Question Similarity Machine Learning

Jay Darji
201806305
CSCI 361
St. Francis Xavier University

1 Abstract

Natural Language Processing (NLP) can be effectively applied in the field of question similarity by leveraging its ability to understand and analyze linguistic patterns, semantic structures, and syntactic relationships within text data. Through various techniques such as word embeddings, sentence embeddings, and advanced deep learning models like BERT, NLP can identify and quantify the degree of similarity between different questions. This similarity measurement can facilitate improved search engine functionality, assist with question deduplication on Q and A platforms, and enhance chatbot comprehension, leading to more efficient and accurate user experiences. Moreover, NLP models can be fine-tuned to adapt to different domains and languages, ensuring broader applicability and versatility in question similarity assessment.

2 Why doing this project

Undertaking a project of question similarity classification using various machine learning algorithms such as KNN-1,5,10, SVM Linear, SVM RBF, and Random Forest provides an opportunity to compare and evaluate the performance of these methods in identifying similar questions. By using different algorithms and configurations, researchers can gain insights into the strengths and weaknesses of each model, as well as their suitability for specific tasks and data types. These insights can lead to a better understanding and fine-tuning of models, ultimately resulting in more effective and accurate question similarity detection systems. Additionally, such a project contributes to the ongoing exploration of the best practices in the field of NLP, stimulating innovation and advancing the state-of-the-art in-question similarity classification techniques.

3 Introduction

In this project, my primary objective was to develop an NLP-based question similarity classification system using the Quora question pair dataset from Kaggle. To achieve this goal, I carried out a series of preprocessing steps on the input text, including converting text to lowercase, removing leading and trailing white spaces, replacing special characters with their string equivalents, decontracting words, removing HTML tags with the help of the BeautifulSoup library, and eliminating punctuations using regular expressions. After preprocessing, I delved into data visualization tasks to gain a better understanding of the dataset, I inspected the first five rows of the data frame, checked for null values and duplicate rows, and calculated the count and percentage of duplicate and non-duplicate rows in the 'is-duplicate' column. Additionally, I saved relevant visualizations to provide an overview of the data characteristics, such as a bar plot of duplicate and non-duplicate rows and a histogram depicting question repetition patterns. **Image – picture4.jpg**

To further enhance my machine learning models' performance, I carried out feature engineering tasks, adding new columns containing length and token-based features, and calculating fuzzy matching features for each question pair. These features serve as valuable inputs that enable the models to predict if two questions are duplicated more accurately. Lastly, I computed n-gram features and Jaccard similarity between question pairs in the dataset, providing additional information that can be used as inputs for my machine-learning models. By employing a comprehensive approach that encompasses preprocessing, data visualization, and feature engineering, I aimed to build a robust and effective question similarity classification system.

4 Dataset

In this project, I used the Quora Question Pairs dataset, which is publicly available on Kaggle at https://www.kaggle.com/competitions/quora-question-pairs/data. This dataset is designed to identify duplicate questions on the Quora platform, a popular question-and-answer website. The dataset comprises over 400,000 pairs of questions, with each pair labelled as either duplicate or non-duplicate. Image- dataset.jpg The dataset contains the following columns:

1. id: A unique identifier for each question pair.
2. qid1: The unique identifier for the first question in the pair.
3. qid2: The unique identifier for the second question in the pair.
4. question1: The text of the first question.
5. question2: The text of the second question.
6. is-duplicate: A binary label indicating whether the question pair is considered a duplicate (1) or not (0).

This dataset offers a comprehensive collection of question pairs with varying degrees of similarity, making it an ideal resource for developing and testing question similarity classification models. By using this dataset, my project aimed to build a robust and effective system to detect duplicate questions on Quora, thereby enhancing the platform's user experience and reducing content redundancy.

5 Data Analysis

In this project, I conducted a thorough data analysis to understand the patterns and relationships between various features in the Quora question pair dataset. I utilized the Seaborn library to create visualizations, allowing us to analyze different aspects of the dataset, such as question length, word count, common words, and token-based features. These visualizations helped us identify the key differences between duplicate and non-duplicate question pairs. We extracted an extensive set of features to maximize the predictive power of my machine-learning models. By investigating the relationships between question lengths, word counts, common words, token and stopword counts, first and last words, mean length, length differences, and longest substring ratios, I aimed to capture the nuances in the questions that might indicate similarity or dissimilarity. Furthermore, I compared fuzzy ratios, fuzzy partial ratios, token sort ratios, and token set ratios between duplicate and non-duplicate questions, providing additional insights into their linguistic patterns.

The primary reason behind extracting such a diverse set of features is to enhance my models' ability to detect duplicate questions accurately. A comprehensive feature set allows the models to better understand the underlying structure and semantic relationships within the questions, thereby improving their classification performance. Moreover, a rich feature set enables us to identify the most relevant features contributing to the detection of similar questions, facilitating model optimization and ensuring robustness in the question similarity classification system. **Image – picture1.jpg, picture2.jpg, picture3.jpg, picture5.jpg**

Overall, my data analysis and feature extraction efforts aimed to create a solid foundation for building effective and accurate question similarity classification models.

6 Models

In this project, I utilized six different machine learning models to evaluate their performance in question similarity classification. The models include:

6.1 K-Nearest Neighbor (KNN) with k = 1, 5, and 10

KNN is a simple and intuitive algorithm used for classification tasks. The primary benefit of KNN is its ease of use and interpretability. It considers the k nearest neighbours to classify an instance based on the majority class among the neighbours. In my project, I used KNN with k values of 1, 5, and 10 to explore its performance with varying neighbourhood sizes. However, KNN can suffer from the curse of dimensionality and might be computationally expensive for large datasets.

6.2 Support Vector Machine (SVM) with Linear Kernel

SVM is a powerful and versatile algorithm used for both classification and regression tasks. It aims to find the optimal hyperplane that maximizes the margin between classes. The linear kernel SVM is particularly useful for datasets with linearly separable data points. It offers robustness and has strong generalization capabilities. However, it may not perform well on non-linearly separable datasets, and its training time could be slow for large datasets.

6.3 Support Vector Machine (SVM) with Radial Basis Function (RBF) Kernel

The RBF kernel SVM is an extension of the linear SVM that can handle non-linearly separable data. The RBF kernel introduces a higher-dimensional feature space, allowing the model to find the optimal hyperplane even in complex datasets. It offers flexibility and can capture intricate patterns within the data. However, the RBF kernel SVM can be sensitive to hyperparameters, and its training time may be slow for large datasets.

6.4 Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. It offers several benefits, such as robustness against overfitting, handling missing data, and dealing with imbalanced datasets. Random Forest can capture complex relationships within the data and is relatively fast compared to other algorithms like SVM. However, the algorithm may become complex with a large number of trees, leading to increased memory usage and slower prediction times.

Results In this project, I used an evaluation function to compare the performance of the six machine-learning models on the test dataset. The function accepts trained model files, the testing dataset, and the corresponding test labels as input parameters. It predicts the test labels using the trained models and computes evaluation metrics such as error, accuracy, precision, recall, and F1 score for each model using the ”evaluate-score” function. The evaluation results for each model are as follows:

Model KNN-1 KNN-5 KNN-10 SVM-Linear SVM-RBF RandomForest Error 0.2252 0.2914 0.2504 0.2935 0.2833 0.1671 Accuracy 0.7748 0.7086 0.7496 0.7065 0.7065 0.8329 Precision 0.7070 0.6291 0.6269 0.7835 Recall 0.6940 0.5520 0.6265 0.7734 F1-Score 0.7005 0.5881 0.6267 0.7784

The evaluation metrics are consolidated into a dictionary containing the model name, error, accuracy, precision, recall, and F1 score. The metrics help us compare the performance of different models on the testing dataset and select the best-performing model. Based on the results, the RandomForest model has the lowest error (0.1671) and the highest accuracy (0.8329), precision (0.7835), recall (0.7734), and F1 score (0.7784) among all the models, indicating that it outperforms the other models in question similarity classification. These evaluation metrics provide valuable insights into the strengths and weaknesses of each model and guide us in selecting the most suitable model for my task.**Image – picture6.jpg**

7 conclusion

In this project, I developed a question similarity classification system using natural language processing and machine learning techniques. I used the Quora Question Pairs dataset to train and evaluate six different models, including K-Nearest Neighbors with k = 1, 5, and 10, Support Vector Machines with linear and RBF kernels, and Random Forest. Through a thorough data analysis, feature extraction, and evaluation process, I was able to identify the most suitable model for my task, which was the Random Forest model. The RandomForest classifier outperformed the other models in terms of error, accuracy, precision, recall, and F1 score, demonstrating its effectiveness in detecting duplicate questions. By developing such a system, I can improve the user experience on question-and-answer platforms like Quora, reduce content redundancy, and ensure that users find the information they seek efficiently.

8 Future Work

There are several potential directions for future work to further enhance the performance and capabilities of the question similarity classification system: **1.** Experiment with additional machine learning models, such as deep learning techniques like recurrent neural networks (RNNs), long short-term memory (LSTM) networks, or transformer models like BERT, to explore their effectiveness in capturing complex linguistic patterns and semantic relationships.

2. Investigate the impact of different feature selection methods to identify the most relevant features and reduce the feature space, potentially improving the performance of the models.

3. Explore the use of advanced text representation techniques like word embeddings (e.g., Word2Vec, GloVe) or sentence embeddings (e.g., Universal Sentence Encoder) to capture more nuanced semantic information in the questions.

4. Optimize the hyperparameters of the machine learning models using techniques like grid search or random search to further improve their performance.

5. Develop an online learning system that can adapt and update the models in real-time as new question pairs are added to the platform, ensuring that the question similarity classification system remains accurate and up-to-date.

By addressing these potential future directions, I can continue to improve the question similarity classification system, making it an even more valuable tool for question-and-answer platforms and their users.

9 References

1.Li, X., and Roth, D. (1970, January 1). Learning question classifiers. ACL Anthology. Retrieved January 29, 2023, from https://aclanthology.org/C02-1150/

2.Gollapalli, S.D. and Ng, S.K. (no date) QSTS: A question-sensitive text similarity measure for question generation, ACL Anthology. Available at: https:// aclanthology.org/2022.coling-1.337/ (Accessed: January 29, 2023).

3.Quora Question Pairs. (n.d.). Kaggle. Retrieved from https://www.kaggle.com/competitions/quora-question-pairs

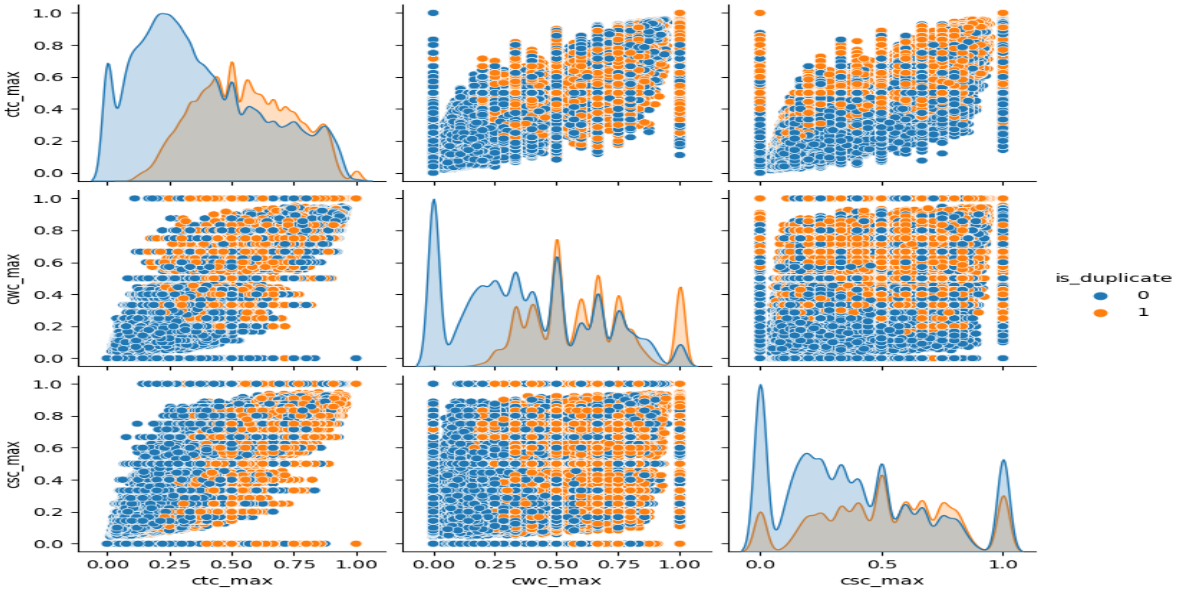

Figure 1: picture 1

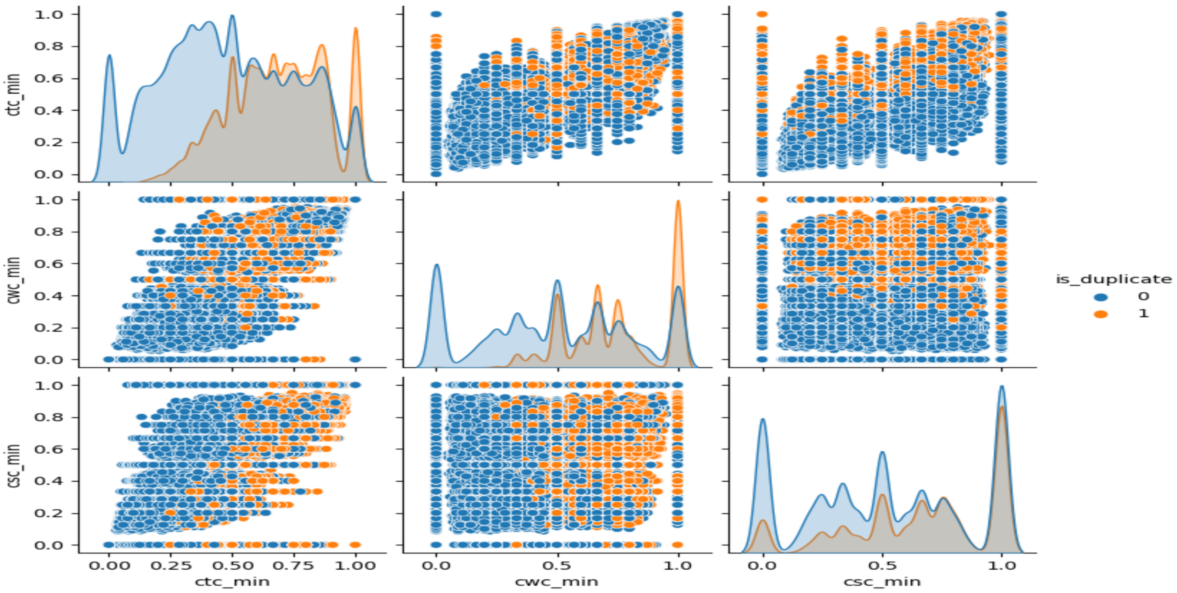

Figure 2: picture 2

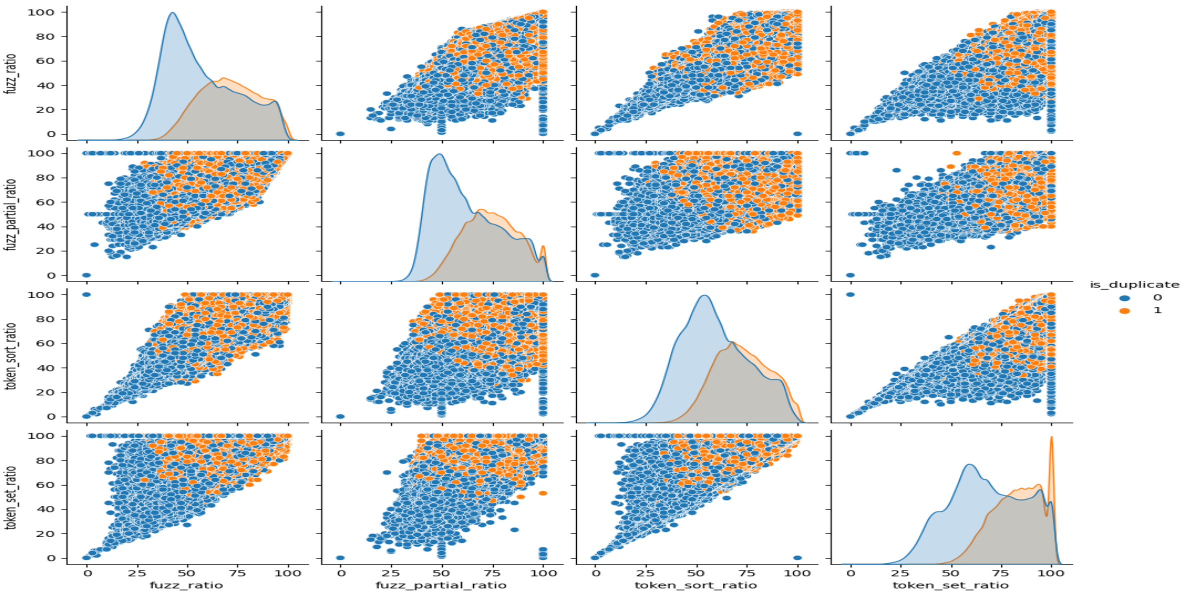

Figure 3: picture 3

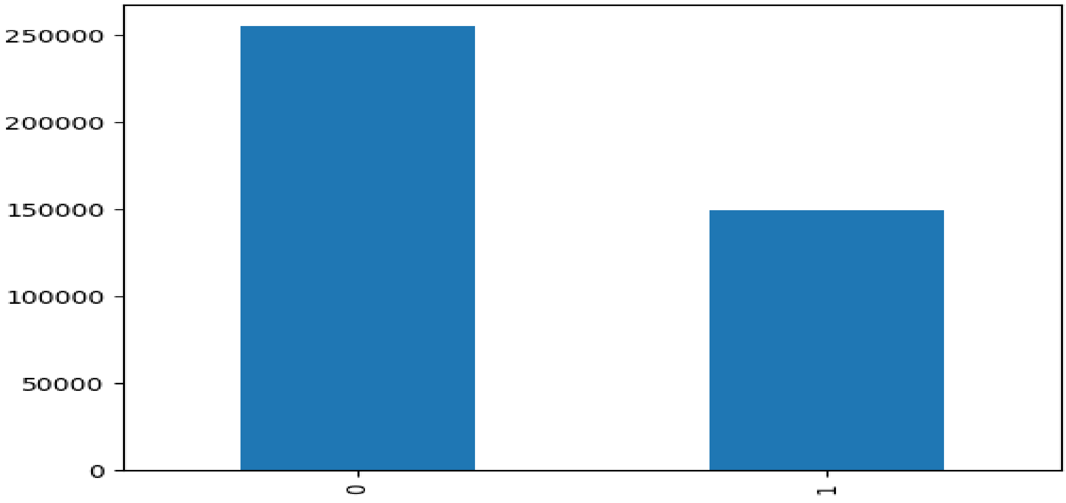

Figure 4: picture 4

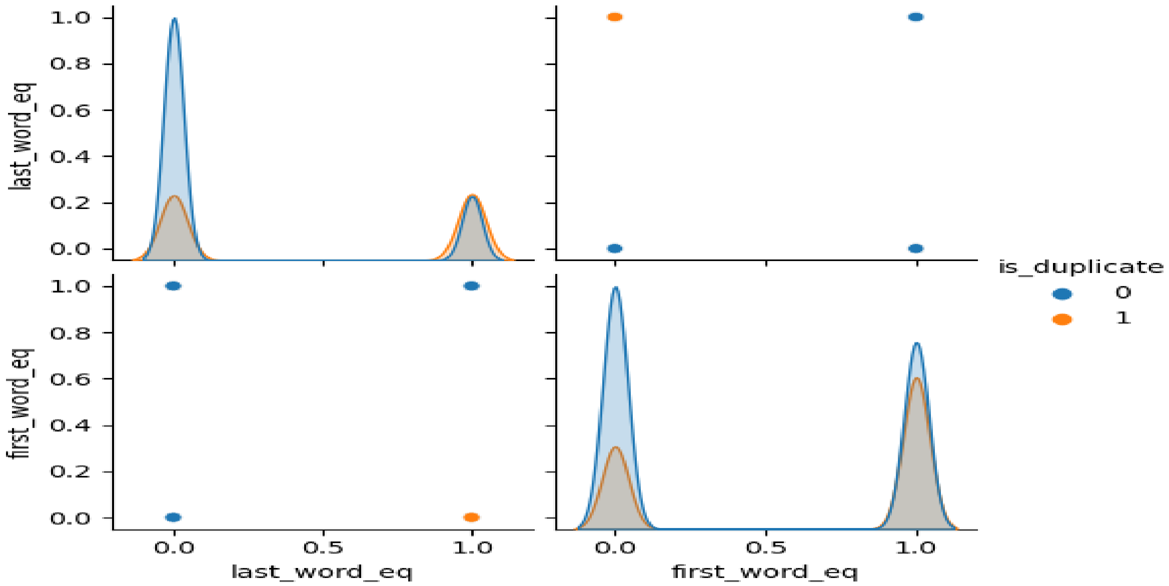

Figure 5: picture 5

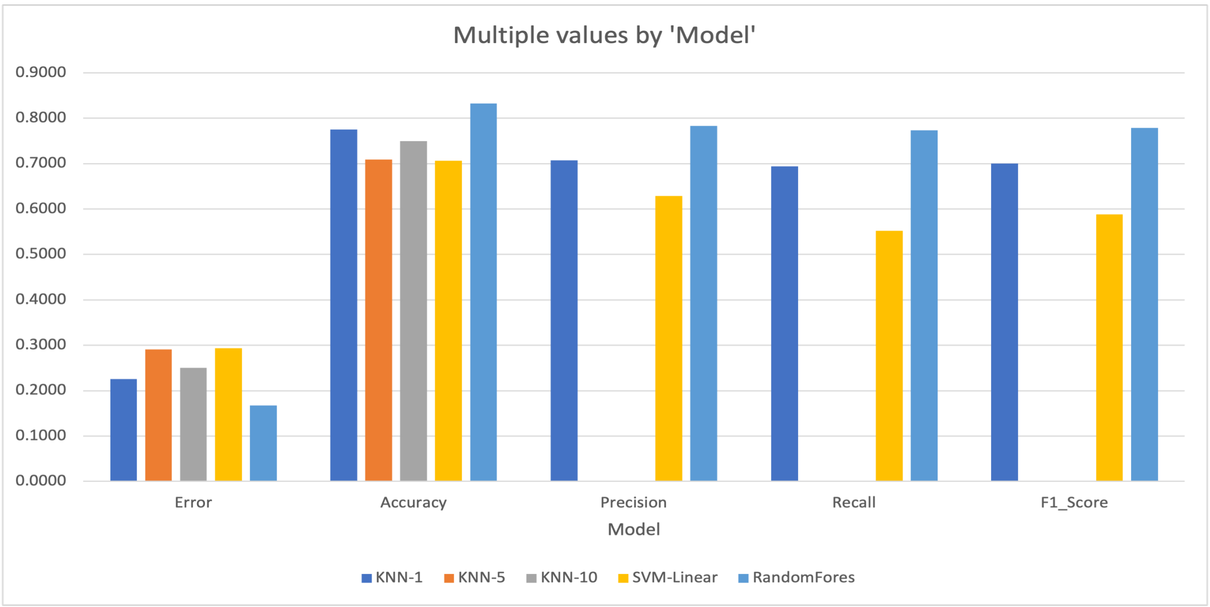

Figure 6: picture 6