CS584: DETECTION OF DUPLICATE QUESTIONS

Kush Jani[1], Raj Shah[1]
[1]Stevens Institute of Technology
kjani1@stevens.edu, rshah97@stevens.edu

ABSTRACT

In this report, we are going to explore methods to determine if the two questions asked have the same meaning meaning and are duplicates of each other or not. For this project we have used duplicate question dataset by Quora. Detecting duplicate questions can be challenging as the words used and the sentence structure can be varied significantly. Traditional Natural Language Processing techniques have found to have getting limited success. In this project, we have used modern NLP techniques with rich features like LSTM and some traditional classifier algorithms to compare the results.

1 Introduction

Determining duplicate questions has been gaining a lot of attention by the experts in many QA online forums like Quora, StackOverflow, Reddit,etc. since they are gaining a lot of popularity now a days. There is estimated to round 100 million users in Quora per month, since the user base will only grow the number will gradually increase. As a result, the possibility of a question that is asked by the user answered previously increases greatly. For these forums to improve the user experience and to be a great platform for right knowledge sharing, it would be ideal to suggest the questions asked previously that are similar to the current questions asked by the user. Also detecting duplicate questions will help the forums to reduce the burden on the storage processing infrastructure of the application.

In the project, we have used modern approach of NLP to determine the duplicate questions. Duplicate questions are defined when a particular questions have the same exact answer and have same meaning. We first tried our hand with the traditional classifier algorithms such as Random Forest and XG Boost. But we obtained results that were not so satisfying and hence we decided to use a modern algorithm with rich features like LSTM to check whether a question given is duplicate or not. After this we got a desired output.

2 Related Work

Detection of duplicate questions is a very long standing problem. This same problem is been solved by many other ways and we have taken two of the approaches as a reference. The first one is of Dey, Kuntal, Ritvik Shrivastava, and Saroj Kaushik, wherein they have demonstrated the various machine learning algorithms such as traditional approach of SVM. Hand picked and heterogenous features were used. They used words overlap, negation modelling. The data was extremely preprocessed to perform well.

Wang et.al are the only published results on Quora dataset. They got a very good result and they used modern NLP techniques. They observed that the encoding procedure does not provide interaction between the two input sequences. As a result, they proposed a bilateral LSTM model. They used the approach of matching aggregation which proved to be performing better than the CNN and LSTM that they tested. In our project we will however use both the techniques of traditional approach and modern technique and compare the results.

3 Our Approach

In order to detect the duplicate questions we have first explored our dataset and preprocessed it.Data Preprocessing refers to the technique of preparing the raw data to make it suitable for building and training models.In preprocessing we first checked our dataset for null values and removed if any values were missing or not. We checked the dataset for the duplicate values and then we played with the dataset somewhat. We cleaned up the data and found many changes happening to the data.

After the data was cleaned and we got every information about the values of our data we tried our first approach of implementation i.e. we implemented two of the traditional algorithms namely Random Forest and XG Boost. Random Forest is a supervised Machine Learning algorithm and it is the most widely used algorithm because of its accuracy, simplicity and flexibility.it can be used for classification and regression tasks, combined with its non linear nature makes it highly adaptable to a range of data and situations.

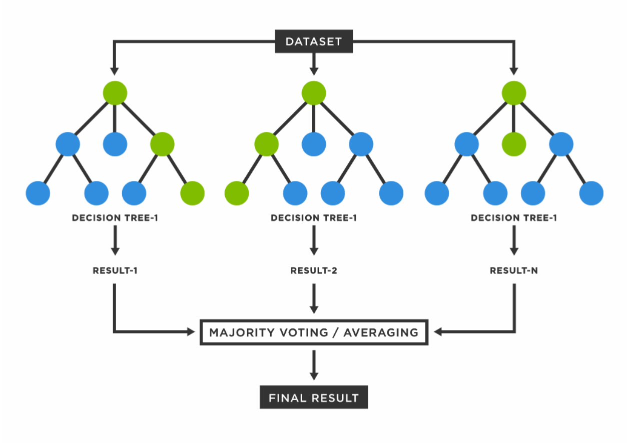


Figure 1: Random Forest

The second algorithm that we used in this approach was XG Boost algorithm. It is an implementation of gradiant boosted decision trees designed for speed and performance. It is a decision tree based ensemble algorithm that uses a gradiant boosting framework and predicts problems involving unstructured data such as images,texts,etc.in neural networks. It generally outperforms all other traditional algorithms or frameworks.

Since our model failed to gain much accuracy on traditional approach, we decided to tweak our dataset by making in some changes and than again try with these models. We did feature engineering i.e we used the knowledge and selected and transformed the relevant variables from raw data and created a predictive model of the same. We added 7 new features according to us which made the dataset more stable. But still we did not get the desired accuracy from doing feature engineering and hence we got into modern NLP techniques

We used advance techniques like word2vec and LSTM. Word2vec is a common method of generating word embeddings and has many applications. It is the modern algorithm that uses neural network model to learn word sucession and learn word associations from large corpus of data. LSTM is also a neural network with feedback connections that can handle both single data points and full data sequences. It has the capacity of learning long term dependencies in data. This is achieved because the recurring module of the model has a combination of four layers interacting with each other.

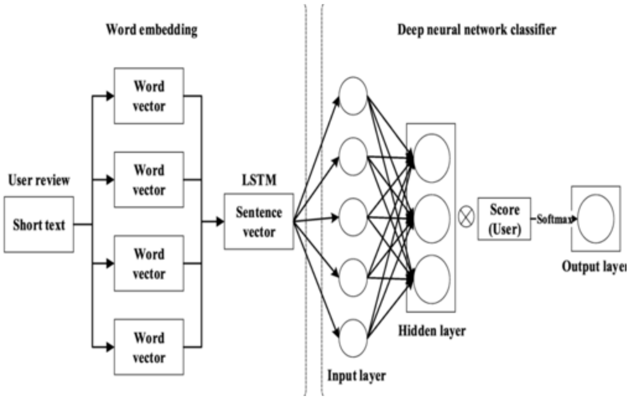


Figure 2: LSTM

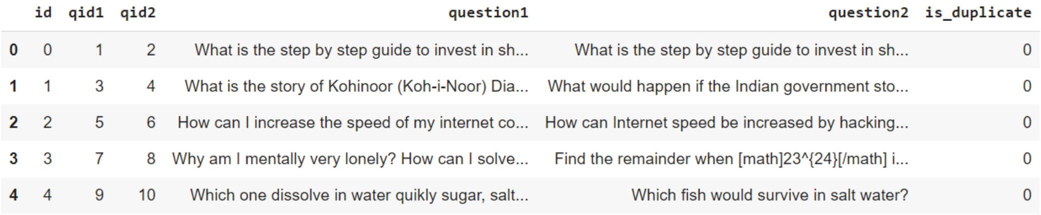
4 Experimental Design

4.1 Description of Dataset

For this project, we have used the dataset which was released by Quora . It is a publically available dataset on Quora Question Pairs dataset on Kaggle. The dataset contains around 400,00+ labelled question pairs. There are six features in our dataset. the fields description are shown below in the table. The link of the dataset is : https://www.kaggle.com/datasets/quora/question-pairs-dataset

Fields	Description
id	unique identifier for the question pair
qid1	unique identifier for the first question
qid2	unique identifier for the second question
question1	full unicode text for the first question
question2	full unicode text of the second question
is_duplicate	1 if the questions are duplicate, 0 otherwise

Here in this dataset we assume that the questions that are marked as duplicates in the dataset are truely duplicates of each other.Of all the question around 250,000 questions are not duplicates and 150,000 are duplicates of each other. The dataset has been labelled by humans manually and hence there can be chances of some noise in the dataset. The information of the data is given in following images



	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0

Figure 3: Dataset example

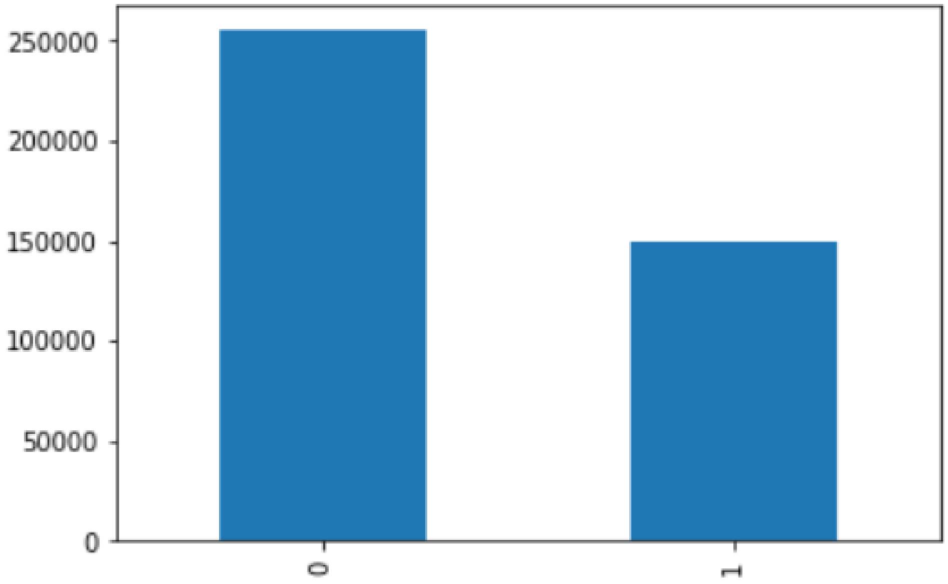


Figure 4: Duplicate and Non duplicate values

4.2 Data Preprocessing

Our dataset was first cleaned up. We first of all removed all the null values and all the unwanted objectives from the dataset. We removed all the punctuation, commas , full stop and all the things that we did not want in our dataset. Since our dataset was labelled manually it was tend to have some nois enad overfitting. We divided our dataset into three sets of training, validation and test sets. The training set contained around 300,000 entries whereas the validation and test entries contained around 50,000 entries.

For the sake of undersanding the data we first used only 30,000 rows of data but we did not get the results that we wnated and hence we did feature engineering inorder to improve our accuracy. Feature Engineering generally refers to the process of using domain knowledge to select and trasnform the most relevant variables from raw data when creating a predictive model using machine learning. For gaining more accuracy we added seven new features in our dataset and then again run the model. The sevens features that we added are givn in the tabel below:

Fields	Description
Length_Q1	Character Length of Question1
Length_Q2	Character Length of Question2
#_words_Q1	Number of words in Question1
#_words_Q2	Number of words in Question2
#_Common_Words	Number of Common UNIQUE Words
Total_Words	Total Number in Q1 + Total + number of words in Question2
share_word	Word Common/Word Total

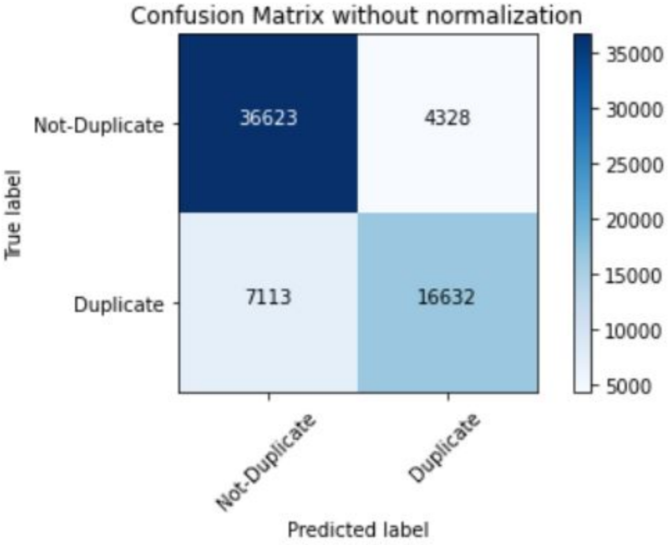


Figure 5: Confusion Matrix

5 Results

We observed that on the first try the traditional classifiers that we used like XG Boost and Random Forest proved to be inefficient and the accuracy that we got was very low. As a result we did feature engineering and got a very good result. Before manipulating the data our accuracy of XG Boost was around 68% which got increased to around 74%. The same trend was seen in Random Forest classifier whose accuracy got increased from 73% to around 77%. Some of the hand picked features had gone a long way inorder to improve the accuracy of our results. Still the results we were expecting was nor obtained. We tried to learn our own embeddings and tried to fit in word2vec to our dataset, but it gave us an error due to overfitting as our dataset was too small.

Our final classifier LSTM implementated well on our dataset and did not gave overfitting error. We also observed that we got an overall accuracy of around 82% from LSTM classifier which was pretty good as comapred to the previous classfiers that we implemented. We also observed that there were difference in the results as well of these algorithms

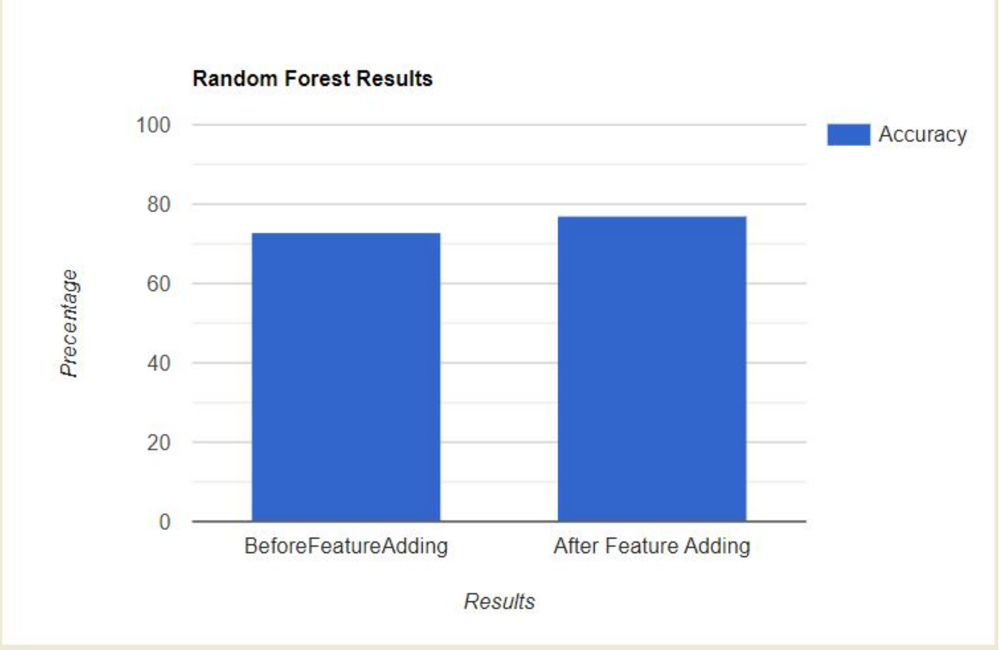


Figure 6: Random Forest Results

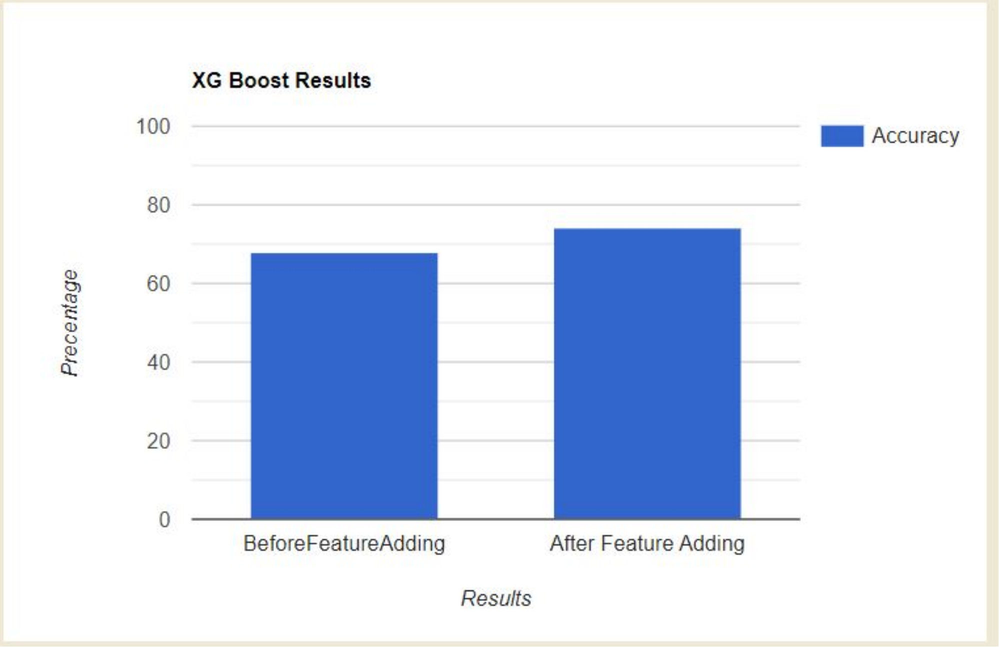


Figure 7: XG Boost Result

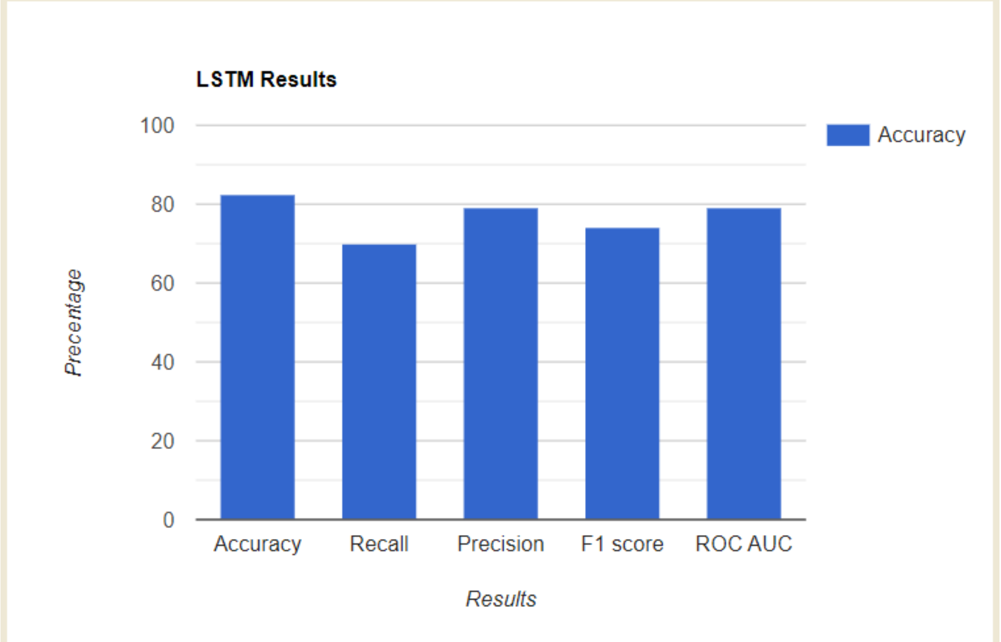


Figure 8: LSTM Results

with implementation using Keras and Pytorch. Keras gave better results as compared to Pytorch results.

6 Conclusion and Future Scope

In this project, we have tried our best to give the solution for long lasting problem encountered by QA forums about detecting duplicate questions on the dataset provided by Quora. We explored two different approaches to solve this problem. The first one was using traditional classifiers namely XG Boost and Random Forest and the other approach was using modern NLP technique like word2vec and LSTM classifier. Our results are promising with the modern technique of LSTM performing very well when compared to traditional approaches.

This work can be extended in a variety of way. The first being that we can include data augmentation in order to balance the data. The architecture can also tried by using GRU instead of LSTM. Some of the recent NLP studies have shown that this technique is reliable and shown good results but the cost being very high. But, using this technique will give surely high accuracy as compared to our approach.

References

[1] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. 2017.

[2] Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. A paraphrase and semantic similarily detection system for user generated short-text content on microblogs. In COLING, volume 16, pages2880-2890, 2016

[3] https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d.

[4] https://www.tibco.com/reference-center/what-is-a-random-forest

[5] https://towardsdatascience.com/word2vec-explained-49c52b4ccb71

[6] https://www.kaggle.com/c/quora-question-pairs