IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 January 2024, accepted 16 February 2024, date of publication 20 February 2024, date of current version 28 February 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3367978*

RESEARCH ARTICLE

# Employing Siamese MaLSTM Model and ELMO Word Embedding for Quora Duplicate Questions Detection

**ABDULAZIZ ALTAMIMI**[1], **MUHAMMAD UMER**[2], **DANIAL HANIF**[2], **SHTWAI ALSUBAI**[3], **TAI-HOON KIM**[4], **AND IMRAN ASHRAF**[5]

[1]Department of Computer Science and Engineering, University of Hafr Al Batin, Hafar Al Batin 39524, Saudi Arabia
[2]Department of Computer Science and Information Technology, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[3]Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[4]School of Electrical and Computer Engineering, Yeosu Campus, Chonnam National University, Yeosu-si, Jeollanam-do 59626, Republic of Korea
[5]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea

Corresponding authors: Imran Ashraf (ashrafimran@live.com) and Tai-Hoon Kim (taihoonn@chonnam.ac.kr)

This work was supported by Prince Sattam bin Abdulaziz University under Project PSAU/2024/R/1445.

**ABSTRACT** Quora is an expanding online platform, that contains a growing collection of questions and answers generated by users. The content on this platform is managed by its users which involves creating, editing, and organization. Due to the vast number of users, it is not uncommon to find multiple questions with similar intents, leading to the problem of duplicate and identical questions. Detection of these duplicates could effectively lead to a more efficient search for high-quality answers, ultimately improving the user experience for both readers and writers on Quora. This study utilizes the dataset of Question Pairs for Quora obtained from Kaggle for identifying questions that are duplicates or identical. To vectorize the questions and for model training, six types of word embeddings are implemented including GoogleNewsVector, FastText crawl, FastText crawl sub-words, bidirectional encoder representations from transformers (BERT), robustly optimized BERT pretraining approach (RoBERTa), and embeddings from language models (ELMO) containing 100 dimensions. The Siamese Manhattan long short-term memory (MaLSTM) neural network model, where Ma is Manhattan distance, is applied with ELMO word embedding to predict duplicate questions in the dataset. Experimental results demonstrate that the proposed model attained an accuracy of 95.68% which surpasses the state-of-the-art models.

**INDEX TERMS** Quora, identical questions, word vector representation, MaLSTM.

## I. INTRODUCTION

Quora one of the largest question-and-answer platforms on the internet, hosts a vast amount of user-generated content [1]. With millions of questions being asked and answered, the issue of duplicate questions arises frequently [2]. Duplicate questions not only clutter the platform but also lead to redundant content and a poor user experience. Therefore, the task of automatically detecting duplicate question pairs on Quora has gained significant attention.

Quora duplicate question pair detection involves developing machine learning models and algorithms capable of identifying whether a given pair of questions are duplicates or not [3]. This task poses several challenges due to the semantic complexity of natural language and the nuances involved in determining question similarity. Researchers and data scientists have explored various approaches and techniques to tackle this problem effectively [4], [5].

Quora is a social media platform in terms of websites and apps that allow users to ask and answer questions from other users on various topics. These platforms facilitate communication and knowledge sharing among people with different backgrounds, expertise, and interests. Quora is an example of such a platform where users post questions that are later answered by the users having expertise in domains related to the asked question. Users can collaborate

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

**TABLE 1.** Examples of non-duplicate and duplicate pairs of questions.

| Question 1 | Question 2 | Target class |
|---|---|---|
| How do I make friends. | How to make friends? | Duplicate |
| Why do Slavs squat? | Will squats make my legs thicker? | Non-Duplicate |
| Best way to train for a marathon? | What are some tips for training for a marathon? | Duplicate |
| She is always sad? | Aerodynamically what happens when the propellor rotates? | Non-Duplicate |

by modifying questions and proposing more precise answers to the posted queries. 300M unique users visit Quora every month as per the stats of the Director of Product Management of Quora, and this causes problems of redundant or the same questions asked by different users with a choice of different words. As a result of these questions, it is hard to find the best answer for the reader to the questions, and also writers need to answer multiple questions in the same context. As a result, maintaining a single question thread for logically distinct questions is an important rule on Quora. For instance, questions like "Which laptop is most suitable for coding?" and "What are some good laptops for programming?" are alike as they have the same meaning and require answers only once. Although the questions "What is the best book ever made?" and "What is the most important book you have ever read?" use different phrasing, they share the same context and are considered duplicate questions, and maintaining separate pages for such questions can be a burden. Identifying and merging duplicate questions on Quora enhances the efficiency and effectiveness of knowledge sharing in several ways. By compiling all the questions on a single thread, readers can access answers to multiple questions while writers can avoid repeating the same response on multiple pages for the same question. If readers are divided into several threads, then a large number of users can be acquired to visit. At present, the random forest (RF) model is used by Quora along with several handcrafted features to combine duplicate questions. However, this approach may not be highly efficient when processing large amounts of data. In 2017, Quora held a contest on Kaggle that was motivated by developments in deep learning and machine learning models. The participants were asked to use cutting-edge machine learning and deep learning techniques to the dataset to increase the accuracy and reliability of the results. By utilizing neural network architecture, the main idea of [6] is to increase accuracy and cut down the time used in complex feature engineering. A key component of natural language processing (NLP), which has many applications including text classification, recognizing textual entailment (RTE) [6], information retrieval, plagiarism detection, and paraphrase recognition is the identification of duplicate questions [7].

The degree of similarity between two fragments of questions is measured to determine if they are semantically similar, which indicates that they may receive the same answer and are considered duplicates. Since the true meaning of a sentence cannot be accurately determined due to the ambiguous language and synonymous expression, it can be difficult to identify questions that are duplicates. Measuring sentence semantic similarity has been the focus of some studies. In [8], a method is designed for measuring the similarity between sentences semantically. The method utilizes WordNet, leveraging a tool developed by the researcher. Notably, this method does not involve the use of machine learning or deep learning models. The study presents a model specifically designed for the identification of similar or duplicate questions. The organization of a pair of questions is done on the basis of similarity in both their semantic meaning and wording. For experimentation, the dataset of a pair of Questions from Quora can be downloaded from Kaggle.

In Table 1, the first and third question pairs have similar words like 'friends' in the first and 'marathon, training' in the third. That is why they lie in the duplicate target class. The second and fourth example does not contain many similar words not even in the same sense of synonyms and they are placed under the label of non-duplicate in the dataset.

- A novel model is introduced for duplicate question-pair detection. The proposed Manhattan long short-term memory (MaLSTM) model makes use of embeddings from language model (ELMO) word embedding for the final predictions.
- The performance of the proposed model is tested with five other word embeddings like GoogleNewsVector, Fasttext, Fasttext crawl sub-word, bidirectional encoder representations from transformers (BERT), and robustly optimized BERT pretraining approach (RoBERTa).
- Furthermore, the performance of the proposed model is compared with other state-of-the-art approaches.

The rest of the article is divided as follows. Section II gives an overview of relevant studies associated with this work. In Section III, the proposed model, the dataset used, the steps of data preprocessing, and the underlying rationale behind the deep learning models employed in this study are discussed. Section IV examines the findings of experiments and their analysis. Lastly, Section V concludes the article by offering future suggestions.

## II. RELATED WORK

Given the variety of language and the difficulty in determining a sentence's exact meaning, it can be difficult to spot duplicate questions. A well-researched task in NLP is identifying paraphrases, which is comparable to this task [9]. Natural language sentence matching (NLSM), a technique used to spot duplicate questions, ascertains whether two sentences, even those with different wording, express the same idea [10]. Researchers' primary interest in conventional approaches has been feature engineering, and popular features include term frequency (TF), a bag of words (BoW), inverse document frequency (IDF), and N-grams. Support vector machine (SVM) model, which uses various techniques of feature extraction like BoW vectors, is a key technique in the categorization of text [11]. Deep learning techniques have recently demonstrated impressive performance in a variety of NLP tasks, especially in the area of semantic text similarity [12], [13], [14]. Deep learning models that were trained using task-specific feature engineering have achieved outstanding outcomes in similarity measurement and semantic analysis. Pre-trained word embeddings have been demonstrated by researchers to be able to capture significant semantic symmetries [15]. Word embeddings and deep models can be combined to accurately represent the semantic meaning of the text.

In tasks like text categorization and information retrieval, long short-term memory (LSTM)-based neural networks [16] have displayed exceptional performance. The study [17] suggested semi-supervised and supervised methods utilizing LSTM, along with a technique known as region embedding, to embed text regions of varying sizes. In another study [18], a neural network model was introduced that incorporated a recurrent neural network (RNN) with distributed vector representation. This model effectively integrated document vectors while analyzing the contextual information and relationships between sentences. For tasks such as the classification of text and sentence representation, the study employed a methodology called the C-LSTM network, which combines convolutional neural network (CNN) and LSTM neural networks. High-level features were extracted by this architecture using CNN and later were fed to LTSM [19]. In [20], an LSTM model was proposed for predicting the resemblance between two sentences. A skip-thought-based approach was also proposed that utilized a skip-gram approach from word to sentence level with RNN to obtain a skip-through vector for each sentence, which was then used to reconstruct previous and next sentences [21].

The Siamese architecture is still frequently used as a learning framework for placing question-answer pairs in a common space [22] despite the earlier studies. In a separate investigation [12], a different strategy was studied and sentences were converted using pre-trained word embedding vectors and Siamese LSTM. To arrive at the conclusion, the Manhattan distance was used to gauge the proximity of the sentence pairs. CNN-based approaches have demonstrated exceptional performance in a number of NLP tasks [23] other than just classification. Attention-based CNN is also used for finding duplicate question pairs [24]. Another study [25]] used the Siamese CNN model, which combined convolutional and pooling processes to create sentence embeddings. However, using pre-trained word embeddings that are not specifically adapted to the dataset limits the performance of these models.

The Quora dataset has received limited attention in the research community, with only a few studies conducted [26]. In one of these studies, a CNN-based model leveraging global vectors for word representation (GloVe) embeddings, which are $100 dimensions$ Wikipedia vectors, achieved an accuracy of 80.4% [10]. Another approach presented in [27] employed a bi-layer similarity network along with a Siamese gated recurrent unit (GRU), resulting in an accuracy of 85%. In addition, the accuracy of a support vector classifier trained using precomputed features such as longest common substring, subsequences, and word similarity derived from lexical and semantic resources was 85.0% [27]. Furthermore, a "matching-aggregation" framework called bilateral multi-perspective matching (BiMPM) was utilized in [28], achieving an accuracy of 88.17%. Recently, a graph-based matching model has been proposed to detect duplicate questions on the Quora network [29].

Detecting duplicate question pairs is challenging because natural language is ever-changing and nuanced, and conventional rule-based methods often fail to capture these subtle variations. The authors applied deep learning models LSTM and bidirectional LSTM (BiLSTM) in combination with word embedding to encode question pairs [30]. Chandra et al. [31] applied GLoVe word embedding to extract similarity and proposed siamese LSTM. The authors applied the Siamese network, Bert and BiLSTM, and MaLSTM to detect duplicate questions on the Quora network and achieved the highest accuracy with the Bert model [32]. These progressive methodologies signify the ongoing efforts to conquer the challenges of duplicate question pair detection within the ever-evolving landscape of natural language.

Another study along the same lines is [33] which focused on the detection of duplicate questions on community-based platforms. The authors propose an interaction-based Siamese network (ISN) to resolve the issues of error propagation and low-level semantics loss. In addition, an aggregation strategy is proposed for the propagation of low-level to high-level interaction features. This strategy helps to preserve low-level semantic information. Results show that the proposed ISN model outperforms base models with a 0.86 accuracy score while obtaining a 0.85 score each for precision, recall, and F1. Similarly, the study [34] presents an approach to detect near-duplicate and semantically related questions using an unsupervised approach. The proposed approach combines statistical and neural approaches for this purpose. The proposed model QDup focused on increased accuracy and speed for duplicate question detection. The proposed approach shows better accuracy than keyphrases-based and closest neighbor methods with an 81.5% accuracy.

Unlike the majority of the techniques mentioned above, this study takes a different track. For higher-level feature engineering, it uses a variety of word embeddings rather than just one pre-trained one, such as the GooglNewsVector, FastText crawl, FastText crawl sub-word, BERT, RoBERTa, and ELMO embeddings. The inclusion of word vectors from several fields in the embeddings leads to a larger training word-vector size and a wider variety of training domains. This work employs the MaLSTM deep model, which produces an output vector containing the final hidden state, to process

**TABLE 2.** Dataset attributes description.

| Attribute | Description |
|---|---|
| Question 1 ID | Question 1 complete ID. |
| Question 2 ID | Question 2 complete ID. |
| Question Pair ID | Question pair ID. |
| Question 1 | Full text of question 1. |
| Question 2 | Full text of question 2. |
| Duplicate | Target class, the label is 1 if the question pair is a duplicate else 0 if the question pair is not duplicated. |

the input vectors of each sentence. To determine how similar these representations are to one another, the Manhattan distance is used. By correctly identifying 20 out of 20 pairs of questions, the proposed method outperforms other feature extraction and approaches to deep learning in terms of accuracy.

## III. MATERIALS AND METHODS

A sentence is made up of a group of words that combine to form clauses and phrases. By looking at a sentence's structure and constituent parts, one can determine its meaning. The relationships between words can be examined using neural networks from a variety of angles. In order to determine the semantic relevance between the two questions, this paper introduces a novel Siamese MaLSTM model. The use of two or more identical network structures at once is referred to as "Siamese" usage. The initial "Ma" in "MaLSTM" stands for the Manhattan distance approximation method, and is used for calculating how similar two textual features are to one another. By usage of three gates for processing of input, the LSTM serves as a sequence modeling technique that can detect long-term dependencies. The proposed model adopts an approach that combines three feature engineering methods: GoogleNewsVector, FastText crawl, and FastText crawl sub-words. The deep learning model siamese MaLSTM is implemented using the Keras framework and visualization is done using matplotlib and seaborn library.

### A. DATASET

A data set containing 404351 entries of question pairs was made public by Quora in January 2017 [35]. These pairs of questions cover a wide range of domains, such as technology, entertainment, politics, culture, and philosophy. The dataset was obtained from Kaggle [36]. Every entry in the dataset consists of a question pair and target class representing whether questions are identical or not. Then dataset is divided into two parts, i.e., 75% for training and 25% for testing. Table 2 provides details of attributes of the dataset, along with their descriptions.

As the classifier focuses solely on the attributes "Question 1" "Question 2", and "Duplicate," the remaining attributes are ignored from the dataset. In Table 3, samples of non-duplicate and duplicate questions are listed from the dataset.

**TABLE 3.** Examples of duplicate question pairs.

| Duplicate | Non-duplicate |
|---|---|
| How can I be a good geologist? | Why do Slavs squat? |
| What should I do to be a great geologist? | Will squats make my legs thicker? |

### B. PREPROCESSING

Preprocessing is the process of addressing redundant, inconsistent, and missing values in order to arrange data in a comprehensible format. Several preprocessing techniques are used to remove noise and reduce data complexity. NLP techniques, such as converting text to lowercase, removing stop words, stemming, and tokenization, are implemented using readily available libraries like natural language processing tool kit (NLTK) and Keras. These preprocessing steps enhance the data quality by removing unnecessary details. Using the tokenizer method from Keras's library, each question is converted into a word vector. The extraction of useful features also makes use of word embeddings like GoogleNewsVector, FastText crawl, FastText crawl subwords, BERT, RoBERTa, and ELMO. The total number of characters in questions may not exceed 45 in order to maintain consistency. Zero padding is used for questions under 45 characters. The Siamese MaLSTM architecture is then used to predict labels using the preprocessed features. The architecture uses the processed data to generate predictions based on the features that were extracted.

### C. PROPOSED METHODOLOGY

Figure 1 shows the architectural workflow of the proposed approach. The dataset is used with ELMO word embedding to extract features which are later fed into MaLSTM. The model is trained on these features to determine the duplicate and non-duplicate question pairs.

### D. WORD EMBEDDING

Deep models lack natural comprehension of spoken or written input. Each query must be vectorized in order for these models to understand such input. The first layer in the proposed model is an embedding layer made to take the input of the question pairs and convert each word into a vector representation. The maximum sequence length is 45 and the embedding dimension is set at 100. In this particular study, six distinct word embeddings GoogleNewsVector, FastText crawl, FastText crawl subword, BERT, RoBERTa, and ELMO are employed. These embeddings provide valuable representations of words, enhancing the model's ability to capture semantic information and meaning.

#### 1) GOOGLENEWSVECTOR

Google offers pre-trained word embeddings derived from a news corpus. These word embeddings encompass a vast vocabulary of 3 million English words, each represented by $300-dimensions$. In total, this pre-trained model provides an extensive collection of 3 billion word vectors [37].

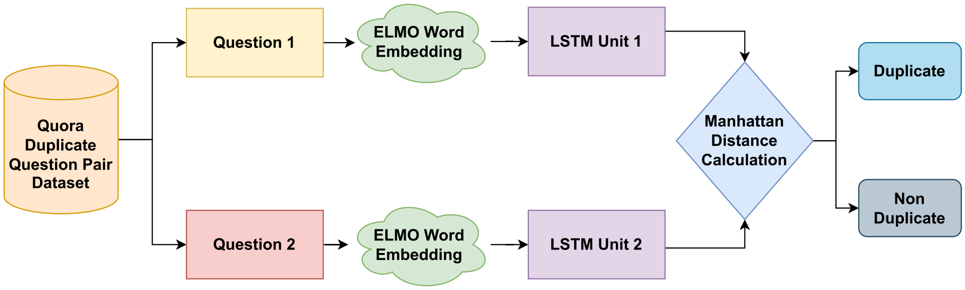


**FIGURE 1.** Workflow diagram of the proposed model.

### 2) FASTTEXT

A library named FastText, developed by Facebook for learning word representation is known for its efficiency. This library encompasses a comprehensive set of 2 million common words, each represented by 300 − *dimensions*. This amounts to a vast collection of 600 billion word vectors. What sets FastText apart from Google word embeddings is its distinctive feature of providing n-gram character-level representations of words [38].

### 3) FASTTEXT SUBWORD

FastText Subword comprises a collection of 2 million word vectors trained on the Common Crawl dataset, which consists of a massive 600 billion tokens. In contrast to traditional word embeddings, sub-word embeddings offer more detailed information by breaking down each word into its constituent sub-words. For instance, if we consider the word "where" with a value of $n$ equal to 3, the resulting sub-words would be "we," "her," and "ere." Ultimately, the FastText sub-word provides a dictionary that encompasses the union of these sub-words, enriching the representation of words [39].

### 4) BIDIRECTIONAL ENCODER REPRESENTATION FROM TRANSFORMERS

BERT revolutionized NLP with its innovative implementation of a highly effective bidirectional self-attention mechanism. This mechanism is trained on an extensive collection of data including the BookCorpus, which encompasses 11038 unpublished books in plain text format across 16 diverse genres. Additionally, it utilizes 2500 million words extracted from English Wikipedia passages to further enhance its capabilities [40]. Unlike context-free models such as Word2Vec, BERT utilizes a bidirectional contextual model that takes into account both the preceding and following words in a sentence. Consequently, contextual models capture different word representations based on the specific sentence context, offering a more comprehensive understanding of language nuances. In contrast, context-free models assign the same representation to a given word regardless of its context in different sentences. The BERT model undergoes training on diverse unlabeled data corpora, encompassing various scenarios. In the subsequent fine-tuning phase, the model starts with pre-trained parameters as its initialization. BERT utilizes the "[MASK]" symbol to predict missing tokens in the text. However, BERT does have a few noteworthy drawbacks. Firstly, the reconstruction of all masked tokens and corrupted versions in the joint conditional probability is conducted independently, which can be seen as a limitation. Secondly, masked tokens do not appear in the downstream tasks, resulting in a disparity between the pre-training and fine-tuning stages. However, one of the significant advantages of BERT's autoencoder (AE) language modeling approach is its ability to capture bidirectional context, enabling a more comprehensive understanding of language.

### 5) ROBUSTLY OPTIMIZED BERT APPROACH

RoBERTa, an enhanced version of BERT, shares many similar configurations with BERT but shows improved performance [40]. This can be observed from the GLUE leaderboard. RoBERTa surpasses BERT in performance by implementing several significant modifications. These changes involve leveraging a larger training dataset, adopting dynamic masking patterns, training on lengthier sequences, and replacing the next sentence prediction task. In essence, RoBERTa fine-tunes BERT by primarily augmenting the data size and optimizing hyperparameters. In RoBERTa, dynamic masking is applied to every training instance during each epoch. This is accomplished by replicating the training dataset ten times, resulting in each sequence being masked in ten distinct ways throughout the course of forty training epochs.

### 6) EMBEDDINGS FROM LANGUAGE MODELS

Traditional word embedding methods often struggle to capture contextual information and accurately distinguish between polysemous words [41]. As a result, these methods tend to generate the same representations for words like "read" regardless of the specific context in which they appear. In contrast, word embeddings derived from ELMo align with the contextual nuances of different sentences. These embeddings are generated by leveraging the learned functions of all the internal layers within a bidirectional LSTM model. As a result, the representations of the word "read" in different contexts vary, capturing their unique contextual usage. ELMo provides notable benefits in generating contextualized representations, making it essential to utilize ELMo embeddings for various NLP tasks, particularly those involving text similarity calculations.

### 7) SIAMESE DEEP LEARNING NETWORK

Siamese networks are AI neural networks capable of simultaneously processing one or more input vectors and combining the output vectors through sub-identical neural network computations [42]. To reduce training parameters and the risk of overfitting, the weights in Siamese networks are shared among all the inputs. The concept of shared weights was initially proposed in 1994 [43]. Siamese networks can accommodate various types of input data, such as textual, graphical, or numeric data.

These networks have shown to be effective for a variety of tasks involving the establishment of relationships between two patterns, such as recognizing forged signatures, pattern recognition, and paraphrase identification [43]. Sub-identical network models process inputs with comparable properties, enabling the extraction of comparable and similar features. At the output, the Siamese network performs binary classification, indicating whether or not the inputs are members of the same class. If input belongs to the same class, it suggests that they are marked as duplicates and are similar in some way. The Siamese network uses a neuron to determine how far apart the feature vectors are when merging the output of processed inputs. The questions are categorized as whether they are identical or not based on the calculated distance, giving information about how similar they are.

### 8) MALSTM

By utilizing its numerous internal layers, LSTM, a well-known sequence modeling technique, efficiently generates long-term sequences. The output gate ($o_t$), cell memory block ($c_t$), input gate ($i_t$), and forget gate ($f_t$) are its four fundamental parts. The LSTM layer receives real-valued vectors as input. The gates sequentially update the hidden state representations ($h_t$), with the update procedure heavily reliant on the cell memory block ($c_t$). The inclusion or exclusion of information in the final prediction is determined by these four factors taken together. Different LSTM variations are created to handle various problem types [44].

In this experiment, two LSTM variants; the 1st variant (1) and the 2nd variant (2) were applied. For constructing lengthy sequences out of textual data, these variations work best. They employ a sigmoid layer to determine whether a piece of information is relevant to the final prediction. The output of the sigmoid layer ranges from 0 to 1, with 0 indicating information omission and 1 indicating information used in the final prediction. In this context, the terms "$W_i$", "$h_t$", and "$b_i$" refer to weights applied to the input vectors, "current input" to the neuron, and "bias value added to the inputs". Equations 3 - 6 show LSTM variants and they make use of *tanh* activation function for producing the sequences based on the topic of the text.

**TABLE 4.** Hyperparameter details of proposed model.

| Hyperparameter | Value |
|---|---|
| Dropout LSTM | 0.23 |
| Dropout Dense | 0.23 |
| Regularizing | 0.002 |
| Hidden Layers | 300 |
| Batch Size | 1024 |
| Optimizer | Adam |
| Activation | ReLu |

$$I_t = sigmoid(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = sigmoid(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$c_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (3)$$

$$c_t = i_t \odot c_t + f_t \odot c_{t-1} \quad (4)$$

$$o_t = sigmoid(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t \odot tanh(c_t) \quad (6)$$

Combinations on variable-length space vectors, specifically combinations made up of $d_{in}$-dimensional vectors, can be produced by LSTM. The input vectors in this experiment have a dimension size of 100 ($dim = 100$). Word vectors ($a_1, a_2, a_3, \ldots, a_N$) are used to represent each question. The maximum length of a sequence is 20 to ensure consistency. As a result, no question in the dataset is longer than 20 characters. Zero-padding is used for questions with a length of less than 40 to maintain uniformity.

The MaLSTM employs a Siamese architecture in which each sentence in the input sentence pair is processed by one of two identical LSTM sub-networks (LST$M_a$ and LST$M_b$). These input sentences are given equal weights and converted to a real-valued word vector form, which fixes the variable length input sequences into a fixed length vector form [12]. One question is processed for each assigned weight. Manhattan distance is determined by MaLSTM for the final forecast. The Manhattan distance to some extent outperforms other substitutions like cosine similarity [12].

Figure 2 displays a diagram of the Siamese MaLSTM architecture that has been used in this study. Due to our work on a large set of multidimensional word embedding, we chose the Manhattan distance over other similarity measures. Since the Manhattan distance similarity measure calculates the absolute distance between two points that are at right angles to determine the similarity between textual features, many researchers have noticed that it not only performs well with very high dimensional data but also computes more quickly [9], [11], [28]. The Manhattan distance equation is given in Equation 7 for the two points $x$ and $y$

$$M_a = |x_1 - x_2| + |y_1 - y_2| \quad (7)$$

where $x_1$ and $y_1$ represent the output of the first model and $x_2$ and $y_2$ represent the output of the second.

The absolute difference between the two inputs to the model indicates how similar they are to one another. We used a threshold of 0.5 in this experiment to tell which questions

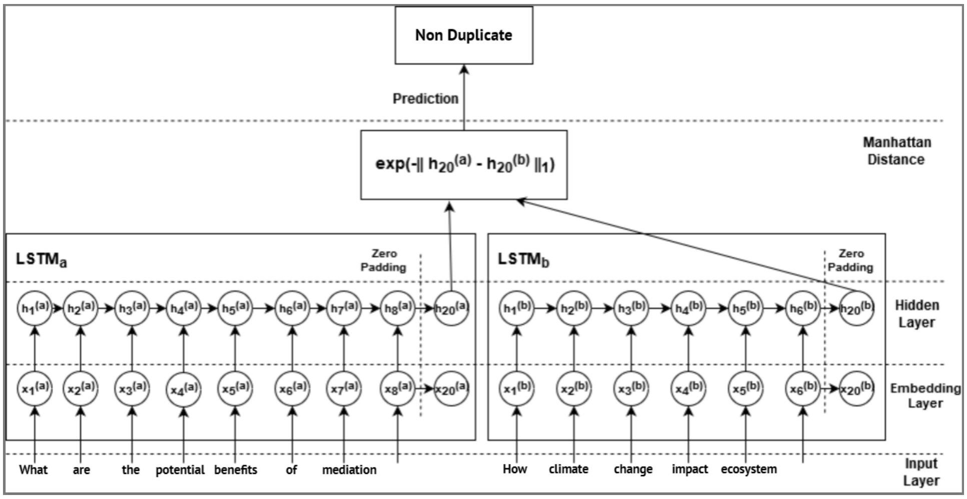

Non Duplicate
Prediction
Manhattan Distance
$\exp(-\| h_{20}^{(a)} - h_{20}^{(b)} \|_1)$
$LSTM_a$
Zero Padding
$LSTM_b$
Zero Padding
$h_1^{(a)}$ $h_2^{(a)}$ $h_3^{(a)}$ $h_4^{(a)}$ $h_5^{(a)}$ $h_6^{(a)}$ $h_7^{(a)}$ $h_8^{(a)}$ $h_{20}^{(a)}$
$h_1^{(b)}$ $h_2^{(b)}$ $h_3^{(b)}$ $h_4^{(b)}$ $h_5^{(b)}$ $h_6^{(b)}$ $h_{20}^{(b)}$
Hidden Layer
$x_1^{(a)}$ $x_2^{(a)}$ $x_3^{(a)}$ $x_4^{(a)}$ $x_5^{(a)}$ $x_6^{(a)}$ $x_7^{(a)}$ $x_8^{(a)}$ $x_{20}^{(a)}$
$x_1^{(b)}$ $x_2^{(b)}$ $x_3^{(b)}$ $x_4^{(b)}$ $x_5^{(b)}$ $x_6^{(b)}$ $x_{20}^{(b)}$
Embedding Layer
What are the potential benefits of mediation
How climate change impact ecosystem
Input Layer

**FIGURE 2.** Working explanation of siamese MaLSTM model.

**TABLE 5.** Performance comparison of the proposed model using all word embeddings.

| Model | Word Embedding | Accuracy | Precision | Recall | F1 score |
|---|---|---|---|---|---|
| Siamese LSTM | GoogleNewsVector | 81.77% | 78.77% | 69.93% | 74.09% |
| Siamese LSTM | FastText crawl | 82.77% | 79.20% | 70.58% | 74.64% |
| Siamese LSTM | FastText crawl-subword | 82.57% | 78.26% | 70.29% | 74.04% |
| Siamese LSTM | BERT | 90.45% | 87.67% | 89.45% | 88.86% |
| Siamese LSTM | RoBERTa | 91.42% | 89.52% | 90.37% | 89.41% |
| Siamese LSTM | ELMO | 95.68% | 93.12% | 95.42% | 94.27% |

were duplicates and which were not. The question pair is categorized as duplicate if the final Manhattan distance value is greater than 0.5; otherwise not.

Table 4 showcases all the hyperparameters utilized for the design of siamese MaLSTM learning models.

## IV. RESULTS AND DISCUSSION

Experimental setup, results, and performance analysis of MaLSTM are presented in this section.

### A. EXPERIMENTAL SETUP

The Siamese-LSTM model receives initial training on each of the word embeddings (GoogleNewsVector, FastText, FastText subword, BERT, RoBERTa, and ELMO) in the most recent series of experiments. Subsequently, the predictions from these separately trained models are merged to generate the final prediction. Following the evaluation of the models on 303K samples, they are further tested on an additional 100K instances. The training is conducted on a machine with 32 GB DDR4 RAM, and 2GB Dell PowerEdge T430 GPU with a frequency of 2.4GHz on two Intel Xeon processors. For running the epochs, on the "Quora Question Pair Dataset" for every embedding and displaying the result of classification, the time taken by the training was 1.5 hours.

### B. RESULTS

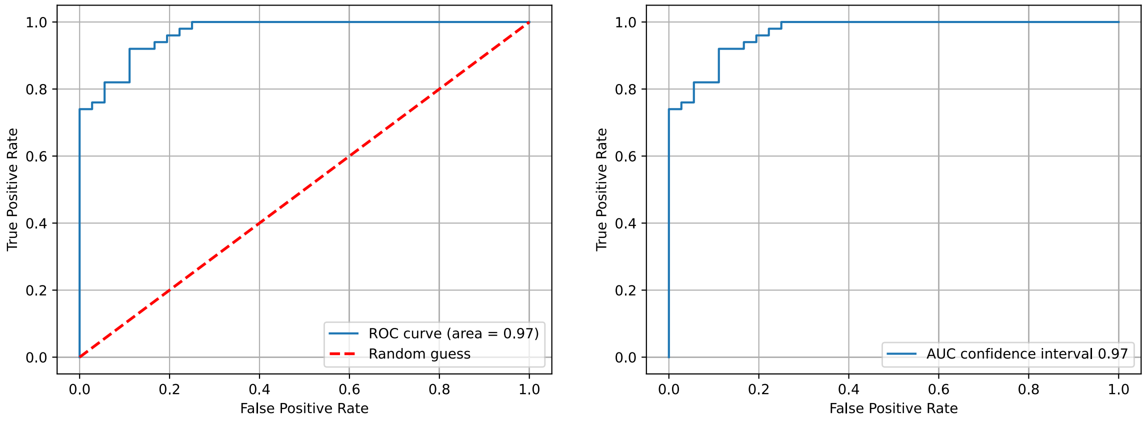
Results given in Table 5 show that by combining Siamese MaLSTM with ELMO word representation, the model was able to achieve an accuracy of 95.68%. By analyzing the predicted results of these different word embeddings, a good accuracy of 91.42% is attained by RoBERTa but still, it is less than the 95.68% obtained from the ELMO. This accuracy surpasses other state-of-the-art word representation techniques like BERT and Fasttext. Figure 3 shows the ROC-AUC curve for the proposed model showing the superior performance of the model regarding true predictions.

**TABLE 6.** Results using all word embeddings and MaLSTM model with 20 test samples.

| Model | Word Embedding | Accuracy |
|---|---|---|
| Siamese LSTM | GoogleNewsVector | 80.0% |
| Siamese LSTM | FastText crawl | 90.0% |
| Siamese LSTM | FastText Crawl-subword | 90.0% |
| Siamese LSTM | BERT | 95.0% |
| Siamese LSTM | RoBERTa | 95.0% |
| Siamese LSTM | ELMO | 100.00% |

To learn more about the class prediction process, we built models using 20 test samples from the test data. Table 6 displays the dependability of our findings. It shows that the MsLASTM model shows a 100% accuracy when 20 test samples are tested using eLMO word embedding which is much better compared to other embedding approaches like GoogleNewsVector, FastText crawl, BERT, and RoBERTa.

Each of the aforementioned cases revealed the anticipated outcomes. 16 out of 20 records are correctly predicted when GoogleNewsVector and Siamese LSTM are combined, as shown in Table 7. The number of word-vector tokens in the GoogleNewsvector, which is relatively less than the number



(a) ROC Curve.

(b) AUC Curve.

**FIGURE 3. ROC-AUC curve of the proposed model.**

**TABLE 7. Results of 20 samples for all embedding approaches.**

| Sr.# | Question 1 | Question 2 | Actual Label | Google News | FastText | FastText sub-word | BERT | RoBERTa | ELMO |
|---|---|---|---|---|---|---|---|---|---|
| 1 | How do I make friends. | How to make friends? | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | Is Career Launcher good for RBI Grade B preparation? | How is career launcher online program for RBI Grade B? | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | Will a Blu Ray play on a regular DVD player? If so, how? | How can you play a Blu Ray DVD on a regular DVD player? | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | Nd she is always sad? | Aerodynamically what happens when propellor rotates? | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | What is the best/most memorable thing you've ever eaten and why? | What is the most delicious dish you've ever eaten and why? | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 6 | How GST affects the CAs and tax officers? | Why can't I do my homework? | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | How difficult is it get into RSI? | Do you apply for programs like RSI when you're a rising senior? | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | Who is israil friend? | Is my boyfriend lying, he secretly attracted to her? | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | What are some good rap songs to dance to? | What are some of the best rap songs? | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | How do I prevent breast cancer? | Is breast cancer preventable? | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 11 | How can I make money through the Internet? | What are some different ways to make money online,? | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | What is purpose of life? | What's the purpose of life? What is life actually about? | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 13 | What is a profitable way to trade binary options? | What is binary options trading? | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 14 | How can I learn computer security? | How can I get started to learn information security? | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 15 | How do I earn from Quora? | Can I earn money on Quora? | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 16 | What is my puk code? | What's the PUK for TF64SIMC4? | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 17 | How can I find job in Japan? | How can I find an IT job in Japan? | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | How do you get a book published? | What are the good ways to write and publish a book? | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | Who is disrupting Bloomberg? | Who will disrupt Bloomberg? | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 20 | Why can flash run so fast? | The Flash (DC character): How fast can the Flash run a mile? | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

in the other two embeddings used in this paper, is about 3 billion. Additionally, the word vectors only relate to the domain of news. The Quora QnA dataset utilized in this study includes entries from various fields. Because of this, the model developed using this word embedding is incapable of recognizing questions from other domains.

As shown in Table 7, when FastText crawls with Siamese MaLSTM is tested, 18 out of 20 records are correctly predicted. More word-vector tokens (600 billion) are available in the FastText crawl than in the GoogleNewsVector embedding. Because it uses word vectors from numerous domains, it was able to predict two more records with greater accuracy, leading to better results. The final two records that are incorrectly predicted are the fifth and twelfth question pairs. After the stop words are removed from the fifth question, the remaining features are very dissimilar.

Last but not least, Table 7 shows that using Siamese LSTM with FastText crawl sub-words, 18 out of 20 records are correctly predicted. Fast Text crawl sub-words have 600 billion word vectors from various domains, similar to FastText crawl. But it also takes into account the supporting words that each word has. Additionally, this word embedding is unable to distinguish between question pairs in the fifth and fourteenth questions. It is well known that each word in the FastText subword has an n-gram, which results in a richer word2vec dictionary. The word root forms can occasionally change as a result of n-gram breaking, making it difficult to accurately determine the semantic resemblance between two sentences. When we combined the MaLSTM with ELMO word embedding techniques, the accuracy score is 100%. According to Table 7, it accurately predicts 20 out of 20 records. The ELMO representation takes good care of synonyms and root forms to achieve these results. The other distinguishing feature of ELMO representation is that it takes good care of polysemy. Let's consider an example for better understanding: Consider the following set of sentences:

1. *Yesterday, I engaged in reading the book.*
2. *Are you capable of reading the letter at this moment?*

Take a moment to reflect on the distinction between these two sentences. The verb "read" in the first sentence is expressed in the past tense, while in the second sentence, it is presented in the present tense. This exemplifies Polysemy, where a word possesses various meanings or senses.

### C. LIMITATIONS OF PROPOSED FRAMEWORK

ELMo word representation has several limitations that should be considered:

- Lack of interpretability: ELMo embeddings are derived from complex language models, making it challenging to interpret the specific factors that contribute to the embedding. Understanding the reasoning behind the representations can be difficult, limiting their interpretability.
- Computationally expensive: ELMo models are computationally expensive and require significant resources for training and inference. This can pose challenges for applications with limited computational power or real-time constraints.
- Limited transferability: While ELMo word representations are pre-trained on large-scale datasets, their transferability to different tasks or domains may vary. Fine-tuning or additional adaptation may be necessary to achieve optimal performance in specific applications.
- Large memory footprint: ELMo models can have a large memory footprint, especially when multiple layers or contextual representations are used. This can limit their deployment on resource-constrained devices or systems.

Siamese MaLSTM models, similar to other models, have certain limitations that should be considered as well.

- Training data requirements: Siamese MaLSTM models require a substantial amount of labeled training data with pairs of similar and dissimilar examples to learn effective similarity measures. Obtaining large and diverse labeled datasets can be challenging in certain domains or for specific applications.
- Computational complexity: Siamese MaLSTM models can be computationally expensive, especially during training and inference. The use of LSTM networks adds computational overhead, requiring powerful hardware and longer training times.
- Difficulty in handling long sequences: LSTMs, including MaLSTM models, can struggle with processing very long input sequences due to memory limitations and vanishing gradient issues. In such cases, the model's ability to capture long-range dependencies may be compromised.
- Difficulty in handling imbalanced datasets: Siamese MaLSTM models may struggle with imbalanced datasets, where the number of similar and dissimilar pairs is significantly different. This can affect the model's ability to learn accurate similarity measures and result in biased performance.

**TABLE 8. Results of comparison with state-of-the-art models.**

| Reference | Accuracy |
|---|---|
| [4] | 92.34% |
| [5] | 93.50% |
| [45] | 88.80% |
| Proposed model | 95.65% |

### D. PERFORMANCE COMPARISON WITH STATE-OF-THE-ART MODELS

For performance comparison of the proposed framework, several works are considered that worked on the same task of finding similarity between pair questions. For a fair comparison, only those works are selected that utilized the dataset used in this study. Table 8 shows the performance comparison results indicating the superior performance of the proposed framework compared to existing studies on the same dataset. The notable difference in accuracy, with the proposed model achieving 95.65%, compared to the highest accuracy of 93.50% in previous studies by [5] can be attributed to several key factors. The proposed framework introduces an architectural design that capitalizes on the strengths of deep learning techniques by combining MaLSTM with the ELMO word embedding technique. The engineered features through word embedding play a pivotal role in improving the model's ability to extract relevant information from question pairs and distinguish between similar and dissimilar questions. These factors collectively contribute to the exceptional accuracy achieved by the proposed model.

## V. CONCLUSION AND FUTURE WORK

This study proposed a model that, when combined with ELMO word embedding achieves a much higher level of accuracy in identifying duplicate question pairs. This study proposes a Siamese MaLSTM model that incorporates Manhattan distance for measuring the semantic similarity of questions. The model demonstrates an impressive accuracy of 100% when tested on 20 test samples of data and 95.68% when tested on 100,000 samples of test data. This surpasses the performance of other word embedding approaches utilized in this research work. A close examination of the Manhattan values reveals that they classify the pair of questions more accurately with ELMO word embedding than other combinations. The values generated by the non-duplicate pairs are closer to zero whereas values are close to 1 which are generated for the duplicate question pair. In the future, we plan to analyze a bigger dataset and experiment with hybrid neural networks that have attention layers. We will also explore various techniques for measuring similarity.

## REFERENCES

[1] S. Iyer, N. Dandekar, K. Csernai, and S. Amer-Yahia, "First Quora dataset release: Question pairs," Tech. Rep., 2017.

[2] W. Y. Wang, "'I know the answer; I just don't know what the question is!' An examination of duplicate questions on stack exchange," in *Proc. ACM Conf. Inf. Knowl. Manag.*, 2017, pp. 1389–1398.

[3] A. Singh and M. Kaur, "A review on duplicate question detection using machine learning approaches," in *Proc. IEEE 8th Int. Conf. Rel., INFOCOM Technol. Optim. (ICRITO)*. Noida, India: Amity Univ., Jun. 2020, pp. 1080–1084.

[4] Y. Liu, G. Xiang, D. Song, J. Zhang, and Y. Yu, "Deep matching models for question answering and duplicate question detection," *Inf. Sci.*, vol. 493, pp. 355–366, 2019.

[5] Y. Zhang, Q. Lu, and H. Zhu, "Deep learning based duplicate question detection on Quora," *Exp. Syst. Appl.*, vol. 137, pp. 15–25, 2019.

[6] S. AbdelRahman and C. Blake, "Sbdlrhmn: A rule-based human interpretation system for semantic textual similarity task," in *Proc. 1st Joint Conf. Lexical Comput. Semantics-Volume 1, Main Conf. Shared Task, Volume 2, 6th Int. Workshop Semantic Eval.*, 2012, pp. 536–542.

[7] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre, "Semeval-2012 task 6: A pilot on semantic textual similarity," in *Proc. 1st Joint Conf. Lexical Comput. Semantics-Volume 1, Main Conf. Shared Task, Volume 2, 6th Int. Workshop Semantic Eval.*, 2012, pp. 385–393.

[8] T. Dao and T. Simpson. (2019). *Measuring Similarity Between Sentences*. Accessed: 15, Sep. 2019. [Online]. Available: https://www.codeproject.com/Articles/11835/WordNet-based-semantic-similarity-measurement

[9] W. Zhu, T. Yao, J. Ni, B. Wei, and Z. Lu, "Dependency-based Siamese long short-term memory network for learning sentence representations," *PLoS ONE*, vol. 13, no. 3, Mar. 2018, Art. no. e0193919, doi: 10.1371/journal.pone.0193919.

[10] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," 2017, *arXiv:1702.03814*.

[11] B. N. Patro, V. K. Kurmi, S. Kumar, and V. P. Namboodiri, "Learning semantic sentence embeddings using pair-wise discriminator," 2018, *arXiv:1806.00807*.

[12] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2016, pp. 2786–2792. [Online]. Available: http://dl.acm.org/citation.cfm?id=3016100.3016291

[13] M. Tsubaki, K. Duh, M. Shimbo, and Y. Matsumoto, "Non-linear similarity learning for compositionality," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2828–2834. [Online]. Available: http://dl.acm.org/citation.cfm?id=3016100.3016297

[14] B. Rychalska, K. Pakulska, K. Chodorowska, W. Walczak, and P. Andruszkiewicz, "Samsung Poland NLP team at SemEval-2016 task 1: Necessity for diversity; Combining recursive autoencoders, WordNet and ensemble methods to measure semantic similarity," in *Proc. 10th Int. Workshop Semantic Eval.* San Diego, CA, USA: Association for Computational Linguistics, 2016, pp. 602–608. [Online]. Available: https://www.aclweb.org/anthology/S16-1091

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013, *arXiv:1310.4546*.

[16] P. Sravanthi and B. Srinivasu, "Semantic similarity between sentences," *Int. Res. J. Eng. Technol.*, vol. 4, no. 1, pp. 156–161, 2017.

[17] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 1422–1432. [Online]. Available: https://www.aclweb.org/anthology/D15-1167

[18] R. Johnson and T. Zhang, "Supervised and semi-supervised text categorization using LSTM for region embeddings," in *Proc. ICML*, 2016, pp. 526–534.

[19] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," 2015, *1511.08630*.

[20] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.* Beijing, China: Association for Computational Linguistics, 2015, pp. 1556–1566. [Online]. Available: https://www.aclweb.org/anthology/P15-1150

[21] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors," 2015, *arXiv:1506.06726*.

[22] L. Yu, K. Moritz Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," 2014, *arXiv:1412.1632*.

[23] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1746–1751. [Online]. Available: https://www.aclweb.org/anthology/D14-1181

[24] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1953048.2078186

[25] H. He, K. Gimpel, and J. Lin, "Multi-perspective sentence similarity modeling with convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1576–1586. [Online]. Available: https://www.aclweb.org/anthology/D15-1181

[26] C.-H. Shih, B.-C. Yan, S.-H. Liu, and B. Chen, "Investigating Siamese LSTM networks for text categorization," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2017, pp. 641–646.

[27] Y. Homma, S. Sy, and C. Yeh, "Detecting duplicate questions with deep learning," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS*, 2016, pp. 25964–25975.

[28] K. Abishek, B. R. Hariharan, and C. Valliyammai, *An Enhanced Deep Learning Model for Duplicate Question Pairs Recognition*. Berlin, Germany: Springer, 2019.

[29] K. Du, X. Zhang, C. Gao, R. Zhu, Q. Nong, X. Yang, and C. Yin, "GIMM: A graph convolutional network-based paraphrase identification model to detecting duplicate questions in QA communities," *Multimedia Tools Appl.*, pp. 1–28, Sep. 2023, doi: 10.1007/s11042-023-16592-3.

[30] M. B. Baby, B. Ankhari, M. Shajalal, M. Atabuzzaman, F. Rabbi, and M. I. Afjal, "Identifying duplicate questions leveraging recurrent neural network," in *Proc. 4th Int. Conf. Trends Comput. Cognit. Eng.* Singapore: Springer, 2023, pp. 331–341.

[31] M. Chandra, A. Rodrigues, and J. George, "An enhanced deep learning model for duplicate question detection on Quora question pairs using Siamese LSTM," in *Proc. IEEE Int. Conf. Distrib. Comput. Electr. Circuits Electron. (ICDCECE)*, Apr. 2022, pp. 1–5.

[32] G. V. R. Priyanka, T. Anuradha, and N. Malladi, "Duplicate Quora questions pair detection using Siamese BERT and Ma-LSTM," in *Proc. 3rd Int. Conf. Adv. Comput., Commun., Embedded Secure Syst. (ACCESS)*, May 2023, pp. 192–196.

[33] W. Gao, B. Yang, Y. Xiao, P. Zeng, X. Hu, and X. Zhu, "Duplicate question detection in community-based platforms via interaction networks," *Multimedia Tools Appl.*, vol. 83, no. 4, pp. 10881–10898, Jan. 2024.

[34] M. Chowdhary, S. Goyal, M. Mohania, and V. Goyal, "Unsupervised question duplicate and related questions detection in e-learning platforms," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, Feb. 2023, pp. 1200–1203.

[35] S. I. N. Dandekar and K. Csernai. (2017). *First Quora Dataset Release: Question Pairs*. Accessed: 15, Sep. 2019. [Online]. Available: https://data.quora.com/FirstQuora-DatasetRelease-Question-Pairs

[36] Quora. (2017). *Quora Question Pairs, Version 1*. Accessed: Sep. 20, 2019. [Online]. Available: https://www.kaggle.com/c/quora-questionpairs/data

[37] M. Mihaltz. (2016). *Word2vec-GoogleNews-Vectors*. Accessed: Sep. 20, 2019. [Online]. Available: https://github.com/mmihaltz/word2vecGoogleNews-vectors

[38] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models," 2016, *arXiv:1612.03651*.

[39] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2016, *arXiv:1607.04606*.

[40] P. Rajapaksha, R. Farahbakhsh, and N. Crespi, "BERT, XLNet or RoBERTa: The best transfer learning model to detect clickbaits," *IEEE Access*, vol. 9, pp. 154704–154716, 2021.

[41] Z. Huang and W. Zhao, "Combination of ELMo representation and CNN approaches to enhance service discovery," *IEEE Access*, vol. 8, pp. 130782–130796, 2020.

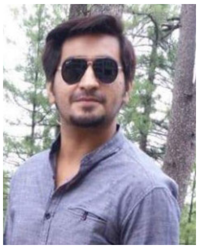
[42] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," in *Proc. 6th Int. Conf. Neural Inf. Process. Syst.* San Francisco, CA, USA: Morgan Kaufmann, 1993, pp. 737–744. [Online]. Available: http://dl.acm.org/citation.cfm?id=2987189.2987282

[43] P. Premtoon. (2017). *What are Siamese Neural Networks, What Applications are They Good for, and Why?* Accessed: 15, Sep. 2019. [Online]. Available: https://www.quora.com/What-are-Siamese-neural-networks-what-applications-are-they-good-for-and-why

[44] K. Greff, R. Kumar Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," 2015, *arXiv:1503.04069*.

[45] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 670–680.

**ABDULAZIZ ALTAMIMI** received the master's degree in computer systems security from the University of South Wales, U.K., and the Ph.D. degree in cyber security and digital forensics from the University of Plymouth, U.K. He is currently with the Department of Computer Science and Engineering, University of Hafr Al Batin, Hafar Al Batin, Saudi Arabia. He is also a Former Assistant Professor in cyber security and digital forensics for part-time bachelor's and master's students with the University of Hail, Saudi Arabia, where he is also a Supervisor of the master's student's thesis on cybersecurity and digital forensics. He is also an Assistant Professor with the College of Computer Science and Engineering, University of Hafr Al-Batin, and a Consultant with the University of Hafr Al-Batin for Social Affairs. He is also an Assistant Professor of Cyber Security and Digital Forensics for part-time post-graduate students with Imam Muhammad Ibn Saud University, Riyadh, Saudi Arabia. He has published several research articles in cyber security and digital forensics. He is a Founding Member of the Scientific Research Association.



**MUHAMMAD UMER** received the B.S. and M.S. degrees from the Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology (KFUEIT), Pakistan, in October 2020, and the Ph.D. degree in computer science from KFUEIT, in February 2024. He was a Research Assistant with the Fareed Computing and Research Center, KFUEIT. Currently, he is the Head of the Software Engineering Department, The Islamia University of Bahawalpur, Pakistan. His recent research interests include data mining, mainly working with the machine learning and deep learning-based IoT, text mining, and computer vision tasks.

**DANIAL HANIF** received the M.S. degree in CS from The Islamia University of Bahawalpur, Pakistan. After the M.S. degree, he joined his alma mater. Before this, he was working as a Web Developer with PySofts & BeCreative Group Lahore. He is currently a Lecturer with the Department of Computer Science and IT, The Islamia University of Bahawalpur. His research interest include software testing, cyber security, and digital forensics.

**SHTWAI ALSUBAI** received the Ph.D. degree in computer science from Manchester University, London. He is currently an Assistant Professor with the Computer Engineering Department, Prince Sattam bin Abdulaziz University, Saudi Arabia. His research interests include computer vision, optimization techniques, and performance enhancement.

**TAI-HOON KIM** received the M.S. and Ph.D. degrees in electrics, electronics, and computer engineering from Sungkyunkwan University, Seoul, South Korea, and the Ph.D. degree in information science from the University of Tasmania, Hobart, Australia, in December 2011. He is currently a Professor with Chonnam National University, Gwangju, South Korea. His research interests include statistical analysis, image processing, system design, XML, XML query processing, XML query optimization, machine learning, and natural language processing.

**IMRAN ASHRAF** received the M.S. degree (Hons.) in computer science from the Blekinge Institute of Technology, Karlskrona, Sweden, in 2010, and the Ph.D. degree in information and communication engineering from Yeungnam University, Gyeongsan, South Korea, in 2018. He was a Postdoctoral Fellow with Yeungnam University, where he is currently an Assistant Professor with the Information and Communication Engineering Department. His research interests include positioning using next generation networks, communication in 5G and beyond, location-based services in wireless communication, smart sensors (LIDAR) for smart cars, and data analytics.