International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Exploring NLP Techniques for Duplicate Question Detection to Maximizing Responses on Q&A Websites

Dr. Nilesh B. Korade[1], Dr. Mahendra B. Salunke[2], Gayatri G. Asalkar[3], Rutuja G. Khedkar[4], Ashwini U. Bhosale[5], Dhanashri M. Joshi[6], Amol C. Jadhav[7]

Submitted: 27/01/2024 **Revised**: 05/03/2024 **Accepted**: 13/03/2024

Abstract: Emerging technologies known as Question Answering Systems (QAS) offer accurate and precise responses to common questions. Duplicate Question Detection (DQD) has demonstrated its capacity to enhance the user experience and drastically decrease response time by utilizing past responses. Word choice and sentence construction might differ significantly, making it difficult to determine if the two questions are asking the same thing. Finding questions on question-and-answer sites such as Quora, Stack Overflow, Blurtit, etc. that are semantically identical is very important to make sure that users receive both high-quality and high-quantity content according to the question's purpose, improving the user experience entirely. Quora's dataset of four lacks labelled question pairs used in the presented research. Our research has involved the construction of new features and the demonstration of their ability to improve accuracy. The study examines various vectorization methods and how they affect accuracy, with Word2Vec proving to be a good performer among the methods. In order to identify duplicate questions in the question pair dataset, we explored and used various machine learning and deep learning techniques. The cascaded CNN outperforms other modern algorithms and offers outstanding value over all assessment metrics.

Keywords: *Duplicate Question Detection, Feature engineering, Vectorization, Word2Vec, Cascaded CNN.*

1. Introduction

Users can pose questions on question-and-answer online platforms, and other users can respond to them. A lot of the questions that are asked at any given moment have previously been asked by other users, usually in a different format [1]. It would be ideal to combine these redundant questions into a single canonical question because doing so would have the following advantages:

- It can be irritating for users looking for original answers when there are too many duplicate queries on the platform. By identifying and eliminating duplicates, users can quickly locate pertinent information without having to explore through repetitive content [2].
- If a question has already been addressed on the website, the person asking the inquiry saves time. They don't have to wait too long to hear back; they can get an answer instantaneously [3].
- Instead of repeatedly responding to the same question, users can concentrate on answering new and unique questions, which enhances the quality of their responses and enables the effective use of community resources [4].
- When users see that their contributions are recognized and the platform is well-structured, they are more inclined to engage with a Q&A site [5].

To increase the effectiveness of resource usage over the internet, it is imperative to discover such repeated questions. It is not feasible to manually discover and then eliminate duplicate questions. Using some autodetection techniques, the duplicate questions are expected to be recognized automatically [6].

Using Quora's dataset, we have developed many features based on factors like question length, string occurrences, common strings on both questions, fuzzy logic, etc. A different algorithm was trained to compare the old dataset with the recently added feature set, and the evaluation was conducted using several metrics [7]. Several vectorization techniques are used to convert text datasets to numerical features; the results indicate that Word2Vec performs well in comparison to other methods. Cascaded CNN

[1]*Assistant Professor, Department of Computer Engineering, JSPM's Rajarshi Shahu College of Engineering, Tathawade, Pune – 411033, Maharashtra, India, nilesh.korade.ml@gmail.com.*
[2]*Assistant Professor, Department of Computer Engineering, PCET's, Pimpri Chinchwad College of Engineering and Research, Ravet, Pune-412101, Maharashtra, India, mahendra.salunke@pccoer.in.*
[3]*Research Scholar, Department of Computer Science and Engineering, Shri Jagdishprasad Jhabarmal Tibrewala University, Vidyanagari, Churela-333001, Rajasthan, India, gayatri.teke@gmail.com.*
[4]*Assistant Professor, Department of Computer Engineering, JSPM's, Rajarshi Shahu College of Engineering, Tathawade, Pune – 411033, Maharashtra, India, rgkhedkar_comp@jspmrscoe.edu.in.*
[5]*Assistant Professor, Department of Computer Engineering, JSPM's, Rajarshi Shahu College of Engineering, Tathawade, Pune – 411033, Maharashtra, India, aubhosale_comp@jspmrscoe.edu.in.*
[6]*Assistant Professor, Department of Computer Engineering, JSPM's, Rajarshi Shahu College of Engineering, Tathawade, Pune – 411033, Maharashtra, India, jdhanashrim@gmail.com.*
[7]*Assistant Professor, Department of Computer Engineering, JSPM's, Rajarshi Shahu College of Engineering, Tathawade, Pune – 411033, Maharashtra, India, acjadhav_comp@jspmrscoe.edu.in.*
** Corresponding Author Email: nilesh.korade.ml@gmail.com*

outperforms other algorithms in a variety of machine learning and deep learning algorithms that were trained on a newly constructed feature dataset. Assessment standards, including accuracy, recall, f1 score, and precision, are used to assess how effectively applied strategies perform [8].

The rest of the content of the document is organized as follows: The literature on duplicate question detection is addressed in Section II. The methodology is covered in Section III and includes the research flow, dataset used, preprocessing performed, feature development, approaches for vectorization, and algorithm employed. Accuracy with and without new features, accuracy after applying various vectorization approaches, and machine learning and deep learning strategies are all covered in Section IV. In Section V, we provide a summary of our research and offer recommendations for further exploration.

2. Literature Survey

In today's world, finding solutions to questions is simple because of well-known community-based services like Answerbag or Quora. These websites manage a Q&A database with millions of queries and responses. Even though there are particular guidelines instructing users to browse for answers before posting their own, users are still posting duplicate queries at a higher rate [9].

Research into building an automated system for detecting duplicate questions has been ongoing for many years. Machine learning techniques, including the Adaboost classifier, decision tree, decision tree with bagging, and random forest, can be used to construct the DQD system. After preprocessing the input dataset, vectorization is performed using term frequency and inverse document frequency (TF-IDF) to extract the importance of each term or word in the given document. A different machine learning model is then trained using the vectorized array, and the results indicate that AdaBoost provides higher accuracy than the other models. Adaboost yields an accuracy of 81.73%, while logistic regression, decision trees, and decision trees with bagging random forest yield, respectively, 79.21%, 79.29%, and 81.70% accuracy [10]. Depending on the queries posed, a search query in online applications may or may not get the expected results. The user might not feel confident seeing so many queries that might not even be related to his query. The Word Embeddings technique captures the inter-word semantics found in queries posted on Stack Overflow by representing individual words as real-valued vectors in a lower-dimensional space. J. Babu et al. used the cosine similarity method to determine the similarity between two sentences and rank the queries. For every sentence, the mean of the word embeddings is determined, and the similarity among questions is identified. Finally, methods such as discounted cumulative gain (DCG) and k score are examined to determine the ranking [11].

A well-known community-based question-answer (CQA) website primarily focused on software engineering; Stack Overflow has seen an increase in visitors in recent years. On Stack Overflow, duplicate questions frequently appear and are manually flagged by highly reputable members. Users with a good reputation can save time and effort by using automatic duplicate question detection. Existing methods for automatically identifying duplicate questions extract textual data, but they have limitations as semantic information may be lost. In order to address this issue, L. Wang et al. investigate the application of effective deep learning methods to identify duplicate questions on Stack Overflow. The vector representations of words are obtained using Word2Vec, which is capable of fully capturing semantic information at the word and document levels, respectively. To detect similarities between two questions, deep learning models based on Word2Vec, such as WV-RNN, WV-CNN, and WV-LSTM, were implemented. The evaluation's findings demonstrate that WV-CNN and WV-LSTM have significantly outperformed previous baseline techniques [12].

It would be quicker to locate good responses and save time if duplicate questions could be identified effectively. This would enhance the Quora user experience for both writers and viewers. Z. Imtiaz et al. used the Manhattan distance LSTM neural network model (MaLSTM) to predict duplicate questions in the dataset. To vectorize every question and train the model, three different word embeddings were applied independently to the question dataset. The embedding approaches were Google News embeddings, FastText crawls, and FastText crawls with subwords as the embedding methods. MaLSTM model, which takes into consideration the Manhattan distance to assess how similar the questions are semantically. The Manhattan score, where the non-duplicate pair values are substantially closer to zero and the duplicate question score is extremely close to 1, classifies the question pairs more accurately than any other embedding. According to the experiments, the suggested model outperforms the state-of-the-art model with an accuracy of 91.14%. [13].

In business intelligence applications and recommender systems, sentiment analysis could be useful for expeditiously summarizing user input and comments. The faster growth of digital social media serves as an opportunity for individuals to express their thoughts and feelings through text messages. The use of natural language processing to determine or categorize whether a text message's stated opinion is favorable or adverse is referred to as opinion mining. The study by G. Vinodhini et al. uses binary classification to divide text sentiment into evaluations that are favorable or unfavorable. Back propagation neural networks (BPNs) are used as classifiers in the research, and principal components are extracted using principal component analysis (PCA) in order to use

them as predictors. Receiver Operating Characteristics (ROC) analysis was performed to evaluate the performance of PCA-BPN with BPN without PCA. The outcome demonstrates the efficiency of BPN with PCA as a feature reduction technique for text sentiment categorization [14].

Software systems frequently have lots of customers in practice, enabling the generation of contradictory or redundant requirements as well as duplicate defects by various users. By automatically finding conflicts, redundancies, and neutral texts, software text analysis may save an extensive amount of time and work. Software systems frequently have lots of customers in practice, enabling the generation of contradictory or redundant requirements as well as duplicate defects by various users. By automatically finding conflicts, redundancies, and neutral texts, software text analysis may save an extensive amount of time and work. NLP-based techniques, including shuffling, reverse translation, paraphrasing, target-lemma substitution, synonym replacement for actors and actions using word embedding, and synonym replacement for nouns and verbs, are used by G. Malik et al. for data augmentation. In order to feed the BERT tokenizer, the augmented instances are added to the training set. For indicating the start and end of the input texts, the BERT tokenizer includes unique tokens. The class probabilities for the input instance are then obtained by passing the final representation through a softmax function [15].

Due to the widespread use of the Internet and the quick advancement of network technology, small blogs and e-commerce platforms are becoming ever more crucial to people's daily lives, education, and communication. These information pieces are typically small and have an unclear structure of grammar, but they also reveal the users' deep emotional tendencies. In order to effectively detect the semantic aspects and possible emotional features of short messages, the features utilized by contextual machinery training approaches are too sparse on the vector space model and lack the semantic content of short texts. A bidirectional long-term and short-term memory network model based on emotional multichannel is proposed by Z. G. Zhou. It combines deep learning's convolutional neural network features and attention mechanism with shallow learning's technique of learning short texts. The proposed model's accuracy and F1 value have improved significantly in the area of sentiment analysis for short sentences [16].

As CNNs can handle huge amounts of data and generate extremely precise predictions, they are especially helpful for computer vision applications such as picture recognition and categorization. CNN has been used for text classification tasks recently and has delivered pretty outstanding results. CNNs can efficiently recognize spatial correlations and patterns because of its architecture. The selected CNN models, conventional machine-learning models, and other novel approaches were tested using the three datasets: movie reviews, customer reviews, and the Stanford Sentiment Treebank dataset. About 81% and 68% accuracy for binary and ternary classification, respectively, were attained using the proposed CNN models [17].

3. Methodology

3.1. Dataset

The Quora Dataset, which includes 404290 question pairs, question ids, and the is_duplicate feature, has been used for this study [18]. Table 1 provides information about the different features in the dataset along with a description.

Table 1. Dataset Description

Feature	Description
id	An unique number allocated to every dataset row.
qid1, qid2	An identity that is unique to the questions in the columns labelled "question 1" and "question 2."
question1, question2	Actual questions need to be evaluated in order to look for duplicates.
is_duplicate	The outcome of a semantic comparison of question pairs is is_duplicate, where 0 denotes false and 1 denotes true.

3.2. Initial Preprocessing

The dataset undergoes basic preprocessing in order to be approved for use in the subsequent phase of the project. The first step of preprocessing entails looking for any missing questions, lowercasing, punctuation removal, HTML and URL tag removal, stopword removal, and chartword handling (e.g., N.A. stands for not applicable). Preprocessing techniques, including decontracting words, replacing some numbers with their string equivalents, replacing some special characters with their string equivalents, Tokenization, are applied to the dataset under consideration [19].

3.3. Feature Creation

The new features have been developed based on questions found in the dataset. By feeding datasets with and without newly created features to various machine learning algorithms, the effect of newly created features on accuracy is evaluated. The outcome demonstrates that the newly discovered features improve accuracy and are crucial for predicting the detection of duplicate questions. The following are the newly developed features [20]:

q1_size, q2_size= Question length, including all characters.

no_of_words-in_q1, no_of_words-in_q2= word count in question, including words that are repeated.

common_word= the number of words that are used similarly in both questions.

total_words=unique words in question1 + unique words in question2.

$$\mathbf{word\_share} = \frac{\text{common\_word}}{\text{total\_words}} \quad (1)$$

$$\mathbf{CWC_{min}} = \frac{\text{number of common word}}{\text{Min}[\text{len}(q1), \text{len}(q2)]} \quad (2)$$

$$\mathbf{CWC_{max}} = \frac{\text{number of common word}}{\text{Max}[\text{len}(q1), \text{len}(q2)]} \quad (3)$$

$$\mathbf{CSC_{min}} = \frac{\text{number of common stop word}}{\text{Min}\begin{bmatrix}\text{stop word count}(q1),\\ \text{stop word count}(q2)\end{bmatrix}} \quad (4)$$

$$\mathbf{CSC_{max}} = \frac{\text{number of common stop word}}{\text{Max}\begin{bmatrix}\text{stop word count}(q1),\\ \text{stop word count}(q2)\end{bmatrix}} \quad (5)$$

$$\mathbf{CTC_{min}} = \frac{\text{number of common token}}{\text{Min}\begin{bmatrix}\text{token count}(q1),\\ \text{token count}(q2)\end{bmatrix}} \quad (6)$$

$$\mathbf{CTC_{max}} = \frac{\text{number of common token}}{\text{Max}\begin{bmatrix}\text{token count}(q1),\\ \text{token count}(q2)\end{bmatrix}} \quad (7)$$

eq_last_word= 1 if both questions have an equal last word; otherwise, 0.

eq_first_word= 1 if both questions have an equal first word; otherwise, 0.

$$\mathbf{Mean\_len} = \frac{\begin{matrix}\text{question1 tokens} +\\ \text{question2 tokens}\end{matrix}}{2} \quad (8)$$

$$\mathbf{Abs\_len\_diff} = \left\|\begin{matrix}\text{question1 tokens} -\\ \text{question2 tokens}\end{matrix}\right\| \quad (9)$$

$$\mathbf{Longest_{substr_{ratio}}} = \frac{\begin{matrix}\text{common longest substring}\\ \text{in question 1 and 2}\end{matrix}}{\text{Min}\begin{bmatrix}\text{token count}(q1),\\ \text{token count}(q2)\end{bmatrix}} \quad (10)$$

$$\mathbf{fuzz\_ratio} = \frac{1}{1 + \text{Levenshtein distance}} \quad (11)$$

Usually, fuzz ratio is computed using algorithms that assess how similar two strings are to one another. The Levenshtein distance, which denotes the smallest number of single-character changes necessary to convert one string into another, is the foundation for one renowned calculation. Assume that the two strings, s1 and s2, have respective lengths of m and n. The Levenshtein distance is D(m,n), which is the least number of changes required to change s1 into s2. This distance is efficiently computed by the dynamic programming technique by filling up a (m+1)×(n+1) matrix.

$$\mathbf{fuzz_{partial_{ratio}}} = \frac{2 * \text{Common Characters}}{\begin{matrix}\text{len(question1)} +\\ \text{len(question2)}\end{matrix}} * 100 \quad (12)$$

The degree of similarity between the strings based on partial matching increases with increasing fuzz_partial_ratio.

$$\mathbf{token\_sort\_ratio} = \frac{\text{Levenshtein Distance}\begin{pmatrix}\text{Sorted Tokens in question1,}\\ \text{Sorted Tokens in question2}\end{pmatrix}}{\max\begin{pmatrix}\text{Length of Sorted Tokens in question1,}\\ \text{Length of Sorted Tokens in question2}\end{pmatrix}} * 100 \quad (13)$$

$$\mathbf{token\_set\_ratio} = \frac{\text{Levenshtein Distance}\begin{pmatrix}\text{Token Set in question1,}\\ \text{Token Set in question2}\end{pmatrix}}{\max\begin{pmatrix}\text{Length of Token Set in question1,}\\ \text{Length of Token Set in question2}\end{pmatrix}} * 100 \quad (14)$$

3.4. Vectorization

The term vectorization refers to the conventional approach of taking input data and turning it from text into vectors of real numbers, which is the format that machine learning models can understand. The vectorization algorithms CountVectorizer [21], N-gram [22], TF-IDF [23], Bag of Words [24], and Word2Vec [25] were used in the study.

Word2Vec: Word embedding is a technique for expressing words as vectors. Its primary objective is to maintain contextual similarity within the corpus while converting the high-dimensional feature space of words into low-dimensional feature vectors. Word2Vec first learns the vector representation of words by building a vocabulary from the training text input. Word2Vec utilized skip-gramme and CBOW architecture. The CBOW model predicts the middle word by combining the scattered representations of context, or surrounding words. The distributed representation of the input word is employed in the Skip-gram model to predict the context. Figure 1 illustrates the architecture of the CBOW and Skip-Gram, including details regarding the hidden layer, input layer, output layer, and related weight [25,26].

CBOW: The CBOW architecture includes a classification model based on deep learning that attempts to predict the target word, Y, by using context words, X, as input. It is frequently used to pre-train word embeddings that may be utilized for various NLP tasks like sentiment analysis, text classification, and machine translation. It is a form of "unsupervised" learning, which means that it can learn from unlabeled input. Both the target and the input layer are one-hot encoded with a size of [1 X V]. Two weight sets are randomly initiated, with one between the input and hidden layers W and the other between the hidden and output layers W'. Hidden activation is the result of multiplying the input by the input-hidden weights. The output Y is computed by multiplying the hidden input by the hidden-output weights. To re-adjust the weights, the difference between the output

and the target is computed and reported back [27,28].

Skip-Gram: The skip-gramme architecture simply reverses the CBOW layout by predicting the context words yi to ym for a given target word X. The input layer and the target are both one-hot encoded, and the random weight is assigned. The softmax function determines the probability of context words, computes the loss between anticipation and actual, and then backpropagates the error to adjust the assigned weight [29,30].

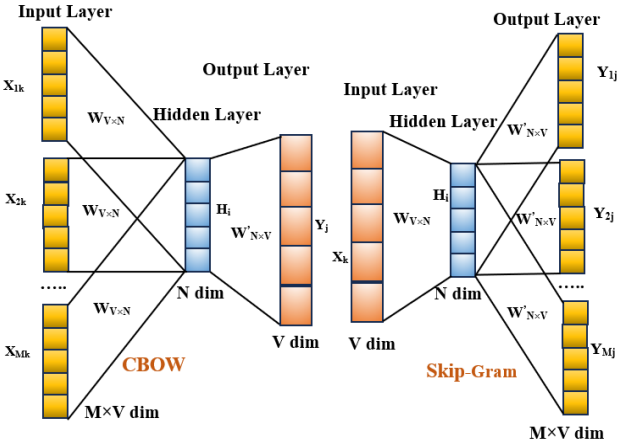

Fig. 1. Architecture for CBOW and Skip-Gram

3.5. Dimensionality Reduction

Dimensionality reduction is a form of feature extraction that tries to minimize the number of input features while retaining the largest portion of the original data in order to increase computing speed. One of the most effective methods for decreasing the dimensionality of datasets while retaining critical details in data analysis is principal component analysis (PCA) [31]. The principal components of PCA are a set of orthogonal axes that represent the largest variance in the data, and they are linear combinations of the original variables in the dataset, arranged in decreasing order of significance. The primary objective of PCA is to minimize the number of variables in the collection of data while preserving as much information as is possible. PCA is mostly used for significant feature selection and dimension reduction.

The data's major variation is captured by the first principal component, the second principal component, which captures the majority of the variance orthogonal to the first principal component, and so on. According to PCA, a feature's variance indicates how much information it has; the greater the variation in a feature, the more information it contains. Figure 2 illustrates how to identify the principal component [32].

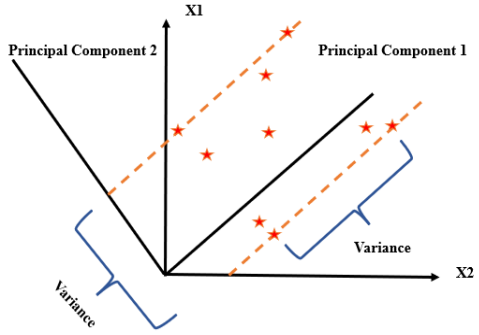

Fig. 2. Principal Component Analysis

To make sure that every variable has a mean of 0 and a standard deviation of 1, we must first standardize our dataset.

$$z = \frac{X - \mu}{\sigma} \qquad (15)$$

where σ is the independent feature's (X) standard deviation and μ is its mean.

The degree to which two or more variables fluctuate in relation to one another is indicated by covariance. We can use the following formula to get the covariance:

$$cov(x,y) = \frac{(x_i - \bar{x})/(y_i - \bar{y})}{N-1} \qquad (16)$$

Where xi, yi are the data values and x̄, ȳ are the mean.

Eigenvectors are non-zero vectors that stay in the same direction after applying a linear transformation. Assuming that V is an "n × n" linear transformation matrix and that λ is its eigenvalue, x, a non-zero vector, is an eigenvector if it meets the criteria stated below;

$$Vx = \lambda x \qquad (17)$$

After being multiplied by V, almost all vectors shift direction. A few uncommon vectors say x is in the same direction as Vx called Eigenvectors. The principal axes of the data are the eigenvectors of the covariance matrix, and the principal components are the projections of the data instances onto these principal axes. Afterwards, dimensionality reduction is achieved by keeping only the axes (dimensions) that contribute the most to the variance and eliminating the others [33].

3.6. Model Training

The various algorithms were assessed using several kinds of evaluation measures after being trained on vectorized data using Word2Vec. CNN is observed to outperform other algorithms when evaluated using classification metrics, along with Random Forest, Adaboost, XGBoost, and LSTM.

CNN: A popular neural network type for natural language processing applications is the Convolutional Neural Network (CNN) [34, 35], which is skilled at processing text

and other data sequences. In NLP, CNNs can be used for text categorization, automated translation, and language modelling. 1D-CNN is a subclass of CNN designed exclusively to analyze one-dimensional data sequences, such as text. It identifies patterns and features in the incoming data by running many filters over it [36,37]. The convolutional layers convolved the input, extracted features from the input, and passed the output to the next layer by swiping the filter over the input matrix.

The size of an output matrix is controlled by CNN using padding, which specifies how many pixels are added to an input matrix during the convolution process, and stride, which specifies the number of pixels moved, which regulates how the filter convolves across the input matrix [38, 39]. The pooling layer attempts to progressively decrease the spatial dimension of the representation in order to minimize the number of parameters and computations in the network by multiplying the resultant matrix from the convolution layer and pooling matrix [40]. By periodically setting the input units to zero at a random probability at each training step, the dropout layer helps to reduce overfitting. After being flattened into a one-dimensional array, the feed-forward neural network uses the array as input for further computation [41,42]. The different elements of CNN are highlighted in Figure 3, which provides a detailed description of CNN architecture.

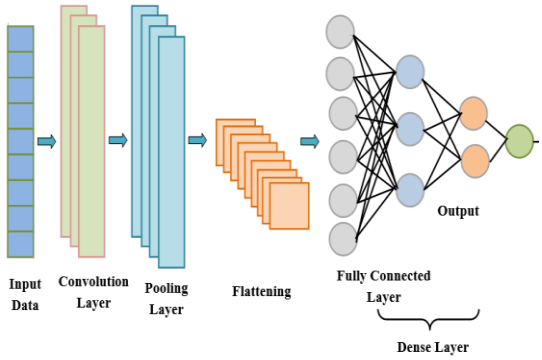

Fig. 3. CNN Architecture

C-CNN: Cascaded convolutional neural networks (C-CNN) can be constructed using several concurrent CNN architectures with different kernel sizes that read the input independently. The convolution, max pooling, flattening, flattening layers, etc. are components of each independent architecture [43]. The most frequent output from each channel is determined to form the final output

3.7. Evaluation Metrics

The various algorithms were assessed using several kinds of evaluation measures after being trained on vectorized data using Word2Vec [44,45].

Accuracy: The percentage of accurate predictions our classification model produces is referred to as accuracy.

$$\text{Accuracy} = \frac{[TP + TN]}{N} \qquad (18)$$

Precision: The precision shows the percentage of true positive predictions among all positive ones. The definition of it is the ratio of accurately predicted positive outcomes to all predicted positive outcomes.

$$\text{Precision} = \frac{TP}{[TP + FP]} \qquad (19)$$

Recall: Recall shows the percentage of truly positive values that are also anticipated to be positive. It is the proportion of accurate positive predictions to all positive occurrences in the dataset.

$$\text{Recall} = \frac{TP}{[TP + FN]} \qquad (20)$$

F1-Score: There are numerous circumstances in which recall and precision are equally crucial. In these circumstances, we utilise the F1-score, which is the harmonic mean of the recall and precision.

$$\text{F1Score} = 2 * \frac{[\text{Precision} * \text{Recall}]}{[\text{Precision} + \text{Recall}]} \qquad (21)$$

4. Results

Based on factors such as question length, string occurrences, common strings on both questions, fuzzy logic, etc., we have built several kinds of features. To evaluate the effectiveness of a newly created feature, different algorithms were trained on Quora's dataset and the new feature. Tables 2 and 3 provide a detailed description of the training algorithm used and the value of the evaluation metrics for the dataset without and with new features. The outcome demonstrates that adding a newly developed feature significantly raises the value of evaluation metrics.

Table 2. Evaluation Metrics Value for Different Algorithms Trained on Quora's Dataset

	Accurac	Precisio	Recal	F1scor
RandomFore	0.73	0.73	0.73	0.71
Adaboost	0.68	0.67	0.68	0.64
XGBoost	0.72	0.71	0.72	0.70
LSTM	0.71	0.70	0.71	0.70
CNN	0.72	0.72	0.72	0.71

Table 3. Evaluation Metrics Value for Different Algorithms Trained on Quora's Dataset with New Features

	Accurac	Precisio	Recal	F1scor
RandomFore	0.78	0.78	0.78	0.78
Adaboost	0.75	0.75	0.75	0.75
XGBoost	0.79	0.79	0.79	0.79
LSTM	0.77	0.77	0.77	0.77
CNN	0.79	0.79	0.79	0.79

The data's major variation is captured by the first principal Textual data is transformed into numerical form using vectorization techniques such as Word2Vec, CountVectorizer, N-Gramme, Bag of Words, and TFIDF. Using evaluation metrics, various ML algorithms were trained to determine the most effective vectorization strategy. The outcome demonstrates that Word2Vec vectorized data classification produces good results for each kind of classifier. Tables 4 to 7 demonstrate the evaluation metrics values for several algorithms trained on a dataset vectorized using various vectorization approaches.

Table 4. Evaluation Metrics for Various Algorithms Trained on A Bag of Words Vectorized Dataset

Bag of Words Vectorization				
	Accurac	**Precisio**	**Recal**	**F1scor**
RandomFore	0.78	0.78	0.78	0.78
Adaboost	0.75	0.75	0.75	0.75
XGBoost	0.79	0.79	0.79	0.79
LSTM	0.77	0.77	0.77	0.77
CNN	0.79	0.79	0.79	0.79

Table 5. Evaluation Metrics for Various Algorithms Trained on a Countvectorizer Vectorized Dataset.

CountVectorizer				
	Accurac	**Precisio**	**Recal**	**F1scor**
RandomFore	0.78	0.78	0.78	0.78
Adaboost	0.75	0.75	0.75	0.75
XGBoost	0.79	0.79	0.79	0.79
LSTM	0.77	0.77	0.77	0.77
CNN	0.79	0.79	0.79	0.79

Table 6. Evaluation Metrics for Various Algorithms Trained on a N-Gram Vectorized Dataset

TFIDF Vectorization				
	Accurac	**Precisio**	**Recal**	**F1scor**
RandomFore	0.78	0.78	0.78	0.78
Adaboost	0.75	0.74	0.75	0.74
XGBoost	0.78	0.78	0.78	0.78
LSTM	0.78	0.78	0.78	0.78
CNN	0.79	0.79	0.79	0.79

Table 7. Evaluation Metrics for Various Algorithms Trained on a Word2vec Vectorized Dataset

Word2Vec Vectorization				
	Accurac	**Precisio**	**Recal**	**F1scor**
RandomFore	0.79	0.79	0.79	0.79
Adaboost	0.76	0.76	0.76	0.76
XGBoost	0.79	0.79	0.79	0.79
LSTM	0.79	0.79	0.79	0.79
CNN	0.80	0.80	0.80	0.80

According to the results, CNN trained on the Word2vec vectorized dataset outperforms other algorithms trained on different vectorized datasets. In comparison to alternative vectorization methods, Word2Vec vectorization performs better. Several CNNs trained concurrently on the Word2Vec Quora's dataset, and the mode value, a common prediction made by most CNNs, is used to determine the final output. As compared to other techniques, the classification metrics value for cascaded CNN in Table 8 indicates that cascaded CNN has a significant potential for classifying duplicate questions.

Table 8. Evaluation Metrics for Cascaded CNN Trained on a Word2vec Vectorized Dataset

Word2Vec Vectorization				
	Accurac	**Precisio**	**Recal**	**F1scor**
CascadedCN	0.83	0.83	0.83	0.83

5. Conclusion

Reusing earlier responses or information, decreasing storage costs, and enhancing the user experience are all benefits of incorporating a Repeated Question Discovery system into a question-answering system. In order to detect questions with the same meaning that have already been asked and answered by another user, we have proposed a cascaded CNN based on Word2Vec vectorization in our work. Without compromising prediction accuracy, the use of PCA for dimensionality reduction significantly helps in training time reduction. As compared to other vectorization techniques, the classification algorithms are able to perform well on the Word2Vec vectorized dataset. Using Cascaded CNN, we were able to obtain outstanding outcomes with 83% classification accuracy in contrast to other state-of-the-art approaches. The platform is capable of maintaining an enhanced standard of content quality by detecting and responding to duplicate inquiries, which encourages users to submit innovative and informative questions and answers.

References

[1] H. Isotani, H. Washizaki, Y. Fukazawa, T. Nomoto, S. Ouji, S. Saito, "Sentence embedding and fine-tuning to automatically identify duplicate bug", Frontiers in Computer Science, vol. 4, 2023, doi: 10.3389/fcomp.2022.1032452.

[2] L. Wang, L. Zhang, and J. Jiang, "Duplicate Question Detection With Deep Learning in Stack Overflow", IEEE Access, vol. 8, pp. 25964- 25975, 2020, doi: 10.1109/ACCESS.2020.2968391.

[3] M. S. M. Jabbar, L. Kumar, H. W. Samuel, M.Y. Kim, S. Prabharkar, R. Goebel, and O. Zaiane, "DeepDup: Duplicate Question Detection in Community Question Answering", Proceedings of the 2021 5th International Conference on Deep Learning Technologies (ICDLT

'21), Association for Computing Machinery, New York, pp. 8–12, 2021, doi: 10.1145/3480001.3480021.

[4] H. Lattar, A. B. Salem, H. B. Ghezala, and H. B. Ghezala, "Duplicate record detection approach based on sentence embeddings", 2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 269-274, 2021, doi: 10.1109/WETICE49692.2020.00059.

[5] S. Rani, A. Kumar, N. Kumar, and S. Kumar, "Deep Neural Model for Duplicate Question Detection Using Support Vector Machines (Svm)", Turkish Journal of Computer and Mathematics Education, vol. 12, no. 6, pp. 4024-4033, 2021.

[6] S. Rani, A. Kumar, N. Kumar, "Eliminating Data Duplication in CQA Platforms Using Deep Neural Model", Computational Intelligence and Neuroscience, vol. 2022, doi: 10.1155/2022/2067449.

[7] O. Rakhmanov, "A Comparative Study on Vectorization and Classification Techniques in Sentiment Analysis to Classify Student-Lecturer Comments", 9th International Young Scientist Conference on Computational Science (YSC 2020), vol. 178, pp. 194–204, 2020, doi: 10.1016/j.procs.2020.11.021.

[8] Z. Vujovic, "Classification Model Evaluation Metrics", International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, 2021, doi:10.14569/IJACSA.2021.0120670.

[9] J. Babu, S. Thara, "Finding the Duplicate Questions in Stack Overflow using Word Embeddings", Procedia Computer Science, vol. 171, pp. 2729-2733, 2020, doi: 10.1016/j.procs.2020.04.296.

[10] D. Basavesha., and Y. S. Nijagunarya, Detecting Duplicate Questions in Community Based Websites Using Machine Learning, Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021, April 2021, doi:10.2139/ssrn.3835083.

[11] J. Babu, and S. Thara, Finding the Duplicate Questions in Stack Overflow using Word Embeddings, "Third International Conference on Computing and Network Communications (CoCoNet'19)", pp. 2729–2733, 2020, doi: 10.1016/j.procs.2020.04.296.

[12] L. Wang, L. Zhang and J. Jiang, Duplicate Question Detection With Deep Learning in Stack Overflow, IEEE Access, vol. 8, pp. 25964-25975, 2020, doi: 10.1109/ACCESS.2020.2968391.

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

[14] G. Vinodhini and R. M. Chandrasekaran, Sentiment classification using principal component analysis based neural network model, International Conference on Information Communication and Embedded Systems (ICICES2014), Chennai, India, 2014, pp. 1-6, doi: 10.1109/ICICES.2014.7033961.

[15] G. Malik, M. Cevik, and A. Başar, Data Augmentation for Conflict and Duplicate Detection in Software Engineering Sentence Pairs, "CASCON '23: Proceedings of the 33rd Annual International Conference on Computer Science and Software Engineering", pp. 34–43, sept. 2023, doi: 10.5555/3615924.3615928.

[16] Z. G. Zhou, Research on Sentiment Analysis Model of Short Text Based on Deep Learning, Hindawi Scientific Programming, vol. 2022, doi: 10.1155/2022/2681533.

[17] H. Kim, and Y. S. Jeong, Sentiment Classification Using Convolutional Neural Networks, Applied Sciences, vol. 9, no. 11,2019, doi:10.3390/app9112347.

[18] Quora Question Pairs: <https://www.kaggle.com/c/quora-question-pairs/data>.

[19] M. A. Palomino, and F. Aider, "Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis", Applied Sciences, vol.12, no. 17, 2022. doi: 10.3390/app12178765.

[20] N. Ansari, and R, Sharma, "Identifying Semantically Duplicate Questions Using Data Science Approach: A Quora Case Study", ACM Conference, 2020, doi: 10.48550/arXiv.2004.11694.

[21] N. Alvi, and K. H. Talukder, "Sentiment Analysis of Bengali Text using CountVectorizer with Logistic Regression," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, pp. 01-05, 2021, doi: 10.1109/ICCCNT51525.2021.9580017.

[22] P. Rajesh, and G. Suseendran, "Prediction of N-Gram Language Models Using Sentiment Analysis on E-Learning Reviews," 2020 International Conference on Intelligent Engineering and Management (ICIEM), London, UK, pp. 510-514, 2020, doi: 10.1109/ICIEM48762.2020.9160260.

[23] S. Sumesh, and S. H. Aswini, "Natural Language Processing based Recommendation System for Courses *," 2023 International Conference on

Inventive Computation Technologies (ICICT), Lalitpur, Nepal, pp. 930-936, 2023, doi: 10.1109/ICICT57646.2023.10134234.

[24] M. Sharma, G. Choudhary, and S. Susan, "Resume Classification using Elite Bag-of-Words Approach," 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, pp. 1409-1413, 2023, doi: 10.1109/ICSSIT55814.2023.10061036.

[25] S. Nazir, M. Asif, S. A. Sahi, S. Ahmad, Y. Y. Ghadi, and M. H. Aziz, "Toward the Development of Large-Scale Word Embedding for Low-Resourced Language," in IEEE Access, vol. 10, pp. 54091-54097, 2022, doi: 10.1109/ACCESS.2022.3173259.

[26] D. S. Asudani, N. K. Nagwani, and P. Singh, "Impact of word embedding models on text analytics in deep learning environment: a review" Artificial Intelligence Review, vol. 56, pp. 10345–10425, 2023, doi: 10.1007/s10462-023-10419-1.

[27] M. Švaňa, "Extending Word2Vec with Domain-Specific Labels," 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS), Sofia, Bulgaria, pp. 157-160, 2022, doi: 10.15439/2022F37.

[28] A. Samih, A. Ghadi, and A. Fennan, "ExMrec2vec: Explainable Movie Recommender System based on Word2vec" International Journal of Advanced Computer Science and Applications(IJACSA), vol. 12, no. 8, 2021, doi:10.14569/IJACSA.2021.0120876.

[29] E. M. Dharma , F. L. Gaol , H. L. H. S. Warnars , and B. Soewito, "The Accuracy Comparison Among Word2vec, Glove, And Fasttext Towards Convolution Neural Network (Cnn) Text Classification", Journal of Theoretical and Applied Information Technology, vol.100, no 2, 2022.

[30] A. Desai, A. Zumbo, M. Giordano, P. Morandini, M. E. Laino, E. Azzolini, A. Fabbri, S. Marcheselli, A. L. Giotta, S. Luzzi, et al. "Word2vec Word Embedding-Based Artificial Intelligence Model in the Triage of Patients with Suspected Diagnosis of Major Ischemic Stroke: A Feasibility Study", International Journal of Environmental Research and Public Health, vol. 19, no. 22, 2022, doi:10.3390/ijerph192215295.

[31] R. Drikvandi, O. Lawal, "Sparse Principal Component Analysis for Natural Language Processing", Annals of Data Science, vol. 10, pp. 25-41, 2023, doi: 10.1007/s40745-020-00277-x.

[32] O. A. Alomari, A. Elnagar, I. Afyouni, I. Shahin, A. B. Nassif, I. A. Hashem, and M. Tubishat, "Hybrid Feature Selection Based on Principal Component Analysis and Grey Wolf Optimizer Algorithm for Arabic News Article Classification," IEEE Access, vol. 10, pp. 121816-121830, 2022, doi: 10.1109/ACCESS.2022.3222516.

[33] S. W. Choi, and B. H. S. Kim, "Applying PCA to Deep Learning Forecasting Models for Predicting PM2.5" Sustainability, vol. 13, no. 7 2021, .doi: 10.3390/su13073726.

[34] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", Journal of Big Data, vol. 8, no. 53, 2021, doi: 10.1186/s40537-021-00444-8.

[35] P. Choudhary, and P. Pathak, "A Review of Convolution Neural Network Used in Various Applications," 2021 5th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, pp. 1-5, 2021, doi: 10.1109/ISCON52037.2021.9702315.

[36] N. A. Mazlan, K. A. Othman, S. Shahbudin, and M. Kassim, "Convolution Neural Network (CNN) Architectures Analysis for Photovoltaic (PV) Module Defect Images Classification," 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, pp. 390-395, 2022, doi: 10.1109/CENIM56801.2022.10037564.

[37] S. Allamy, and A. L. Koerich, "1D CNN Architectures for Music Genre Classification," 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, pp. 01-07, 2021, doi: 10.1109/SSCI50451.2021.9659979.

[38] E. U. H. Qazi, A. Almorjan, and T. Zia, "A One-Dimensional Convolutional Neural Network (1D-CNN) Based Deep Learning System for Network Intrusion Detection", Applied Sciences, vol. 12, no. 16, 2022, doi: 10.3390/app12167986.

[39] D. Kilichev, and W. Kim, "Hyperparameter Optimization for 1D-CNN-Based Network Intrusion Detection Using GA and PSO", Mathematics, vol. 11, no. 17, 2023, doi: 10.3390/math11173724.

[40] N. B. Korade, and M. Zuber, "Stock Price Forecasting using Convolutional Neural Networks and Optimization Techniques", vol. 13, no. 11, pp. 378-385, 2022, doi: 10.14569/IJACSA.2022.0131142.

[41] N. B. Korade, and M. Zuber, "Boost Stock Forecasting Accuracy Using the Modified Firefly Algorithm and Multichannel Convolutional Neural Network", Journal of Theoretical and Applied Information Technology, vol. 101, no. 7, pp. 2668- 2677, 2023.

[42] N. B. Korade, and M. Zuber, “Stock Forecasting Using Multichannel CNN and Firefly Algorithm”, Proceedings of the 2nd International Conference on Cognitive and Intelligent Computing, pp. 447-458, 2023, doi: 10.1007/978-981-99-2742-5_46.

[43] Y. Nam, and C. Lee, “Cascaded Convolutional Neural Network Architecture for Speech Emotion Recognition in Noisy Conditions”, sensors, vol.21, no. 13, 2021, doi: 10.3390/s21134399.

[44] S. Manna, "Small Sample Estimation of Classification Metrics," 2022 Interdisciplinary Research in Technology and Management (IRTM), Kolkata, India, 2022, pp. 1-3, doi: 10.1109/IRTM54583.2022.9791645.

[45] R. G. Guendel, F. Fioranelli, and A. Yarovoy, "Evaluation Metrics for Continuous Human Activity Classification Using Distributed Radar Networks," 2022 IEEE Radar Conference (RadarConf22), New York City, NY, USA, 2022, pp. 1-6, doi: 10.1109/RadarConf2248738.2022.9764181.