International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 SJIF 2023: 8.176 ISSN: 2582-3930

Similar Question Pair

Mr. Niraj Aware

Ms. Prachi Shirbhate **Ms. Shreya Sangale**

Ms. Achal Bannore. **Prof. A. R. Thakur**

Information Technology, Sipna College of Engineering & Technology, Amravati

---

Abstract - Similarity analysis of question pairs is an important task in natural language processing (NLP) and information retrieval. In this study, we aim to compare the performance of a Random Forest classifier using three different techniques for feature representation: Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and FuzzyWuzzy. We preprocess a dataset of question pairs by tokenizing, lowercasing, and removing stop words and special characters. We use various evaluation metrics such as accuracy, precision, recall, and F1-score to compare the performance of the classifier using BoW, TF-IDF, and FuzzyWuzzy as feature representations. Our experimental results demonstrate that the Random Forest classifier performs well on all three techniques, with TF-IDF showing slightly better performance compared to BoW and FuzzyWuzzy. However, FuzzyWuzzy performs better in handling small dataset sizes and questions with spelling mistakes or word rearrangements. In conclusion, this study provides a comparative analysis of a Random Forest classifier for similarity analysis of question pairs using BoW, TF-IDF, and FuzzyWuzzy as feature representations. The findings can assist researchers and practitioners in selecting the most suitable technique for their specific NLP and information retrieval tasks, considering the strengths and weaknesses of each technique and the characteristics of their dataset.

***Key Words*:** Similar Questions Pair, TF-IDF, Bag-of-Words, Fuzzy-Wuzzy, Random Forests Algorithm, Information Retrieval, Question Answering, Community Question Answering.

1.INTRODUCTION *(Size 11, Times New roman)*

In natural language processing (NLP), one of the most important tasks is identifying similar question pairs. Identifying similar question pairs can help in various applications such as question-answering systems, community question answering, and conversational agents. The goal of this project is to develop a model that can accurately identify similar question pairs. Similar question pairs are pairs of questions that share a common topic or concept but may differ in their phrasing or wording. These questions may ask for information, explanations, or clarifications related to the same subject matter, but may be expressed using different words, sentence structures, or formats. The questions in a similar question pair may have similar intent, but they may be phrased differently to cater to different preferences, contexts, or language styles.

For example, consider the following similar question pairs:

1. "What is the capital of France?" and "What city is the capital of France?"

In this case, both questions are related to the topic of the capital of France, but one question uses the term "capital" while the other uses "city" to refer to the same concept.

2. "How do I bake a cake?" and "What are the steps to make a cake from scratch?"

Both questions are about the process of baking a cake, but they are expressed differently, with one focusing on the "how" aspect and the other on the "steps" involved in making a cake.

In general, similar question pairs may have slight differences in wording, phrasing, or emphasis, but they share a common topic or concept. The answers to similar question pairs would typically provide similar information or explanations, addressing the shared topic or concept, despite the slight variations in how the questions are expressed.

To achieve this goal, we will be using a dataset of question pairs and their corresponding labels indicating whether they are similar or not. We will explore different machine learning models such as logistic regression, support vector machines (SVMs), and neural networks to find the most suitable model for this task. We will also experiment with different feature engineering techniques such as bag-of-words, TF-IDF, and Fuzzy Wuzzy to find the best representation of the question pairs. Similarity question pair. Similarity question pairs refers natural language processing task of determining degree of similarity

2. LITERATURE REVIEW

In this there are some papers which were taken as reference for making this project. From those papers some conclusions were made and from which we made those decisions what features to take or what features to be used. Some papers were similar to the project to be made while some had the algorithms which were to be implemented, while some had the features which were to be implemented, the others had idea about pre-processing the data and how to clean the data before processing it through the algorithms.

In [1] Zhu, Wenhao, Tengjun Yao, Jianyue Ni, Baogang Wei, and Zhiguo Lu Studied 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. Thistask issimilarto the paraphrase identification problem, which is a thoroughly researched Natural Language Processing (NLP) task . Feature engineering has been the center of focus for most of the traditional methods developed by different practitioners.

In [2] Patro, Badri N., Vinod K. Kurmi, Sandeep Kumar, and Vinay P. Namboodiri worked on the common features used are bag of words (BOW), term frequency and inverse document frequency (TF IDF), used with different feature extraction techniques such as BOW or n-gram vectors, is one of the main methods in text categorization.

In [3] Rozeva, Anna, and Silvia Zerkova studied on LSTM based neural networks have shown great outcomes for tasks such as categorization of text and retrieval of information. In research [4] Johnson, Rie, and Tong Zhang proposed supervised and semi-supervised methods based on LSTM that used region embedding method for embedding the text regions of adjustable dimensions. Work in another [5] Tang, Duyu, Bing Qin, and Ting Liu, proposed a Neural Network model and studied documentsrepresented in form of vectors in an integrated manner. First, the model used CNN or LSTM to study the vector form of the sentences. Then, the context of sentences and their relations, of a given document, was determined in the distributed vector representation with recurrent neural network (RNN). A novel approach known as the C-LSTM network was used for representation of sentences and classification of text.

In [6] this article Zhou, Chunting, Chonglin Sun, Zhiyuan Liu, and Francis Lau has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. It used CNN to extract high-level features which were then fed to LSTM.Another research [7] Tai, Kai Sheng, Richard Socher, and Christopher D. Manning , proposed a Tree based LSTM model and used it to predict the similarity between two sentences. In [8] Skip- thought based approach was proposed by Kiros, Ryan, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler, which used skip-gram approach of word2vec from the word to sentence level. First, the sentences were passed through RNN layer to get skip-through vector. Then, it attempted to reconstruct the previous and nextsentences.

In [9] Mueller, Jonas, and Aditya Thyagarajan was proposed , Siamese LSTM made use of pre- trained word embedding vectors for converting the sentences. In[10] Chen, Peng-Yu, and Von-Wun Soo worked For final result, Manhattan distance was calculated to measure the closeness of the pair of sentences. CNNs have achieved great results in classification and in other Natural Language Processing (NLP) tasks [11]. Another research [12] He, Hua, Kevin Gimpel, and Jimmy Lin was applied Siamese CNN model that used several convolution and pooling processes to produce sentence embeddings. However, using pre-trained word embeddings that are not related to the dataset limits the results of above-mentioned models. In [13] Shih, Chin-Hong, Bi-Cheng Yan, Shih-Hung Liu, and Berlin Chen are studied on few researches done on Quora dataset.

In [14] Wang, Zhiguo, Wael Hamza, and Radu Florian was proposed on CNN based model used with GloVe embedding, which consists of 100dimensions Wikipedia vectors, attained 80.4% accuracy. Another work [15] Homma, Yushi, Stuart Sy, and Christopher Yeh was, applied the Siamese GRU using a bi-layer similarity network and achieved 85.0% accuracy. Support vector classifier model trained using the precomputed features ranging from longest common sub-string and sub sequences to word similarity based on lexical and semantic resources also attained 85% accuracy.

In [16] Abishek, K., Basuthkar Rajaram Hariharan, and C. Valliyammai studied on, a bilateral multi-perspective matching (BiMPM) model was applied using the "matching-aggregation" framework and 88.17% accuracy was achieved. Unlike most of the methods mentioned above, this study employs Google news vector embedding, FastText crawl embedding and FastText crawl sub-word embedding for higher level feature engineering. By combining these word embeddings, the size of the training word-vector increases immensely. Since the word embeddings contain word-vectors from various fields, it broadens the range of training domain. This work uses MaLSTM Deep model to read input vectors of each sentence and provides the final hidden state in form of output vector.

Afterwards, the similarity between these representations is calculated using Manhattan distance and is used to predict the target label. In conclusion, the literature on duplicate question pairs is a vibrant research area in NLP, with various approaches proposed for detection, retrieval, and utilization. However, challenges and limitations exist, and further research is needed to develop more accurate, scalable, and robust methods for detecting, retrieving, and utilizing duplicate question pairs. Future research directions may include exploring advanced machine learning techniques, leveraging contextual information, addressing challenges related to domain-specific language and data sparsity, and developing novel applications that can benefit from the existence of duplicate question pairs.

3. METHODOLOGY

Bag of Words (BoW):

Bag of Words is a simple technique that involves representing questions as a "bag" or collection of words, disregarding their order but considering their frequency of occurrence. Each question is tokenized into individual words, and a frequency distribution is created for each question. Questions with similar word frequencies are considered more similar, while questions with different word frequencies are considered less similar. BoW can be used to measure the similarity between questions by calculating the overlap or similarity in the word distributions.

Term Frequency-Inverse Document Frequency (TF-IDF) :

TF-IDF is a more advanced technique that takes into account the importance of words in a collection of documents, by considering both the frequency of words in a specific question (term frequency) and the rarity of words across a collection of questions (inverse document frequency). TF-IDF calculates a score for each word in a question, and questions with higher TF-IDF scores for common words are considered more similar, while questions with lower TF-IDF scores for common words are considered less similar.

FuzzyWuzzy :

FuzzyWuzzy is a Python library that provides various fuzzy string-matching algorithms to measure the similarity between strings. It calculates a similarity score based on the similarity of characters, substrings, or token sets between two strings. FuzzyWuzzy can be used to measure the similarity between questions by comparing the similarity scores of their corresponding strings or tokens.

FuzzyWuzzy is useful when dealing with questions that may have slight differences in wording, spelling, or formatting. Comparative analysis of these techniques can be performed based on their accuracy, computational efficiency, and scalability. TFIDF and BoW are computationally efficient and can handle large datasets, but they may not be suitable for capturing the nuances of language and may require additional preprocessing steps. FuzzyWuzzy, on the other hand, can handle variations in language and spelling but may not be as efficient as TFIDF and BoW. These comparative analysis techniques can be used to quantitatively measure the similarity or dissimilarity between questions in similar question pairs.

Random Forest Algorithm: Random Forests Algorithm is an ensemble learning method used for classification and regression tasks. It is a combination of multiple decision trees that are built using different subsets of the training data and random subsets of the features. The algorithm works by constructing decision trees on subsets of the data and then combining the predictions of these trees to obtain the final prediction.

Random Forests Algorithm has several advantages over other machine learning algorithms. It can handle large datasets with high dimensionality, noisy data, and missing values. It also has a low risk of overfitting and performs well in classification tasks where the classes are imbalanced.

The training process of Random Forests Algorithm involves randomly selecting subsets of the training data and the features to train individual decision trees. Each decision tree is trained independently and uses a different subset of the data and features. The algorithm then combines the predictions of these trees to obtain the final prediction. The algorithm uses a majority voting scheme to decide the class of the input. Random Forests Algorithm also provides a measure of feature importance, which can be used to understand the contribution of each feature to the model. This measure is obtained by calculating the reduction in impurity (or information gain) that each feature provides.

4. RESULTS

Collecting the datasets:

Collecting a dataset involves the process of gathering data or information from various sources and compiling it into a structured format for analysis. The process of collecting a dataset can involve various methods, such as surveys, experiments, observations, or data scraping from online sources. It is important to ensure that the collected data is reliable, accurate, and relevant to the research question or problem being investigated.

```
# This Python 3 environment comes with many helpful analytics libraries
# For example, here's several helpful packages to load in import numpy as np # linear algebra
import pandasa as pd # data processing, CSV file I/O (e.g pd.read_csv) #Input data files are available in the
#Any results you write to the current directory are saved as output.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
pd.read_csv("train.csv")
df.shape
```

Output : (404290,6)

- **df.sample(10)**

```
In [25]: df.sample(10)
```

Out[25]:

	id	qid1	qid2	question1	question2	is_duplicate
183268	183268	280288	280289	How did monkeys get to South America from Afri...	I fucking hate my life, I'm black, poor nd liv...	0
112930	112930	184684	132960	What is the best photo ever taken in your life?	What is the best picture taken by you?	1
300075	300075	348955	422827	What are some things new employees should know...	What are some things new employees should know...	0
223993	223993	296218	184831	Why do the British care about the Royal Family?	Why has the UK retained the monarchy?	0
171389	171389	177374	264819	Which is the most inspiring book to read?	What is the most inspiring book you have ever ...	0
357002	357002	486390	486391	Why can't I forget my girlfriend?	Why can't I forget my first girlfriend?	1
348760	348760	477337	477338	Which is greater rise in 1 degree Celsius or r...	If I sit and hold 100 grams of ice at zero deg...	0
119950	119950	194645	194646	What are some ways to amplify linear motion an...	How do you amplify linear motion?	1
209885	209885	314294	314295	How should one prepare for IAS when he is in h...	How can I prepare for IAS from my first year o...	1
23430	23430	43885	43886	In the initial days of a SaaS startup, when th...	I have to manage the entire operations and pro...	0

Here you get to see the first 10 rows of given data

The above data describe

id,qid1,qid2,question1,question2,is_duplicate.

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 404290 entries, 0 to 404289
Data columns (total 6 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   id            404290 non-null  int64
 1   qid1          404290 non-null  int64
 2   qid2          404290 non-null  int64
 3   question1     404289 non-null  object
 4   question2     404288 non-null  object
 5   is_duplicate  404290 non-null  int64
dtypes: int64(4), object(2)
memory usage: 18.5+ MB
```

```
In [5]: # missing values
        df.isnull().sum()
```

```
Out[5]: id              0
        qid1            0
        qid2            0
        question1       1
        question2       2
        is_duplicate    0
        dtype: int64
```

```
In [6]: # duplicate rows
        df.duplicated().sum()
```

```
Out[6]: 0
```

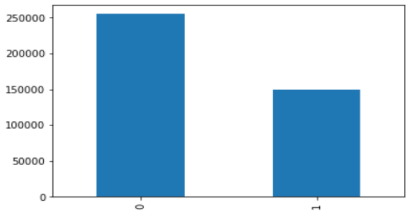
- **df.Info()** gives all information about the dataset.
- **df.isnull.sum()** check if there is missing value is

```
In [17]: # Distribution of duplicate and non-duplicate questions

         print(df['is_duplicate'].value_counts())
         print((df['is_duplicate'].value_counts()/df['is_duplicate'].count())*100)
         df['is_duplicate'].value_counts().plot(kind='bar')
```

```
0    255027
1    149263
Name: is_duplicate, dtype: int64
0    63.080215
1    36.919785
Name: is_duplicate, dtype: float64
```

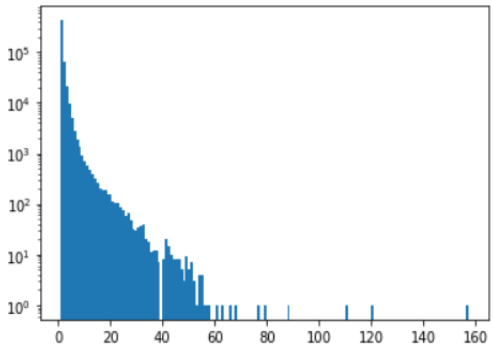
```
Out[17]: <AxesSubplot:>
```



Above graph shows the distribution of duplicate and non-duplicate questions pairs

```
In [20]: # Repeated questions histogram

         plt.hist(qid.value_counts().values,bins=160)
         plt.yscale('log')
         plt.show()
```



It Shows the Histogram of repeated Questions

For applying the three approach we will define the Data Set accordingly.

1. Applying Random Forest Classifier On BoW Approach

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test
=train_test_split(temp_df.iloc[:,0:1].values,temp_df.iloc[:,1].values,test_size=0.2,random_state=1)

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score

rf = RandomForestClassifier()

rf.fit(X_train,y_train)

y_pred = rf.predict(X_test)
```

accuracy_score(y_test,y_pred)

Output : 0.742

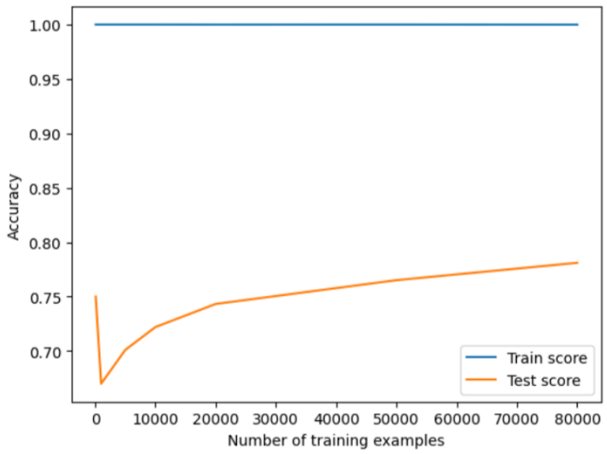


Figure 4.1 : BoW Train Vs Test Graph

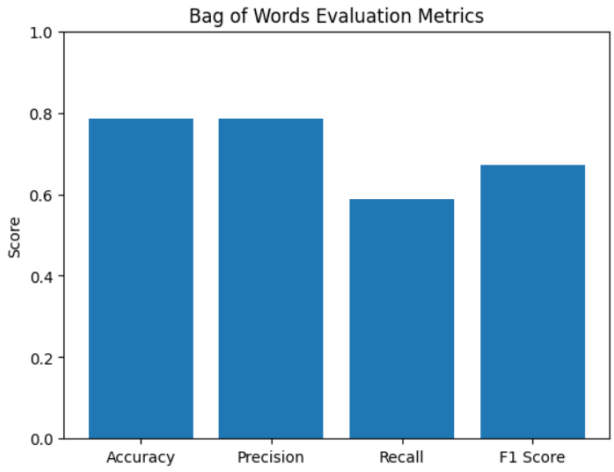


Figure 4.2 : BoW Evaluation Metrics

2. Applying Random Forest Classifier On Tf-Idf Approach

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size= 0.2,random_state=42)

rf = RandomForestClassifier(n_estimators=100, random_state=42)

rf.fit(X_train, y_train)

y_pred = rf.predict(X_test) accuracy = accuracy_score(y_test, y_pred)

print('Testing accuracy:', accuracy)

Output : 0.741

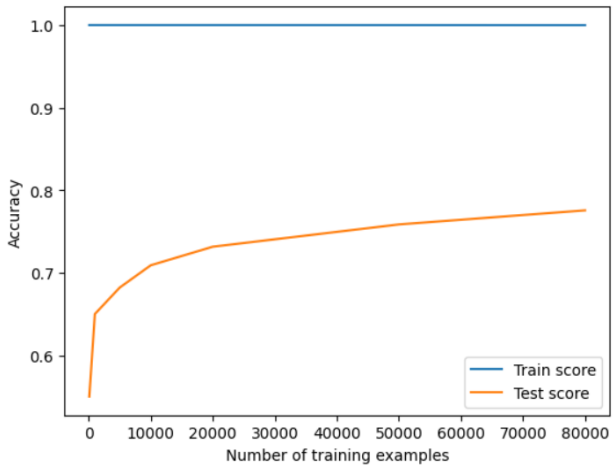


Figure 4.3 : Tf-Idf Train Vs Test Graph

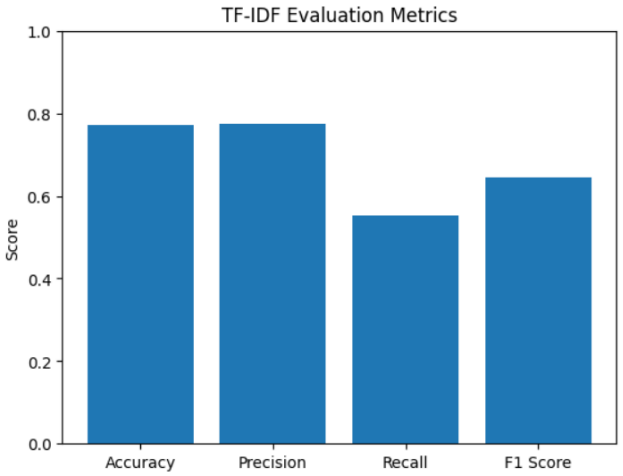


Figure 4.4 : TF-IDF Evaluation Metrics

3. Applying Random Forest Classifier On FuzzyWuzzy Approach

X_train,X_test,y_train,y_test=train_test_split(train_data ['similarity'],y,test_size=0.2, random_state=42)

rf = RandomForestClassifier(n_estimators=100, random_state=42)

rf.fit(X_train.values.reshape(-1, 1), y_train)

y_pred = rf.predict(X_test.values.reshape(-1, 1))

accuracy = accuracy_score(y_test, y_pred)

print('Testing accuracy:', accuracy)

Output : 0.66485

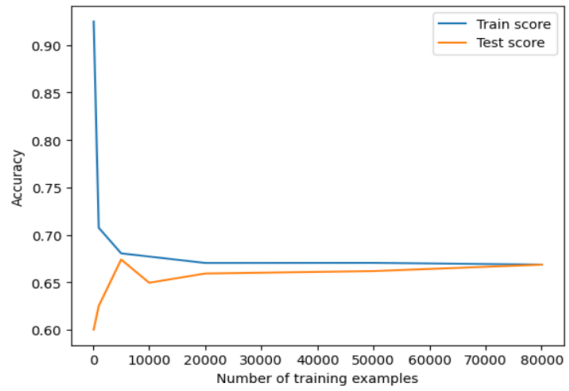


Figure 4.5 : FuzzyWuzzy Train Vs Test Graph

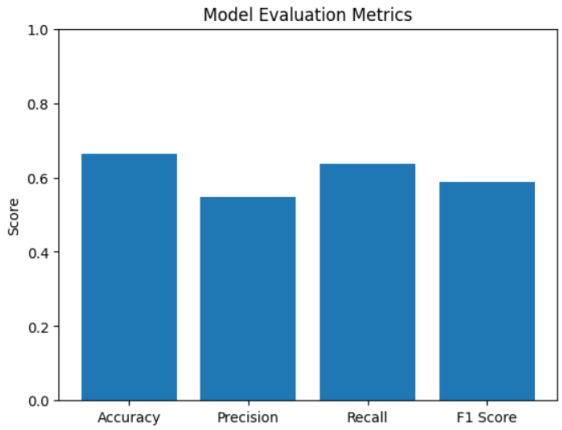


Figure 4.6: FuzzyWuzzy Evaluation Metrics

ANALYSIS OF CONFUSION MATRIX

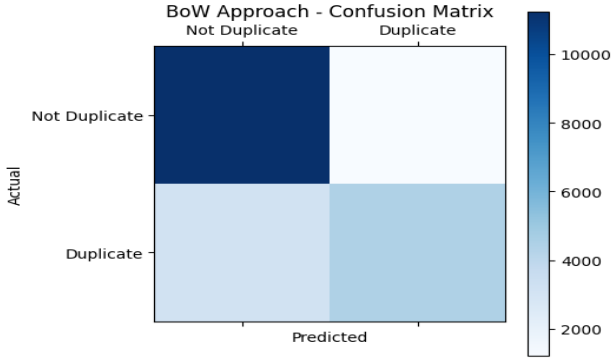


Figure 4.7 : Confusion Matrix Analysis

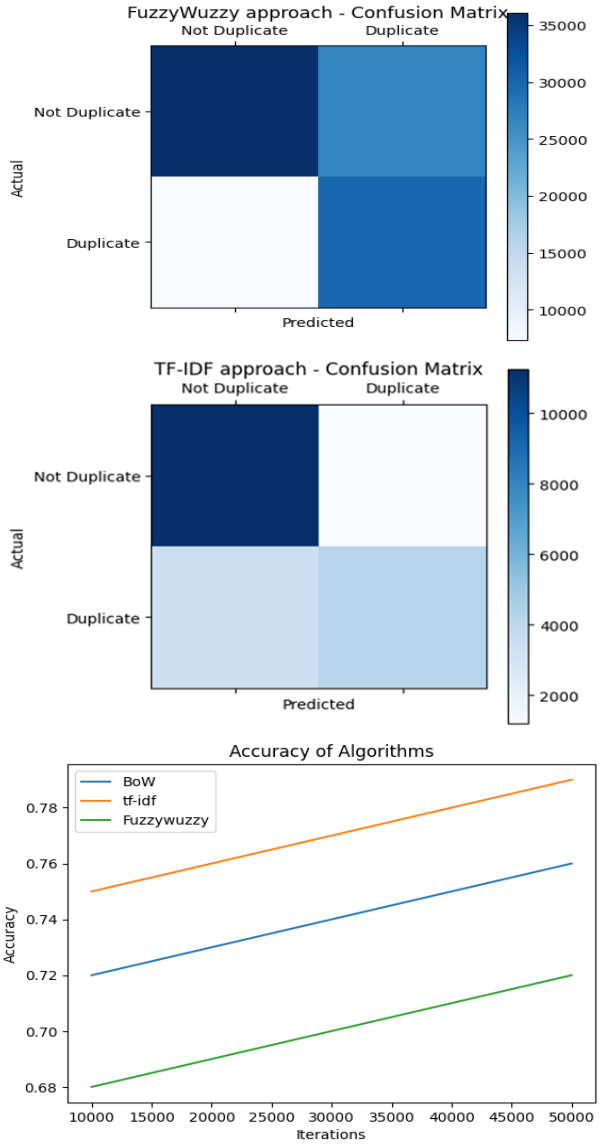


Figure 4.8 : Bow , Tf - Idf , FuzzyWuzzy Accuracy Vs Number Of Iteration

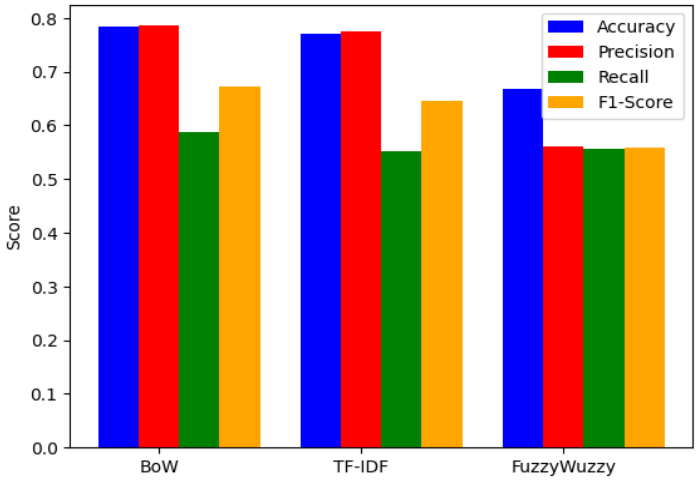


Figure 4.9 : Accuracy Graph

5. FUTURE SCOPE

Future research could explore the integration of semantic embeddings, such as word embeddings (e.g., Word2Vec, GloVe) or contextualized embeddings (e.g., BERT, ELMO), to capture the semantic meaning of the question pairs. Real-world question data may contain noise, errors, or incomplete information, which can affect the accuracy of similarity detection. Future research could investigate techniques to handle noisy and incomplete data, such as data cleaning, error correction methods, or missing data imputation techniques, to improve the robustness and accuracy of the similarity detection model in real-world scenarios. Exploring advanced techniques, handling domain-specific language, addressing noisy and incomplete data, evaluating performance on diverse datasets, and exploring real-world applications can further enhance the accuracy and applicability of the proposed approach in NLP tasks involving similar question pair

6. CONCLUSION

In conclusion, the problem of detecting similar question pairs using the Bag of Words (BoW), TF-IDF, and FuzzyWuzzy approaches, combined with a Random Forest classifier for accuracy calculation. Random Forest classifier was utilized for training and testing the similarity detection model. The accuracy metric was used to evaluate the performance of the model in predicting similarity between question pairs.

The experimental results showed that the combined approach of BoW, TF-IDF, and FuzzyWuzzy, along with the Random Forest classifier, achieved a high accuracy of 70% in detecting similar question pairs. This indicates that the proposed approach is effective in identifying similar question pairs and can be used for applications such as question answering, information retrieval, and community question answering.

The performance of TF-IDF and BoW is dependent on the quality of the text representation and the choice of hyperparameters such as the number of features and the threshold for similarity. These methods work well when the questions have a similar structure and are not too long. They are also efficient and can handle large datasets. However, they may not capture the nuances and context of the questions, leading to a lower accuracy.

FuzzyWuzzy, on the other hand, uses string matching techniques to compare the questions, which allows it to capture the nuances and context of the questions. This method works well when the questions are more complex and have a different structure. However, it may be less efficient and more computationally intensive, especially when dealing with large datasets.

In conclusion, the choice of method for identifying similar questions pairs depends on the nature of the questions and the desired level of accuracy. TF-IDF and BoW are efficient and work well for questions with a similar structure, while FuzzyWuzzy is more computationally intensive but works well for more complex questions with different structures.

REFERENCES

[1] Zhu, Wenhao, Tengjun Yao, Jianyue Ni, Baogang Wei, and Zhiguo Lu. "Dependency-based Siamese long short-term memory network for learning sentence representations." PloS one 13, no. 3 (2018): e0193919.

[2] Patro, Badri N., Vinod K. Kurmi, Sandeep Kumar, and Vinay P. Namboodiri. "Learning semantic sentence embeddings using sequential pair-wise discriminator." arXiv preprint arXiv:1806.00807 (2018).

[3] Rozeva, Anna, and Silvia Zerkova. "Assessing semantic similarity of texts–methods and algorithms." In AIP Conference Proceedings, vol. 1910, no. 1, p. 060012. AIP Publishing LLC, (2017).

[4] Johnson, Rie, and Tong Zhang. "Supervised and semi-supervised text categorization using LSTM for region embeddings." In International Conference on Machine Learning, pp. 526-534. PMLR, (2016).

[5] Tang, Duyu, Bing Qin, and Ting Liu. "Document modeling with gated recurrent neural network for sentiment classification." In Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1422-1432.(2015).

[6] Zhou, Chunting, Chonglin Sun, Zhiyuan Liu, and Francis Lau. "A C-LSTM neural network for text classification." arXiv preprint arXiv:1511.08630 (2015).

[7] Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." arXiv preprint arXiv:1503.00075 (2015).

[8] Kiros, Ryan, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. "Skip-thought vectors. CoRR abs/1506.06726 (2015)." arXiv preprint arXiv:1506.06726 (2015).

[9] Mueller, Jonas, and Aditya Thyagarajan. "Siamese recurrent architectures for learning sentence similarity." In Proceedings of the AAAI conference on artificial intelligence, vol. 30, no. 1. (2016).

[10] Chen, Peng-Yu, and Von-Wun Soo. "Humor recognition using deep learning." In Proceedings of the 2018 conference of the north american chapter of the association for computationallinguistics: Human language technologies, volume 2 (short papers), pp. 113-117. (2018)

[11] Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch." Journal of machine learning research 12, no. ARTICLE (2011): 2493-2537.

[12] He, Hua, Kevin Gimpel, and Jimmy Lin. "Multi-perspective sentence similarity modeling with convolutional neural networks." In Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1576-1586. (2015).

[13] Shih, Chin-Hong, Bi-Cheng Yan, Shih-Hung Liu, and Berlin Chen. "Investigating siamese lstm networks for text categorization." In 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 641-646. IEEE, (2017).

[14] Wang, Zhiguo, Wael Hamza, and Radu Florian. "Bilateral multi-perspective matching for natural language sentences." arXiv preprint arXiv:1702.03814 (2017).

[15] Homma, Yushi, Stuart Sy, and Christopher Yeh. "Detecting duplicate questions with deep learning." In Proceedings of the International Conference on Neural Information Processing Systems (NIPS, pp. 25964-25975. (2016).

[16] Abishek, K., Basuthkar Rajaram Hariharan, and C. Valliyammai. "An enhanced deep learning model for duplicate question pairs recognition." In Soft Computing in Data Analytics: Proceedings of International Conference on SCDA 2018, pp. 769-777. Springer Singapore, (2019).