Duplicate Question Pairs Detection using Machine Learning

Shruti Sureshan
Department of Computer Science and Engineering, Indian Institute of Technology, Jodhpur
sureshan.1@iitj.ac.in

Abstract

Quora is a growing platform comprising of a user-generated collection of questions and answers. Identifying semantically similar questions on Quora is challenging as natural language is very expressive. In this project, our approach primarily focuses on Machine learning techniques for detecting whether a pair of questions asked on Quora is duplicate or not. Our emphasis is to address the drawbacks, as mentioned earlier, by taking the text input and preprocessing it, followed by feature extraction using Bag of Words and TF-IDF algorithm. Further, the questions are classified as duplicate and non-duplicate by applying machine learning classifiers such as Logistic Regression, XGBoost, Support Vector Machine. The proposed algorithms have been experimented using the Quora Question Pair dataset released by Quora. The performance of the algorithms is evaluated and compared using different parameters like accuracy, precision, recall, and F1-score. The proposed application of SVM with Bag of Words Technique in duplicate question pair detection has proved better in terms of all parameters as compared to the other existing algorithms.

1. Introduction

Quora is a social media website where questions are posted by users and answered by experts who provide quality insights. Other users can cooperate by editing questions and suggesting more accurate answers to the submitted questions. Quora receives 300 million unique visitors every month, which raises the problem of different users asking similar questions with same intent but in different words. Multiple questions with similar wording can cause readers to spend more time to find the best answer, and make writers answer multiple versions of the same question [1]. Quora uses random forest model to identify these duplicate questions. But there is need of better model for this recognition [6].

With the growing database, there is a need for Quora to preserve the trust of the users by maintaining quality content by discarding duplicate information. The problem of determining whether the two sentences have the same meaning or not requires a model to capture the lexical and syntactic meanings of the given sentences. It is a binary classification problem where for a given pair of questions we need to predict if they are duplicates or not. In this project, we aimed to present a comprehensive set of machine learning models, and to study their performance on the dataset.

2. Background & Related Work

Much work has been published to date on text classification. Classifying duplicate questions can be a tricky task since the variability of language makes it difficult to know the actual meaning of a sentence with certainty. This task is similar to the paraphrase identification problem, which is a thoroughly researched Natural Language Processing (NLP) task [8].

Many findings comparing the accuracy of different machine learning algorithms against the similarity search have been conducted. In the literature [4], authors have used word ordering and word alignment using a long-short-term-memory (LSTM) recurrent neural network, and the decomposable attention model respectively and tried to combine them into the LSTM attention model to achieve their best accuracy of 81.4%. Their approach involved implementing various models proposed by various papers produced to determine sentence entailment on the SNLI dataset. Some of these models are RNN with GRU and LSTM cell, LSTM with attention, Decomposable attention model. Deep models, trained with task-specific feature engineering, provided impressive results in semantic analysis and similarity measure. Deep models can be combined with word embeddings and used to express the semantic meaning of text chunks with satisfactory accuracy [1]. The common features used are bag of words (BOW), term frequency and inverse document frequency (TF IDF), unigrams and bigrams. Support Vector Machine (SVM), used with different feature extraction techniques such as BOW or n-gram vectors, is one of the main methods in text categorization [9]. We will use some of these methods in our approach to the problem.

3. Approach

The main objective is to determine whether the given pair of questions are duplicates or not, thereby finding quality solutions to question ensuring enhanced user experience.

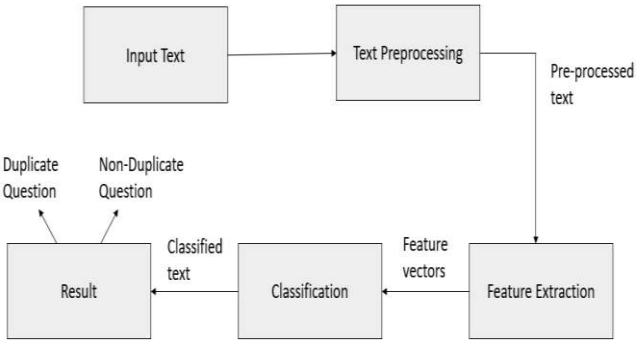

Fig. 1: Workflow of the system

The workflow of the system is shown in Fig.1 which illustrates the process of taking the text input, text preprocessing, feature extraction using Bag of Words and TF-IDF algorithm, and classification using Machine Learning algorithms like XGBoost.

3.1. Dataset

Quora released a public dataset that consists of 404,290 question pairs in January 2017. Each record has a pair of questions and a target class that represents whether the questions are duplicate or not. The attribute details are shown in Table 1.

TABLE 1: Attributes Description of Dataset

Attribute names	Description
id	unique ID of question pair
qid1	ID of first question
qid2	ID of second question
question1	content of first question
question2	content of second question
is_duplicate	The target label, which is set to 1 if question1 and question2 have similar intent, and 0 if not

Example for Duplicate question pairs:

- How can I be a good geologist?
- What should I do to be a great geologist?

Since the classifier is only concerned with "question1", "question2" and "is_duplicate", the rest of the attributes of the dataset are ignored [1]. We performed the necessary statistics on the dataset, which helps us to give a more detailed understanding of the duplicate Quora question dataset as follows:

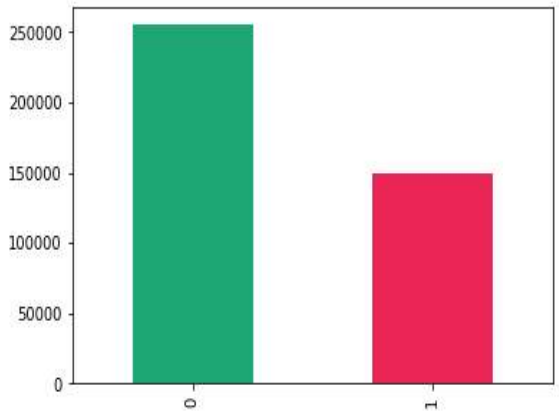

Fig. 2: Count plot

Of the 404,290 question pairs, 255,027 (63.08%) have a negative (0) label, and 149,263 (36.92%) have a positive (1) label. Across all question pairs, there are 537,361 unique questions. 20.82% questions appear more than once, with one of the questions appearing 161 times.

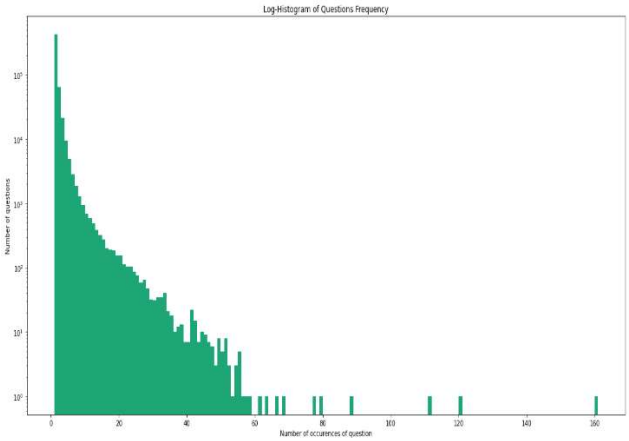

Fig. 3: Histogram

In the histogram plot Fig. 3, the x-axis represents the number of times question occurs, and the y-axis represents how many such questions with occurrence count exist in the dataset. As can be visualized from the graph the majority of questions occurs less than 60 times, and the first bar shows the unique occurrence and second bar the number of the appearance of question twice and so on.

3.2. Preprocessing

Removal of HTML Tags, punctuation marks, comma between numbers, removal extra whitespace, lowercase all texts, removal of special characters, expand contractions are performed first. This is followed by stop words removal and stemming. Stop words are words which do not provide us

with any semantic meaning. Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words performed using Porter stemmer algorithm.

3.3. Feature Extraction

Feature Extraction phase allows the data to be observed to extract the basic features from the data. These features give a basic idea about the similarities and dissimilarities present in the question pairs. Here, features are divided into four categories. This involves the working of basic NLTK mathematics, FuzzyWuzzy parameters, Bag of Words technique, TD-IFD algorithm.

3.3.1. Basic Feature Extraction

The extracted features are length of question in question pairs, number of words present in question 1 and 2, difference between the number of characters in question 1 and question 2, difference between the number of words in question 1 and question 2, number of common words, common words ratio.

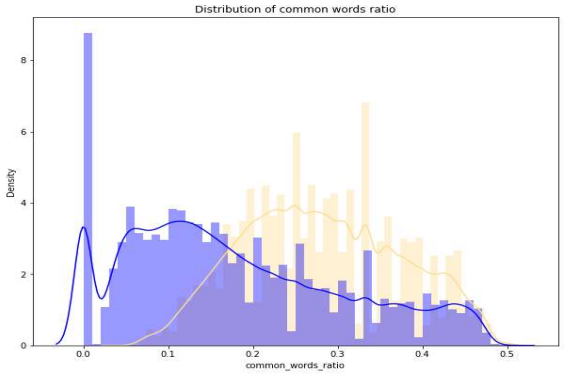

Fig. 4: Basic features extracted for the training model

These features gave information about the available data and gave no additional information. Hence, training the model for better output is not possible with these features.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 404287 entries, 0 to 404289
Data columns (total 14 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   question1           404287 non-null  object
 1   question2           404287 non-null  object
 2   is_duplicate        404287 non-null  int64
 3   q1_len              404287 non-null  int64
 4   q2_len              404287 non-null  int64
 5   q1_word_len         404287 non-null  int64
 6   q2_word_len         404287 non-null  int64
 7   q1_char_len         404287 non-null  int64
 8   q2_char_len         404287 non-null  int64
 9   len_diff            404287 non-null  int64
 10  word_len_diff       404287 non-null  int64
 11  char_len_diff       404287 non-null  int64
 12  common_words        404287 non-null  int64
 13  common_words_ratio  404287 non-null  float64
dtypes: float64(1), int64(11), object(2)
```

Fig. 5: Basic features extracted for the training model

3.3.2. Advanced Feature Extraction

FuzzyWuzzy is a Python library which has methods to compare string equivalence. We have used different string comparison techniques from FuzzyWuzzy to extract fuzzy features.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 404287 entries, 0 to 404289
Data columns (total 18 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   question1           404287 non-null  object
 1   question2           404287 non-null  object
 2   is_duplicate        404287 non-null  int64
 3   q1_len              404287 non-null  int64
 4   q2_len              404287 non-null  int64
 5   q1_word_len         404287 non-null  int64
 6   q2_word_len         404287 non-null  int64
 7   q1_char_len         404287 non-null  int64
 8   q2_char_len         404287 non-null  int64
 9   len_diff            404287 non-null  int64
 10  word_len_diff       404287 non-null  int64
 11  char_len_diff       404287 non-null  int64
 12  common_words        404287 non-null  int64
 13  common_words_ratio  404287 non-null  float64
 14  fuzz_ratio          404287 non-null  int64
 15  fuzz_partial_ratio  404287 non-null  int64
 16  token_sort_ratio    404287 non-null  int64
 17  token_set_ratio     404287 non-null  int64
dtypes: float64(1), int64(15), object(2)
```

Fig. 6: Features extracted for the training model

Fuzzy String Matching is sometimes known as approximate string matching. It is the process of finding strings that approximately match a given pattern [5]. The fuzzy wuzzy ratio calculate the similarity ratio between the two strings. The partial ratio allows us to perform substring matching. This works by taking the shortest string and matching it with all substrings that are of the same length. The token sort will sort the strings alphabetically and then joins them together. The token set ratio is similar to the token sort ratio, except it takes out the common tokens before calculating the fuzzy ratio between the new strings.

3.3.3. Bag of Words Technique

As the system used for this work machine doesn't accept text for training, the text is converted into a form understandable by the machine. So, vectored form data is used [7].

The "Bag of words" representation is a Natural Language Processing technique of text modelling. It is a common way to represent a text document. Here the words that are present in the document are assigned their frequency of occurrence. Vocabulary words which are not present in the document are assigned a frequency 0. This way a text document can be represented using a sparse and multi-dimensional feature vector. At a much granular level, the machine learning models work with numerical data rather than textual data. Hence, to be more specific, by

using the bag-of-words (BoW) technique, we convert a text into its equivalent vector of numbers. Thus, we can now train our model and perform classification using Machine learning algorithms.

3.3.4. Term frequency–inverse document frequency (TF-IDF)

TF-IDF is the ratio of the number of times a word (term) occurs in a particular question, to the number of times the word occurs in all the questions (our entire corpus). A higher TF-IDF value indicates a more important word. The TF–IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

The model learns the inverse frequency of words from the set of combined unique question1 and question2 character set. The corresponding TF-IDF, word level feature, obtained for each of the questions in the pair is then passed as input to the different machine learning classifiers.

4.1. Machine Learning classifiers

We have selected the following three machine learning classifiers.

4.1.1. Logistic Regression

Logistic regression is a machine learning classifier. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables. It uses a sigmoid function to model a binary dependent variable. In the case of a binary logistic model, it has a dependent variable with two values - 0 and 1 (for negative and positive class respectively). The sigmoid function is defined as:

$$f(x) = \frac{1}{1+e^{-(x)}}$$

Gradient descent algorithm can be used for optimization in logistic regression. It is a first order iterative optimization algorithm. Its main goal is to minimize the cost.

Want $\min_\theta J(\theta)$:

Repeat {

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update all θ_j)

}

Our classifier is expected to output the class based on the probabilities.

4.1.2. Support Vector Machines (SVM)

Our objective in SVM is to find a hyperplane that separates positive and negative examples with the largest margin while keeping the misclassification as low as possible. We will minimize the cost/objective function shown below:

$$J(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\left[\frac{1}{N}\sum_i^n \max(0,\ 1 - y_i * (\mathbf{w} \cdot x_i + b))\right]$$

Another version of a cost function:

$$J(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_i^n \max(0,\ 1 - y_i * (\mathbf{w} \cdot x_i + b))$$

Here λ is equal to 1/C

Gradient of the cost function:

$$J(\mathbf{w}) = \frac{1}{N}\sum_i^n \left[\frac{1}{2}\|\mathbf{w}\|^2 + C \max(0,\ 1 - y_i * (\mathbf{w} \cdot x_i))\right]$$

$$\nabla_w J(\mathbf{w}) = \frac{1}{N}\sum_i^n \begin{cases} \mathbf{w} & \text{if } \max(0,\ 1 - y_i * (\mathbf{w} \cdot x_i)) = 0 \\ \mathbf{w} - C y_i x_i & \text{otherwise} \end{cases}$$

For this we can use Stochastic Gradient descent. SGD is an iterative method for optimizing the objective function. SGD operates by using one randomly selected observation from the dataset at a time.

Using the weights, we calculate the cost over all the data points in the training set. Then we compute the gradient of cost w.r.t the weights and finally, we update weights. And this process continues until we reach the minimum. Update step:

$$for\ i\ in\ range(m)\ :$$

$$w_j := w_j - \alpha \frac{\partial J_i}{\partial w_j}$$

J_i is the cost of ith training example

4.1.3. XGBoost

Xgboost is an ensemble technique for machine learning. Ensemble learning offers a systematic solution to combine the predictive power of multiple learners. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

Xgboost uses gradient boosting and it converts weak learners into strong learners [3].

5.1. Experiments

We trained and tested machine learning classifiers like Logistic regression model, XGBoost, Support Vector Machines (SVM).

5.1.1. Logistic regression with Bag of Words

Fig. 7 depicts the performance report of the Logistic regression model with Bag of Words.

Accuracy Score: 0.6902718345741918

Classification Report:

	precision	recall	f1-score	support
0	0.75	0.77	0.76	25407
1	0.59	0.56	0.57	15022
accuracy			0.69	40429
macro avg	0.67	0.66	0.66	40429
weighted avg	0.69	0.69	0.69	40429

Fig. 7: Performance of the model

5.1.2. Logistic regression with TF-IDF

Fig. 8 depicts the performance report of the Logistic regression model with TF-IDF.

Accuracy Score: 0.6736253679289619

Classification Report:

	precision	recall	f1-score	support
0	0.73	0.77	0.75	25407
1	0.57	0.51	0.54	15022
accuracy			0.67	40429
macro avg	0.65	0.64	0.64	40429
weighted avg	0.67	0.67	0.67	40429

Fig. 8: Performance of the model

5.1.3. XGBoost with Bag of Words

Fig. 9 depicts the performance report of the XGBoost model with Bag of Words.

Accuracy Score: 0.7538400652996612

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.76	0.80	25407
1	0.65	0.74	0.69	15022
accuracy			0.75	40429
macro avg	0.74	0.75	0.74	40429
weighted avg	0.76	0.75	0.76	40429

Fig. 9: Performance of the model

5.1.4. XGBoost with TF-IDF

Fig. 10 depicts the performance report of the XGBoost model with TF-IDF.

Accuracy Score: 0.7531227584159885

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.77	0.80	25407
1	0.65	0.72	0.69	15022
accuracy			0.75	40429
macro avg	0.74	0.75	0.74	40429
weighted avg	0.76	0.75	0.76	40429

Fig. 10: Performance of the model

5.1.5. SVM with Bag of Words

Fig. 11 depicts the performance report of the SVM model with Bag of Words.

Accuracy Score: 0.7752108634890796

Classification Report:

	precision	recall	f1-score	support
0	0.84	0.80	0.82	25407
1	0.68	0.74	0.71	15022
accuracy			0.78	40429
macro avg	0.76	0.77	0.76	40429
weighted avg	0.78	0.78	0.78	40429

Fig. 11: Performance of the model

5.1.6. SVM with TF-IDF

Fig. 12 depicts the performance report of the SVM model with TF-IDF.

```
Accuracy Score: 0.7529743500952286

Classification Report:

              precision    recall  f1-score   support

           0       0.75      0.91      0.82     25407
           1       0.76      0.49      0.60     15022

    accuracy                           0.75     40429
   macro avg       0.76      0.70      0.71     40429
weighted avg       0.75      0.75      0.74     40429
```

Fig. 12: Performance of the model

We have evaluated the parameters of Accuracy, Precision, Recall, and F1-score, and we have accordingly summarized the results of the outputs as follows:

TABLE 2: Performance of classifiers

Classifier	Feature extraction	Accuracy
Logistic regression	Bag of Words	69.02%
Logistic regression	TF-IDF	67.36%
XGBoost	Bag of Words	75.38%
XGBoost	TF-IDF	75.31%
SVM	Bag of Words	77.52%
SVM	TF-IDF	75.29%

From Table 2, we can observe that Support Vector Machines (SVM) with Bag of Words technique performs the best among all the other classifiers with an accuracy of 77.52%. The above table also shows the comparison among the two feature extraction techniques for different models. Logistic regression hails the least in terms of accuracy. XGBoost works considerably well with both the feature extraction techniques. Also, Support Vector Machines (SVM) with TF-IDF performs well but not as good as Support Vector Machines (SVM) with Bag of Words. Feature extraction played a significant role in terms of all the performance evaluation parameters with better values. Thus, we can conclude that Natural Language Processing with Machine Learning enhances the performance of the classifier.

6. Conclusion & Future Work

This work proposed a model that address the problem of question duplication in Q&A forums by using basic algorithms from Natural language processing for feature extraction and Machine learning algorithms to classify whether question pairs are duplicates or not. We selected highly dominant features from the questions and compared the different classifiers to arrive at the best-performing model. In this project, we trained and tested three machine learning models to identify duplicate questions using a real-time dataset released by Quora. Our best performing model was SVM with TF-IDF. This has demonstrated that machine learning models are efficient in solving natural language problem of detecting semantically duplicate question.

As a continuation to our project, in the future, a similar approach can be implemented for various other search engines that are available like reedit, stack overflow, etc. This will ensure that search engines are user-friendly. We believe the Quora dataset is a useful resource to further explore the task of Natural Language Processing with Machine Learning and Deep learning algorithms. All of these are promising potential areas of future work.

7. Acknowledgements

Text classification is one of the fundamental tasks in natural language processing with broad applications in various fields. I would like to thank Dr. Gaurav Harit for continuous guidance and providing me the opportunity to explore this area and learn new concepts.

References

[1] Z. Imtiaz, M. Umer, M. Ahmad, S. Ullah, G. S. Choi and A. Mehmood, "Duplicate Questions Pair Detection Using Siamese MaLSTM," in IEEE Access, vol. 8, pp. 21932-21942, 2020, doi: 10.1109/ACCESS.2020.2969041.

[2] Anishaa VKR, Sathvika P, Rawat S. (2021) Identifying Similar Question Pairs Using Machine Learning Techniques. Indian Journal of Science and Technology. 14(20):1635-1641.

[3] Shashank Pathak ,Ayush Sharma ,Shashank Shekhar Shukla , (2018) " Semantic String Similarity for Quora Question Pairs " , International Journal of Advances in Science, Engineering and Technology(IJASEAT) , pp. 77-80, Volume-6, Issue-4

[4] A Tung and E Xu. 2017. Determining Entailment of Questions in the Quora Dataset. , 8 pages

[5] A. Dhakal, A. Poudel, S. Pandey, S. Gaire and H. P. Baral,

"Exploring Deep Learning in Semantic Question Matching," 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), 2018, pp. 86-91, doi: 10.1109/CCCS.2018.8586832.

[6] Ansari, Navedanjum, and Rajesh Sharma. "Identifying Semantically Duplicate Questions Using Data Science Approach: A Quora Case Study." arXiv preprint arXiv:2004.11694 (2020).

[7] Bhalerao, Vivek & Ar, Sathya & Panda, Sandeep. (2021). A Machine Learning Model to Identify Duplicate Questions in Social Media Forums. International Journal of Innovative Technology and Exploring Engineering. 9. 370-373. 10.35940/ijitee.D1362.029420.

[8] Wenhao Zhu, Tengjun Yao, Jianyue Ni, Baogang Wei, and Zhiguo Lu. Dependency-based siamese long short-term memory network for learning sentence representations. PLOS ONE, 13(3):1–14, 03 2018.

[9] Badri N. Patro, Vinod K. Kurmi, Sandeep Kumar, and Vinay P. Namboodiri. Learning semantic sentence embeddings using pair-wise discriminator. CoRR, abs/1806.00807, 2018

[10] Prabowo, Damar Adi, and Guntur Budi Herwanto. "Duplicate question detection in question answer website using convolutional neural network." 2019 5th International Conference on Science and Technology (ICST). Vol. 1. IEEE,201