated on each training epoch. The best model has decreasing loss as it is trained on more epoches, although it seems that the loss function will hit its plateau soon after 25 epoches.



I think this model is robust enough to be used, as long the input data is written in English (since the model is trained based on English questions).

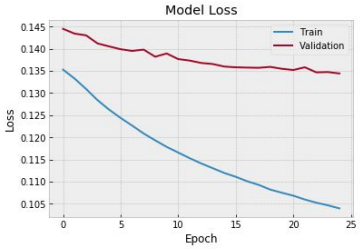
Justification

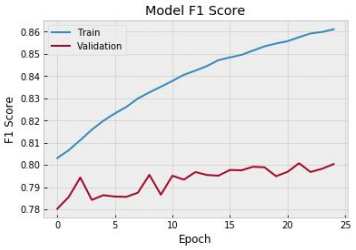
Although the final model doesn't have on par performance with the benchmark model, it still works significantly better than using Jaccard distance or logistic regressions. The benchmark model was optimized after doing hyperparameter search, while before that their best model achieves 81.91% F1 score and 86.68% accuracy using Siamese GRU with 1-layer similarity network on augmented data. On this project I haven't done more parameter tunings and other experiments using combination of other features due to resource limitation, since I train the models on my local computer (Intel core i7 with 16GB RAM) - training LSTM model with 25 epoches require around 80 minutes. Having 79.36% F1 score, I think my model is good enough to help filtering process of duplicate pairs.

V. Conclusion

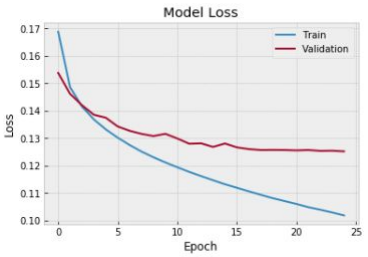
Free-Form Visualization

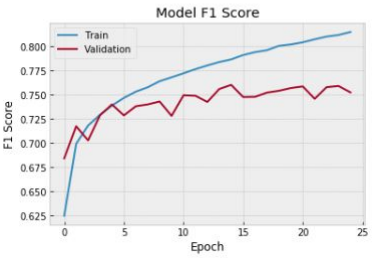
Besides the best model (initial MaLSTM model with Adam optimizer), it might be interesting to look at the customized models' performance. Although the first customized model's F1 score tends to fluctuate more often than the best model's, it has increasing trend and the model's validation loss decreases as it is trained with more epoches. On the other hand, the second customized model results in more unstable F1 score and stagnant model loss. I think I should be more careful on incorporating sentence-based features, especially on combining them to other values in the model architecture, since it might have different scale.



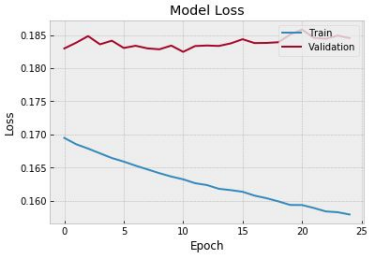


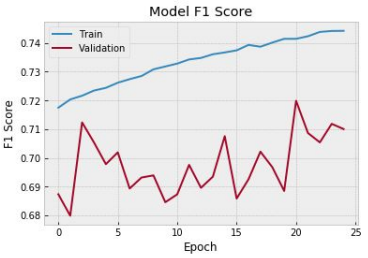
Initial MaLSTM - Adam - MSE





Custom #1 - Adadelta - MSE





Custom #2 - Adam - MSE - sigmoid on dense layer

Reflection

In brief, these are the things I do to solve this project:

1. Literature research: prior to doing the analysis, I search for references from previous researches and other articles on identifying sentence similarity.
2. Exploratory analysis: as initial exploration, I examine the dataset, whether it has missing values and balanced class. I also do quick analysis on the content of the question also distribution of extracted features.
3. Feature extraction: as a result of the literature research, I have a list of possible features to be extracted - which will be used on my customized model. I spend too much time on this phase to collect as many features as possible. On this phase, I preprocess the text and get the word embedding matrix using GloVe.
4. Model training: before the models are trained, I split the dataset into training, validation, and test set. Simple classification models are built based on Jaccard distance and logistic regressions, which are used as baseline value. The deep learning models are trained using MaLSTM model, and I also trained some other customized MaLSTM model which also considers sentence-based features as input.
5. Model evaluation and improvement: since the model performance hasn't achieved the benchmark model, some other optimizers and error functions are implemented to the model. Due to insignificant improvement of this approach, I augment the duplicate question pairs data to improve the training set. Words on *question1* are replaced to its synonyms, so I will have enriched word embedding matrix. The models are re-trained using the augmented data.

I think the most challenging part is on improving the model performance. I initially begin with preparing the possible features, since I think more features could end up with better results. It might be true, but the improvement might not be worth doing. Data augmentation seems to give better improvement on the model performance, and I don't spend that much time doing it. Besides, deciding the appropriate customized model architecture is also challenging for me, since it needs trial and error to evaluate the performance of the model.

Improvement

Other sentence similarity metrics like word order similarity might be a good complementary feature, since Levenshtein distance on first and last word of the question are also correlated to the duplicate class. I haven't used TF-IDF and POS tags as input to the model, perhaps it could also help on improving the model performance. Besides, training a word vector based on the training data might give more relevant results, since the context will be limited to the questions submitted via Quora (which might include unique terms as well as typing errors). For further works, Siamese GRU model is also worth exploring, since it seems to have comparable performance with LSTM and is more computationally efficient[9].

[9]Chung et al. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.