Journal of Innovative Technology, Vol. 7, No. 2, pp. 55-64, September 2025
http://doi.org/10.29424/JIT.202509_7(2).0006

Enhanced Siamese MaLSTM with ELMo: Incorporating Squared Euclidean Distance and Feature Engineering

Munazza Nida [1], Khurram Khan [2], Zahid Halim [3*]

ABSTRACT

Identifying duplicate sentences remains a significant challenge in NLP, which is utilised in question-answering and paraphrase detection systems. One such platform is Quora, where users can post questions and answers. Due to the large number of users, it is commonly seen that most of the inquiries that people post are the same. This makes it challenging to ask and answer the same question multiple times in distinct ways. High-quality answers can be obtained by identifying such repeated requests, which could improve the user experience. One of the already existing approaches, which has employed the Siamese MaLSTM Model and ELMo Word Embedding for Quora Questions Detection, utilized the Manhattan Distance for sentence similarity measurement in the Quora Question pairs dataset available on Kaggle. In this paper, we have proposed an enhancement model by incorporating Squared Euclidean Distance alongside Manhattan Distance. Feature engineering is also used to generate additional features, such as sentence length difference and cosine similarity between ELMo embeddings. In addition, a few preprocessing techniques are also applied to improve the effectiveness of data samples. Due to computational constraints, we utilized a subset of the dataset, and the findings showed that the proposed model outperformed the existing one by 2%. Hence, the suggested model has made a substantial contribution to the detection of duplicate questions. For comparison, we have used multiple transformer-based models from HuggingFace.

Keywords: ELMO, Quora Question Pairs, Squared Euclidean distance, Feature engineering, Duplicate Question Detection.

1. INTRODUCTION

Identifying question pairs that are identical word-for-word is relatively straightforward since it relies on direct word comparisons. However, detecting semantically similar question pairs is far more complex because it requires a deep understanding of meaning and context (Dammu∗ & Alonso, 2024). In the past, platforms like Yahoo Answers and Google Answers attempted to provide question-answering services but struggled to gain popularity due to their inability to maintain high-quality content. The presence of excessive low-value content contributed to their decline. In contrast, Quora is recognized for its commitment to ensuring content quality. One notable effort by Quora is its initiative to detect and manage duplicate question pairs on its platform (Wang, Gill, Mohanlal, Zheng, & Zhao, 2013). One such contribution of Quora is its effort to identify duplicate question pairs posted on its platform. Identification of duplicate questions on the Quora platform has gained significant importance, as this has a direct impact on user experience. Repeated redundant content makes the platform cluttered with unnecessary content. Quora is one of the popular question-answer platforms. Such platforms are used by people to ask any query of their interest and experts from the relevant domain try to answer the query. It happens quite often that users post similar types of questions on multiple pages, which results in redundant content. Experts must also have to post answers separately on separate pages for the same context. In this scenario, finding the best answer becomes difficult, which results in a poor user experience. With the purpose of redundancy reduction and increasing user engagement, this platform launched a competition in 2017 on Kaggle, and many people participated. Since then, this area of research has grabbed the interest of many researchers (Xu & Yuan, 2020).

Various Natural language processing, machine learning, and deep learning techniques have been used to develop algorithms for sentence similarity measurement (Farouk, 2019). This task is not that easy, because natural languages are semantically complex, and the same question can be posed in multiple ways. The approach utilizing a Siamese MaLSTM model with ELMo word embeddings has demonstrated significant performance in detecting duplicate questions on Quora. In this research, we build upon this method, where the Manhattan distance was employed as a similarity metric between sentences (Altamimi, et al., 2024). We have implemented the model proposed in the referenced study and incorporated an additional distance metric, namely the Squared Euclidean distance. Furthermore, we have focused on improving preprocessing by carefully analyzing the dataset structure. The proposed model also used feature engineering techniques to introduce two new features, which are the cosine similarity between Elmo embeddings of both questions and the sentence length difference between both questions. Later, we compared the performance of the proposed

Manuscript received May 20, 2025; revised July 28, 2025; accepted September 4, 2025.

[1] Lecturer, FAST National University of computer and emerging sciences, Islamabad, Pakistan

[2] Assistant Professor, GIK Institute of Engineering Sciences and Technology, Topi, Pakistan

[3*] Professor (corresponding author), National Yunlin University of Science and Technology, Taiwan (email: zahidh@yuntech.edu.tw).

model with various state-of-the-art, transformer-based Hugging Face models (Wolf, et al., 2020). This paper presents the literature review in Section II, Section III gives a detailed explanation of the Dataset and methodology applied to the dataset, attributes of the dataset used, and the proposed methodology. Details of the experiment and results analysis are discussed in Section IV. Section V is about discussion. Section VI gives the concluding remarks and Future work on the proposed methodology.

2. LITERATURE REVIEW

Measuring semantic similarity between sentences is a vital component in tasks that are particularly designed for recognizing rephrased questions or paraphrase identification, etc. For that reason, identifying contextual similarities is essential. Identification of duplicate or similar questions is not a new problem. This task has been researched for so long, as it has overlapping roots with other NLP problems like plagiarism detection, paraphrase detection, etc. However, it is still considered one of the complex problems of NLP. Making duplicate question detection automation systems for such questions and answering platforms requires model training. For supervised training, a fairly large amount of labelled data is needed, which is tedious and costly. To tackle this issue, researchers moved their attention towards a different approach, which was the idea of weak supervision with question-answer pairs, semi-supervised training (Uva, Bonadiman, & Moschitti, 2018) and adversarial domain transfer (Shah, Lei, Moschitti, Romeo, & Nakov, 2018). But these models have their own drawbacks; they still need some labelled data (Rücklé & Moosavi, 2019). That is why the proposed solutions couldn't contribute much to the need for supervised training data, and the problem was still there. Another way of looking at the problem is designing an efficient neural network that could effectively detect paraphrased versions of sentences.

This problem was addressed using the paraphrase detection approach because, in duplicate questions, one question is a paraphrased version of its duplicate (Zhu, Yao, Ni, Wei, & Lu, 2018). Another technique which is called Natural Language Sentence Matching (NLSM), tries to figure out differently written similar sentences (Wang, Hamza, & Florian, 2017). One of the initial and most common techniques was the use of the SVM model, which used BoW for word embeddings (Patro, Kurmi, & Kumar, 2018). Apart from conventional techniques, deep learning techniques are widely used for this problem and have observed significant performance, particularly in sentence semantic analysis and similarity detection (Mueller & Thyagarajan, 2016).

Gradually, research in this domain moved its dimensions toward using pre-trained word embeddings, which was a better approach with significantly better results. Pre-trained word embeddings proved good in capturing the semantic similarity of sentences (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). There exist many pre-trained deep learning models that generate word embeddings for text. These embeddings are vectors that preserve the semantic meaning of the text. A few of the most popular word embeddings are Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013), GloVe (Pennington, Socher, & Manning, 2014), and FastText (Joulin, Grave, Bojanowski, & Mikolov, 2016). All these word embeddings are called static word embeddings because they have a fixed vector representation for each word in the whole document, no matter what context these words are being used. These kinds of embeddings are computationally less costly and are better for word-level similarity checking-related tasks. Because such embeddings struggle with differentiating between words with different meanings depending on their context. For example, riverbank and bank account are entirely different contexts for the word "Bank". Therefore, for better semantic meaning capturing, we need context-specific word embeddings in which the vector for the word is made, depending on its use in the sentence. Elmo embeddings are one of them. Elmo, which stands for Embeddings from Language Models (Peters, et al., 2018) is also one of the pre-trained models for word embeddings. The architectural structure for the Elmo model consists of multiple layers of LSTM. Each layer contributes to capturing the context-dependent nature of the words, and that's how different vector representations for the same word are generated when it is used in different contexts.

For sentence analysis, there is, in fact, sequential data processing going on where the current word is influenced by the word before it and the word after it. This is what we call context-based analysis. There is a class of neural networks specifically designed for sequential data processing, known as recurrent neural networks, or RNNs. Unlike feedforward networks, RNN works on sequences and time series data. There is a concept, known as backpropagation through time or BPTT (Werbos, 1990) an essential concept in RNN, as languages express themselves as temporal sequences (ELMA, 1990) BPTT is a variant of standard backpropagation.

In natural language processing and artificial intelligence, there is an open-source platform called Hugging Face (Wolf, et al., 2020). This platform provides access to thousands of pre-trained models through APIs. These pretrained models are applicable to perform NLP tasks like paraphrase detection, summarization, and text classification, etc. These models are fine-tuned and built upon architectures including BERT (Devlin, Chang, & Lee, 2018), MiniLM (Wang, et al., 2020), RoBERTa (Liu, et al., 2019) and GPT (Yenduri, et al., 2023) etc. These models are pre-trained over diverse datasets. Standard interfaces are designed to experiment with these models and ensure reproducibility.

There are many state-of-the-art pre-trained transformer models implemented by the Hugging Face Transformers library. These transformer models are quite popular in the field of natural language processing and are often used for comparative analysis. Accessing Hugging Face libraries is straightforward API calling and is applicable to perform duplicate detection tasks and compare similarity in sentences. Therefore, we have used a few of these libraries for comparative analysis of the proposed study. Starting from a basic BERT model named bert-base-uncased, which is pretrained over uncased English language text, moving to stsb-roberta-base. stsb-roberta-base is a cross-encoder's Semantic Textual Similarity Benchmark (STSB) dataset based on the RoBERTa model. This model generates similarity scores between sentences after simultaneously analyzing both sentences.

Other than a cross-encoder, we have reviewed sentence-transformers from Hugging Face libraries and selected three of the sentence transformers. First is all-mpnet-base-v2, which uses the Siamese training mechanism along with a triplet training scheme. This architecture is based on MPNet and captures semantic similarity on the sentence level. Second and third sentence-transformers that we used are all-MiniLM-L6-v2 and paraphrase-MiniLM-L6-v2. Both of these models are pretrained to identify paraphrased versions of sentences.

We aimed to develop a hybrid model, inspired by other hybrid models (Saadat, Shah, Halim, & Anwar, 2024), which not only focuses on structural features but also extracts semantic features of the text.

In 2023, (Rasham, et al., 2023) published an innovative approach with the title DBpedia. The purpose of DBpedia is to systematically formalize semantic data in a structured way for Urdu

language. It was basically an organized database for Urdu content. Researchers tried to facilitate the use of the Urdu language in web applications by mapping almost 1,000 Urdu attributes to English language-based corresponding terms. DBpedia used the semantic understanding of the Urdu text for accurate mapping.

Long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) is a type of RNN, which is useful in sequential data. That is why LSTM was used for capturing the semantic meaning of sentences, and it gave outstanding performance (Sravanthi & Srinivasu, 2017). LSTM is also used for measuring sentence similarity, based on its semantic meaning (Tai, Socher, & Manning, 2015). One of the methods used in (Kiros, et al., 2015), is basically an LSTM-based approach. In this approach, sentence vectorisation was done using the Skip-gram. The idea of the Skip-gram model was proposed as part of (Mikolov, Chen, Corrado, & Dean, 2013). Skip-gram model takes a word of a sentence as a target word and predicts the words around that target word, or in simple words, it predicts the context of the target word. Hence, the obtained vectors for each sentence were termed skip-thought vectors. This technique was basically used to predict the surrounding sentences of the current sentence. In (Chandra, Rodrigues, & George, 2022) a Siamese LSTM was proposed, and semantic similarity was extracted using GloVe word embedding. In Siamese architecture, multiple inputs are processed in parallel. So, question pairs are inserted simultaneously into the network (Yu, Hermann, Blunsom, & Pulman, 2014). Siamese LSTM is an LSTM architecture designed using the Siamese approach, in which a shared LSTM is used for training; the embeddings of both sentences are inserted as input to the same LSTM model, resulting in the weights learning of one question influenced by the weights of another question in the question pair (Mueller & Thyagarajan, 2016).

Focusing on Siamese-based literature, in 2024, a Siamese network was developed, which was capable of accepting more than two inputs simultaneously, introduced with the title of Multi-Siam in (Bhoi, Markhedkar, Phadke, & Agrawal, 2024). This model was developed for comparing multiple sentences.

There is a paper with the title "Siamese Recurrent Architectures for Learning Sentence Similarity" that proposed the fundamental idea of the MaLSTM architecture. The paper gave a detailed demonstration and analysis of the application of the Manhattan distance for finding sentence similarity. And concluded that Manhattan distance is better for identifying paraphrase detection than Euclidean distance or cosine similarity. That's why Manhattan distance was aggregated with LSTM as a distance metric to measure the closeness between two sentences. Another Siamese model in aggregation with CNN was used for duplicate sentence detection. As the name CNN suggests, the Siamese CNN model (He, Gimpel, & Lin, 2015) created sentence embeddings using convolution and pooling processes. Another CNN-based model, which used GloVe embeddings for the vector representation of words in sentences, achieved good accuracy. Dimensions for each vector embedding were of size 100, and those were trained over Wikipedia text (Wang, Hamza, & Florian, 2017). If we target the Quora dataset in particular, not much work has been done by the researchers so far (Shih, Yan, Liu, & Chen, 2017). It may be because natural language is complex. Spoken languages do not have fixed standard rules to follow; they change frequently. Designing a fixed-rule-based approach does not suit well for tackling the situation. Besides MaLSTM, other approaches have also been used, including Bert and BiLSTM for duplicate question detection on the Quora dataset (Gao, et al., 2024).

There is another paper that also presents a CNN-based architecture for the Stack Overflow Q&A dataset (FASEEH, et al., 2024). This paper proposed a hybrid model consisting of LSTM and CNN. The embeddings used in this model are Word2Vec and GloVe. These embeddings are static in nature, and hence they are less capable of capturing semantic meaning as compared to Elmo embeddings, which are considered contextualized embeddings. That's why it is possible that the CNN-LSTM model will be unable to identify those sentences that are semantically different and classify them as duplicate sentences. This leads to the inclusion of false positives in the prediction.

With the purpose of effectively reproducing deep learning models, a metadata standardization approach was developed by (Shaheen, et al., 2025) with the title "DeepVoc". DeepVoc is built to ensure accurate reproducibility and standardization of metadata of deep learning experiments. Like transparency in hyperparameter settings, selection of evaluation metrics, and preprocessing steps, etc. DeepVoc provides a generic vocabulary, which may need to be customized additionally for context-specific NLP problems like the Quora question pair dataset. And in order to use DeepVoc, researchers must design an extra layer of custom wrappers, as Deep learning frameworks like TensorFlow, PyTorch, or Hugging Face by default do not support integration of DeepVoc.

In natural language, there is a domain that addresses Roman Urdu; it is Urdu language text written using the English alphabet. This area is often disregarded and ignored while deep learning based natural language processing is being researched. One of the papers (Ali, et al., 2023) presented a Transformer-based multilingual model. This mBERT (finetuned multilingual BERT) model was trained on a manually annotated Roman Urdu corpus. The model focused on emotion detection in Roman Urdu. In this research paper, we have constrained our domain to English text only, and the dataset we used is English language-based. In the future, we might explore the identification of duplicate question pairs in other languages as well, including Urdu and Roman Urdu-based text.

Bao, Dong, Xu, Yang, & Qi (2024) proposed an attention-based model that uses an attention mechanism to focus on those parts of sentences that need critical focus. This model was an extension of existing MaLSTM frameworks. Therefore, the base of the model uses a Siamese network. This helps this model to detect plagiarism; hence, it can be applied for duplicate sentence detection. Those are sentences with the same semantic meaning and different words. But the presence of an attention layer in both LSTM branches could cause an increase in model complexity in comparison to the Siamese MaLSTM architecture. This model also only relied on implicit features introduced by a deep learning model, so it lacks interpretability, and no explicit feature integration is performed in the architecture.

There is another interesting Siamese network-based model, though it is based on image data for a fast and rigorous detection between high-resolution images. This model is trained to compare pixel data (remote sensing image pairs) and uses spatial locality and Convolutional spatial attention. The model presented in this paper, along with using a Siamese-based deep learning architecture, also incorporates feature engineering. This helped in capturing some of the essential features explicitly. The original SiamUNet architecture has enhanced variants, SiamFAUnet and SMDNet architecture. These enhanced architectures use Attention and multi-scale fusion. (Zhang, Xu, Wang, Shi, & Yan, 2023).

In summary, duplicate sentence identification is one of the actively researched domains. Many models and architectures have been proposed. This includes machine learning and deep learning approaches. Also, the development of various hybrid models has been explored. For text vectorization, both static and dynamic em-

beddings have been explored after a thorough literature review and analysis of potential limitations and strategies. We aimed to develop a model that uses contextualized embeddings, such as Elmo, for effectively capturing semantic duplication. Along with that, for simultaneous sentence evaluation, we chose a Siamese network as a base. For relative difference measuring, the choice of distance metric is made through multiple experiments. Furthermore, to capture those aspects of sentences that are neglected by deep learning models, we used custom feature engineering.

3. DATASET AND METHODOLOGY

For determining the similarity between sentences, a semantic understanding of their words is necessary. This involves

- the meaning of individual words and the relationship between
- words semantically. In this paper, an enhanced model for
- the Siamese MaLSTM model and Elmo embeddings (Altamimi, et al., 2024) is proposed.

The proposed model combined the base model with the following additional features:

1. The use of additional preprocessing techniques is applied to each question before vectorization.
2. The use of an additional distance metric, i.e., the Squared Euclidean distance metric.
3. Incorporating a feature engineering technique, which uses additional features like cosine similarity between each question pair and the length difference between each question pair

3.1 Dataset

Quora made a dataset publicly available in 2017, with the title Quora Question Pairs (Quora, 2012). This data is available on Kaggle (Quora). The size of the original dataset is 404,351 entries of question pairs. Due to computational constraints, only 100,000 entries are used for all experiments. Figure 1 represents a sample of the Quora Question Pairs dataset (Quora, 2012).

id	qid1	qid2	question1	question2	is_duplicate
447	895	896	What are natural numbers?	What is a least natural number?	0
1518	3037	3038	Which pizzas are the most popularly ordered pizzas on Domino's menu?	How many calories does a Dominos pizza have?	0
3272	6542	6543	How do you start a bakery?	How can one start a bakery business?	1
3362	6722	6723	Should I learn python or Java first?	If I had to choose between learning Java and Python, what should I choose to learn first?	1

Fig. 1 Quora Question Pairs Dataset Sample, Source (Quora, 2012)

The dataset has attributes id, qid1, qid2, question1, question2 and is duplicate. These features indicate the id of each record, question1 id, question2 id followed by question1 and question2, and is duplicate is a binary class attribute, respectively. The dataset was split into training and validation data such that 75% of the dataset was used for training and 25% was used for validation.

3.2 Methodology

(1)Existing Model: For comparison purposes, the already existing Model, i.e., the Siamese MaLSTM with the ELMo, was implemented. For that purpose, basic pre-processing techniques as mentioned in (Altamimi, et al., 2024) were applied to a dataset of size 100,000 samples. Details of baseline model implementation are as follows:

1. PREPROCESSING: In the preprocessing step, duplicate records are checked, and missing values are addressed. Lowercase conversion of text, stemming, tokenization, and stopword removal are done. Questions with a maximum size of 45 words were selected, and zero padding was performed on questions with several words less than 45. The rest of the records were truncated.
2. WORD EMBEDDING: Elmo embeddings are then generated for each question pair. For this purpose, pre-trained embeddings from the TensorFlow Hub were imported. The dimension for each embedding vector is 1024.
3. Siamese MaLSTM: Those generated embeddings were fed into Siamese LSTM, and Manhattan distance was used as the distance metric between both sentence vectors to classify that pair as duplicate or not. The formula for the Manhattan distance is given in (1)

$$\boldsymbol{D} = \sum_{i=1}^{n} |\boldsymbol{x_i} - \boldsymbol{y_i}| \tag{1}$$

(2) Proposed Improvements: The following three significant improvements have been made in the proposed model to more efficiently identify duplicate question pairs.

1. Additional preprocessing: Careful analysis of the dataset leads to applying more dataset-specific pre-processing techniques. This includes special characters removal and replacing them with their string equivalent, e.g., $ was replaced with a string dollar sign. Similarly, a mathematical expression \[math\] happened to be repeatedly occurring in the dataset, so it was treated by replacing it with a space character. Numbers with many zeros, like billions and trillions, were replaced with their string equivalent representation. There are words in English which are called Decontracting words, such as "ain't", which was replaced with "am not". In order to ensure consistency, preprocessing techniques from the original model remained intact, except that, in the enhanced model, we did not use any zero padding for sentences smaller than 45. It is because ultimately the embeddings of every sentence will be of the same size, i.e., 1024 Elmo embeddings, which are calculated by averaging out the Elmo embeddings of each word of that sentence.
2. Additional Distance Metric: With the purpose of searching for a better and well-suited Distance Metric, we have performed a comparative study between multiple distance metrics. For this purpose, in basic Siamese MaLSTM, different distance metrics were applied instead of the Manhattan distance. The results obtained are shown in Table 1, which clearly shows that the Squared Euclidean distance is performing better than all other distance metrics.

For better results, the Manhattan distance remained intact, and the squared distance was also calculated. The formula for Squared Euclidean distance is given in (2)

$$D_{\text{Squared Euclidean}}(x, y) = \sum_{i=1}^{n} (x_i - y_i)^2 \tag{2}$$

Feature Engineering: In the Feature Engineering technique, two features are manually calculated after applying Elmo embeddings on question pairs. These two features include Cosine Similarity between ELMo embeddings and Sentence Length Difference. These features were selected because Cosine Similarity between embeddings is widely used for measuring similarity between two vectors. Sentences with greater cosine similarity are considered semantically similar. Sentence Length Difference was also considered as an important contributing feature. It was observed that sentences with similar lengths are mostly lying in the

duplicate questions category. This obviously may not be true for some cases, but it is an important contributing feature.

The formula for cosine similarity is given in (3)

$$\text{Cosine Similarity} = \frac{\boldsymbol{A}\cdot\boldsymbol{B}}{|\boldsymbol{A}||\boldsymbol{B}|} \quad (3)$$

Table 1 Comparison of accuracy across different distance metrics

Metric	Manhattan	Euclidean	Cosine Similarity	Squared Euclidean	Chebyshev	Hamming
Accuracy	0.73584	0.62148	0.67048	0.75008	0.73084	0.64660

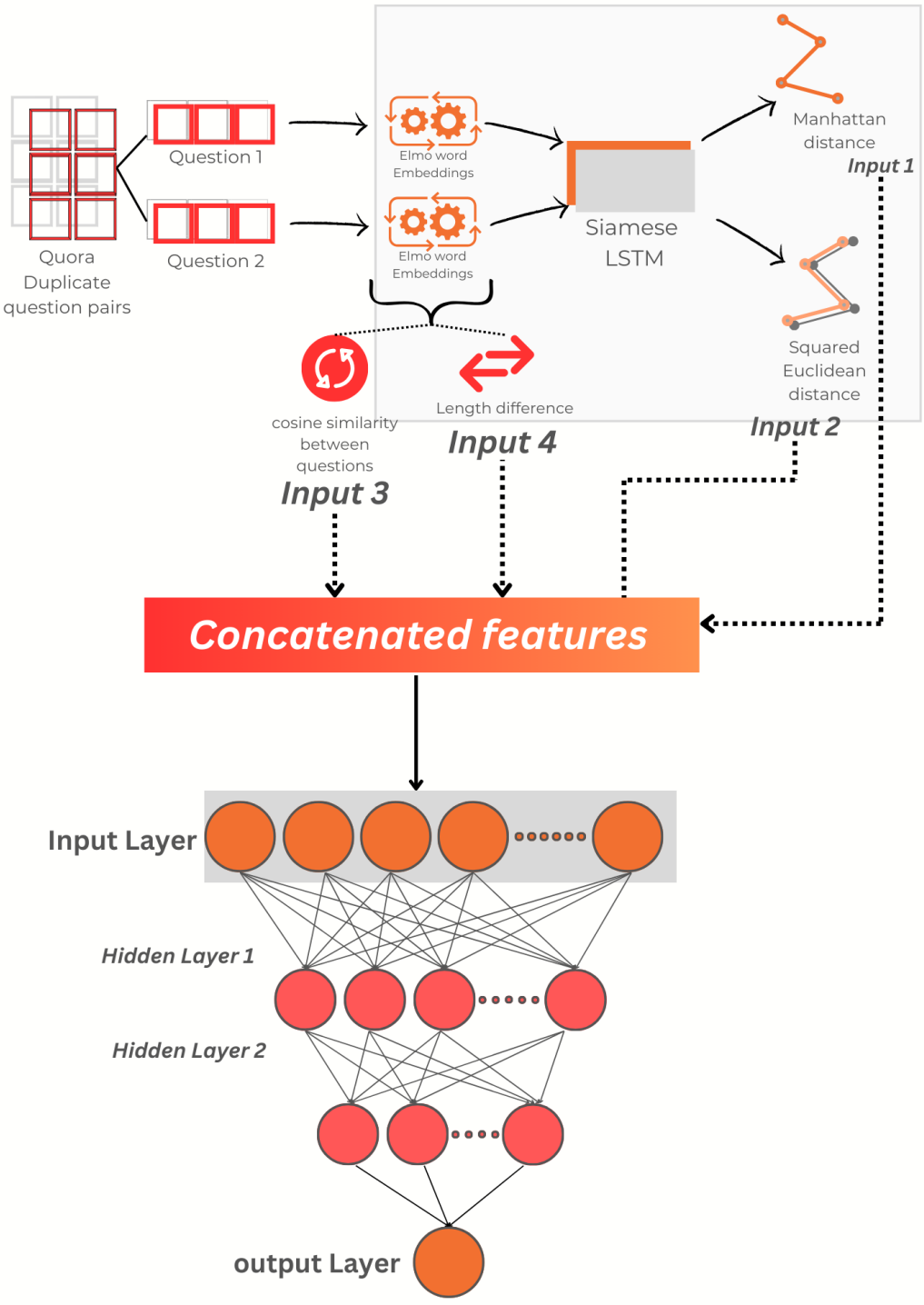

Fig. 2 Enhanced Siamese MaLSTM with ELMo

3.3 Proposed Model

We propose a model that uses existing Siamese MaLSTM architecture and ELMo word embeddings by integrating multiple distance-based similarity measures and feature engineering techniques. The overall architecture of the proposed model is illustrated in Figure 2, which clearly depicts the workflow for feature generation. The proposed approach uses a mix of features that have been learned automatically using the deep neural network of Siamese MaLSTM, along with manually selected features for better results. Question pairs from the dataset are converted into Elmo embeddings of size 1024 each, which are then processed by Siamese LSTM. The details of each component of the workflow diagram are discussed in detail in the following:

1. Sentence Embedding via ELMo: The operation on sentences starts by converting them into some vector representation or embeddings. Static word embeddings like Word2Vec, GloVe, or Fasttext, etc., have a limitation of using a single, context-independent vector representation for each word, regardless of the context in which it appears. We started our workflow by first converting sentences into Elmo word embeddings. Because ELMo provides dynamic word embeddings, assigning different vector representations to the same word based on its context makes them context-dependent rather than context-independent. In our model, each sentence is first tokenized into words. Then each word was converted into its Elmo vector representation. After that, the whole sentence's Elmo vector representation was calculated by averaging the Elmo vector representations of individual words. ELMo embeddings were incorporated from TensorFlow Hub to generate Contextual embeddings.
2. Siamese LSTM: Siamese LSTM is a deep-learning network that has the capability to process multiple inputs simultaneously. The architecture of the Siamese neural network consists of two parallel, identical models; both inputs are processed in parallel. This network operates in such a way that parameters and learned weights are shared simultaneously. Both of these identical, parallel networks are Long Short-Term Memory (LSTM) networks. The Siamese network uses a sharing approach among both parallel LSTM networks while training. The idea initially came into existence back in 1994. Such types of networks are primarily useful in scenarios where there is a need to establish a relationship between two patterns. That is the reason for the success of Siamese architecture when used for duplicate question pair identification. Combined with LSTM, it became the Siamese LSTM network, which consisted of 128 neurons. LSTM accepts the ELMo embeddings and processes the questions in a consistent manner because of the Siamese network. The output of the Siamese LSTM network is fixed-length vector representations for both questions. These are then subjected to further processing, which involves sentence similarity measurements between both questions.
3. Concatenated Features: One of the main novelties of the proposed model is incorporating feature engineering and using more than one distance metric for sentence similarity measures, including Manhattan distance and squared Euclidean distance. All these features are then calculated separately in the Siamese model to determine sentence similarity. It means there are four features in total. Two of them are learned features from Siamese MaLSTM and are derived by applying the Manhattan distance and the Squared Euclidean Distance. The other two features are manually incorporated into the network. Elmo embeddings of both sentences were also used to calculate the cosine similarity and the length difference between questions. Each feature has its own contribution and significance. As a result, till the end of this stage, we have 131-dimensional input feature vectors for each sample. In summary, the feature vector will consist of the following features:

- Manhattan Distance: (1) It is the base element of the MaLSTM model. It allows a smooth learning process of training by avoiding the vanishing gradients problem. The feature vector of length 128 is obtained from the Manhattan Distance.
- Squared Euclidean Distance: Squared Euclidean Distance () is preferred over Euclidean Distance, as it enhances the magnitude differences due to its square factor in the formula. It means that this feature helps the model to identify semantically similar sentences more accurately. This feature is also of size 1.
- Cosine Similarity: This feature is used to identify sentences that are directionally aligned. It means that it doesn't capture the similarity between sentences; it captures the angle between two vectors. Cosine Similarity (3) is applied directly on ELMo embeddings, without passing them through a Siamese LSTM network. Those sentences that portray the same meaning but use different wording can be identified by incorporating this feature. That is why it is a very useful feature. The Cosine Similarity Feature is also of size 1.
- Absolute Length Difference: It is observed that in most cases, sentences with similar semantic meaning are of almost the same length. That is the reason the length difference between sentences is used as a feature for duplicate question detection. This feature is also of size 1.

The hyperparameter details of the proposed model are given in Table 2.

Table 2 Hyperparameter settings

Hyperparameter	Value
LSTM	128
Epochs	100
Learning rate	0.001
Hidden Layer 1	64
Hidden Layer 2	32
Batch Size	32
Optimizer	Adam
Activation(hidden)	ReLu
Activation(output)	Sigmoid

Besides the hyperparameter details mentioned in Table 2, the details of the parameter settings are as follows:

- The Elmo embeddings are used from TensorFlow Hub. And sentence-level embeddings are represented by averaging out the embedding vectors of the words of a sentence.
- The proposed network consisted of an LSTM layer of 128 units for each question.
- The architecture of dense layers is designed as:
  - The first hidden layer consisted of 64 units and ReLU activation.
  - The second hidden layer consisted of 32 units and ReLU activation.
  - The output layer consisted of a single unit and Sigmoid activation.
- Training setting:
  - Adam Optimiser is selected.

- Binary cross-entropy is used as a loss function.
- 100 Epochs are performed.
- The batch size is 32.
- Setting random state 42, the 75% dataset is used as training data, and 25% is used as validation data.
- 100,000 records are evaluated from the dataset, and sentences with a maximum of 45 words are selected (as used in the base paper).

4. Dense layers/Fully connected layers: All four features are concatenated as a single feature vector and are inserted as input to the dense deep neural network, shown in Figure 2. This network consists of the input layer, two hidden layers, and an output layer. The threshold of 0.5 was used for categorization at the output layer. The number of neurons in each layer is mentioned in Table 2. The detail of each layer is discussed in the following:
   a. Input Layer: The input layer of a fully connected neural network consists of 131 neurons (size of concatenated feature vector).
   b. Hidden Layers: There are two hidden layers in our dense neural network. Both layers have used ReLU as an activation function. To minimize computational complexity, through several experiments, the most suitable number of neurons is decided for both layers. Hidden Layer 1 consists of 64 neurons, and Hidden Layer 2 consists of 32 neurons. The learning rate of 0.001 was found to be the most appropriate one. Other than that, the training process consisted of 100 epochs with a batch size of 32. Optimisation is performed using Adam Optimiser.
   c. Output Layer: At the output layer, the sigmoid activation function is used. We wanted our output to be in the form of a probability estimation. We kept the decision threshold of 0.5 at the output layer. The output layer consisted of a single neuron with a binary output.

3.4 Basis for Choosing the Proposed Method

Elmo embeddings capture deep contextual semantic meaning. The base model used the Siamese MaLSTM network, and therefore, the use of Manhattan distance preserved feature-wise absolute difference. Later, through repetitive experiments using various distance metrics and performance comparisons, it was shown that the use of the squared Euclidean distance metric is a better choice (Table 1). The reason is that the square Euclidean distance metric penalizes large mismatches. As this distance metric is often used in vector-space models, it is better at capturing the geometric spread between embeddings.

In our proposed enhanced model, though we have incorporated the squared Euclidean distance, we have still not omitted the already used Manhattan distance. Because of its own significance. It captures the linear gaps between embeddings. Hence, this approach adds diversity to the model.

Furthermore, we wanted our proposed model to capture some specific features that might not be implicitly captured by the deep learning model. So, we introduced explicit features through feature engineering techniques. One of the introduced features is Cosine similarity between questions, which calculates the Directional closeness between questions. Another feature is the Length difference between questions, based on the observation that similar questions have mostly similar length.

4. EXPERIMENTS AND RESULTS

4.1 Dataset Preparation:

The data used for this experiment is the Quora Question Pairs dataset, publicly available on Kaggle. The size of the original dataset is 404,351 entries of question pairs. But for this experiment we have used only 100,000 entries, due to computational constraints. The sample Quora Question Pairs dataset (Quora, 2012) is shown in Figure 1. The dataset is obtained after applying the preprocessing techniques mentioned in the Methodology section of this paper. This preprocessed data is then used for embedding generations and training.

The experiments were performed on the subset of the Quora question pairs dataset (Quora), on both models i.e. the base model and the enhanced model. Input data is passed through the proposed model. Features after being extracted are then fed into the neural network. A Deep Neural network, after being trained on these features, identifies duplicate or non-duplicate question pairs.

4.2 Implementation Details

The dataset was split into training and validation data such that 75% of the dataset was used for training and 25% was used for validation. Random seed is set to 42 for data splitting. The model has accessed pre-trained ELMo embeddings from the TensorFlow Hub ELMo module. Accuracy, Precision, Recall, and F1-score are used as performance evaluation metrics. The comparison of Table 3 and Table 4 clearly shows that all evaluation metric values of the baseline model are less than those of the proposed model.

4.3 Hardware and software specifications

The experiments were conducted on an HP EliteBook 820 laptop with 8GB RAM, running Windows 10 as the operating system. For GPU processing, two NVIDIA T4 GPUs (provided by Kaggle) were used, each equipped with 2560 CUDA cores and 16GB VRAM. The development environment was Kaggle Notebooks, utilizing Python 3.7 as the programming language. Various libraries and frameworks were employed, including TensorFlow, NumPy, Scikit-learn, pandas, TensorFlow Hub, NLTK, PrettyTable, and BeautifulSoup.

Table 3 Performance metrics of the baseline model

Metric	Accuracy	Precision	Recall	F1 Score
Value	0.741680	0.661022	0.651802	0.656380

Table 4 Performance metrics of Proposed model

Metric	Accuracy	Precision	Recall	F1 Score
Value	0.768701	0.698649	0.659533	0.678528

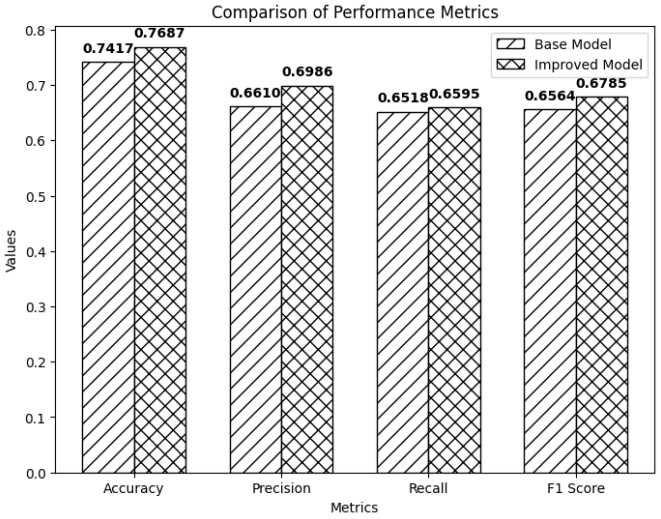

Fig. 3 Results

4.4 Results and Comparative Analysis

The results of the existing model are given in Table 3. The results obtained from the enhanced model are given in Table 4. Comparisons of Accuracy, precision, Recall, and F1-score values in Figure 3 clearly indicate that the enhanced model has performed very well.

For a detailed comparative analysis, we have used several transformer-based state-of-the-art models from Hugging Face. These models include CrossEncoder, BERT, and multiple SentenceTransformer models like all-mpnet-base-v2, all-MiniLM-L6-v2, and paraphrase-MiniLM-L6-v2. All these models are accessed using Hugging Face's API. For that purpose, transformers and sentence-transformers libraries are used. Keeping the dataset limited to 100,000 records and only keeping the records with the length of questions not more than 45 words, we made sure a fair comparison was performed. The detailed comparative analysis report can be observed in Table 5.

The analysis made from Table 5 shows that the Proposed model has efficiently distinguished between duplicate pairs of questions. The Accuracy, Precision, and F1-score have reached significantly higher values. But if we analyze recall value in particular, few models have shown exceptional performance. Their bad performance in terms of precision indicates their tendency to generate a greater number of false positive cases.

Table 5 Comparative Analysis

Model	Accuracy	Precision	Recall	F1 Score
Proposed Model	0.7687	0.6986	0.6595	0.6785
Baseline Siamese MaLSTM + ELMo (Altamimi, et al., 2024)	0.7416	0.6610	0.6518	0.6563
CrossEncoder (stsb-roberta-base)	0.7362	0.5928	0.9425	0.7278
BERT (bert-base-uncased)	0.3744	0.3743	1.0000	0.5448
SentenceTransformer (all-mpnet-base-v2)	0.6188	0.4926	0.9982	0.6596
SentenceTransformer (all-MiniLM-L6-v2)	0.5942	0.4770	0.9979	0.6454
SentenceTransformer (paraphrase-MiniLM-L6-v2)	0.5773	0.4646	0.9975	0.6339

5. DISCUSSION

The two features, Cosine similarity and squared Euclidean distance, accurately captured the semantic meaning of question pairs. Squared Euclidean Distance differentiates sentences based on their positional distance in the embedding space, which is why it is used to measure the absolute difference between two vectors in a high-dimensional space. The presence of a square in its formula helps to magnify differences between vectors. That makes it more effective in distinguishing semantically different question pairs. Hence, effectively helps in training deep learning models by providing stronger gradient updates. In the context of sentence embeddings, Cosine similarity, on the other hand, effectively captures semantic relationships between question pairs. Because it measures the angle between two sentence vectors rather than their magnitude. Sentences with similar meanings often have embeddings that point in the same direction, even if their magnitudes differ; that's why it effectively captures semantic relationships. Another feature, i.e., the difference between the size of question pairs, also proved essential.

6. CONCLUSION AND FUTURE WORK

In this paper, an enhanced model is proposed that is based on an existing model, Siamese MaLSTM and Elmo embeddings (Altamimi, et al., 2024). Experimental comparisons show that this improvement has proved beneficial and efficient. In this study, a hybrid approach is used because the proposed model uses a combination of deep neural network-based feature extraction and manually incorporates some essential features using feature engineering techniques. Each feature makes a significant contribution. The Squared Euclidean distance highlighted differences in ELMo embedding magnitudes of both question pair sentences. Incorporating Cosine similarity detected the similarity between the angles of sentences. That's how duplicate questions posed with different vocabulary were identified.

The use of Elmo model for the vectorization of words instead of any static vectorization, made semantic meaning capturing very efficient. The proposed methodology can not only be used specifically for Quora question pairs detection. Rather, there is a vast range of applications for this model. Such as paraphrase identification, information retrieval, etc.

In the future, we aim to experiment with the whole dataset of Quora question pairs. Also, we will experiment by introducing transformers into the current deep neural network.

7. ACKNOWLEDGMENT

We sincerely acknowledge the editor and reviewers of the Journal of Innovative Technology for giving us constructive feedback and the platform to support the publication of this research paper.

REFERENCES

Ali, N., Tubaishat, A., Al-Obeidat, F., Shabaz, M., Waqas, M., Halim, Z., & Rida, I. (2023). "Towards Enhanced Identification of Emotion from Resource-Constrained Language through a novel Multilingual BERT Approach." *ACM Transactions on Asian and Low-Resource Language Information Processing.* doi:10.1145/3592794

Altamimi, A., Umer, M., Hanif, D., Alsubai, S., Kim, T.-H., & Ashraf, I. (2024). "Employing Siamese MaLSTM Model and ELMO Word Embedding for Quora Duplicate Questions Detection." *IEEE Access,* **12**,29072-29082. doi:10.1109/ACCESS.2024.3367978

Bao, W., Dong, J., Xu, Y., Yang, Y., & Qi, X. (2024). "Exploring Attentive Siamese LSTM for Low-Resource Text Plagiarism Detection., (pp. 488–503). doi: https://doi.org/10.1162/dint_a_00242

Bhoi, S., Markhedkar, S., Phadke, S., & Agrawal, P. (2024). MultiSiam: A MultipleInput Siamese Network for Social Media Text Classification and Duplicate Text Detection. Retrieved from https://arxiv.org/abs/2401.06783

Chandra, M., Rodrigues, A., & George, J. (2022). "An Enhanced Deep Learning Model for Duplicate Question Detection on Quora Question pairs using Siamese LSTM." *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)* (pp. 1-5). Ballari, India: IEEE. doi:10.1109/ICDCECE53908.2022.9792906

Dammu∗, P. P., & Alonso, O. (2024). Near-duplicate Question Detection., (p. 4).

Darsh Shah, T. L. (n.d.). Adversarial Domain Adaptation for Duplicate Question Detection.

Devlin, J., Chang, M.-W., & Lee, K. T. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Retrieved from http://arxiv.org/abs/1810.04805

ELMA, J. L. (1990). "Finding structure in time." *Cognitive Science,* 179-21 .

Farouk, M. (2019, jul). "Measuring Sentences Similarity: A Survey." *Indian Journal of Science and Technology,* **12**, 1–11. doi:10.17485/ijst/2019/v12i25/143977

FASEEH, M., KHAN, M. A., IQBAL, N., QAYYUM, F., MEHMOOD, A., & KIM, J. (2024). "Enhancing User Experience on Q&A Platforms: Measuring Text Similarity Based on Hybrid CNN-LSTM Model for Efficient Duplicate Question Detection." *IEEE Access.* doi:10.1109/access.2024.3358422

Gao, W., Yang, B., Xiao, Y., Zeng, P., Hu, X., & Zhu, X. (2024). "Duplicate question detection in community-based platforms via interaction networks." *Multimedia Tools and Applications,* **83**, 10881–10898.

He, H., Gimpel, K., & Lin, J. (2015). "Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks. In ,. Màrquez, C. Callison-Burch, & J. Su (Ed.)." *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* pp. 1576–1586. Lisbon, Portugal: Association for Computational Linguistics. doi:10.18653/v1/D15-1181

Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory." *Neural Computation,* **9**, 1735-1780. doi:10.1162/neco.1997.9.8.1735

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. doi:10.48550/arXiv.1607.01759

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., & Fidler, S. (2015). "Skip-Thought Vectors. " *CoRR, abs/1506.06726,* 0-11. Retrieved from http://arxiv.org/abs/1506.06726

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., . . . Stoyanov, V. (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach." *CoRR, abs/1907.11692.* Retrieved from https://arxiv.org/abs/1907.11692

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. doi:10.48550/arXiv.1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). "Distributed Representations of Words and Phrases and their Compositionality." *CoRR, abs/1310.4546.* Retrieved from http://arxiv.org/abs/1310.4546

Mueller, J., & Thyagarajan, A. (2016). "Siamese recurrent architectures for learning sentence similarity." *Thirtieth AAAI Conference on Artificial Intelligence.*

Patro, B. N., Kurmi, V. K., & Kumar, S. (2018, aug). Learning Semantic Sentence Embeddings using Sequential Pair-wise Discriminator. (E. M. Bender, L. Derczynski, & P. Isabelle, Eds.) 2715--2729. Retrieved from https://aclanthology.org/C18-1230/

Pennington, J., Socher, R., & Manning, C. D. (2014). "GloVe: Global Vectors for Word Representation." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). "Deep contextualized word representations." *arXiv preprint arXiv:1802.05365.*

Quora. (2012). *First Quora Dataset Release: Question Pairs.* Online: Quora. Retrieved from https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

Quora. (n.d.). *Question Pairs Dataset.* Kaggle. Retrieved from https://www.kaggle.com/datasets/quora/question-pairs-dataset

Rasham, S., Khan, H. U., Maqbool, F., Razzaq, S., Anwar, S., & Ilyas, M. (2023). "Structured knowledge creation for Urdu language: A DBpedia approach." *Expert Systems.* doi:10.1111/exsy.13223

Rücklé, A., & Moosavi, N. S. (2019). "Neural Duplicate Question Detection without Labeled Training Data." *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 1607--1617). Hong Kong, China: Association for Computational Linguistics.

Saadat, H., Shah, B., Halim, Z., & Anwar, S. (2024). "Knowledge Graph-Based Convolutional Network Coupled With Sentiment

Analysis Towards Enhanced Drug Recommendation." *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* **21**, 983--994. doi:10.1109/TCBB.2022.3225234

Shah, D., Lei, T., Moschitti, A., Romeo, S., & Nakov, P. (2018). "Adversarial Domain Adaptation for Duplicate Question Detection. " *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* pp. 1056–1063. Brussels, Belgium: Association for Computational Linguistics. doi:10.18653/v1/D18-1131

Shaheen, L., Tubaishat, A., Shah, B., Mussiraliyeva, S., Maqbool, F., Razzaq, S., & Anwar, S. (2025). "DeepVoc: a linked open vocabulary for reproducible and reliable deep learning experiments." *International Journal of Machine Learning and Cybernetics,* 1--13. doi:10.1007/s13042-025-02709-7

Shih, C.-H., Yan, B.-C., Liu, S.-H., & Chen, B. (2017). "Investigating Siamese LSTM networks for text categorization." *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (pp. 641-646). Kuala Lumpur, Malaysia: IEEE. doi:10.1109/APSIPA.2017.8282104

Sravanthi, P., & Srinivasu, D. B. (2017). SEMANTIC SIMILARITY BETWEEN SENTENCES., **4**, pp. 156–161.

Tai, K. S., Socher, R., & Manning, C. D. (2015). "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks." *CoRR, abs/1503.00075,* 1556–1566. Retrieved from http://arxiv.org/abs/1503.00075

Uva, A., Bonadiman, D., & Moschitti, A. (2018). "Injecting Relational Structural Representation in Neural Networks for Question Similarity. In I. Gurevych, & Y. Miyao (Ed.)." *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers,* pp. 285–291. Melbourne, Australia: Association for Computational Linguistics. doi:10.18653/v1/P18-2046

Wang, G., Gill, K., Mohanlal, M., Zheng, H., & Zhao, B. Y. (2013). "Wisdom in the social crowd." *Proceedings of the 22nd International Conference on World Wide Web* (p. 1352). Rio de Janeiro Brazil: International World Wide Web Conferences Steering Committee (IW3C2) in collaboration with the Association for Computing Machinery (ACM). doi:10.1145/2488388.2488506

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers." *CoRR.* Retrieved from https://arxiv.org/abs/2002.10957

Wang, Z., Hamza, W., & Florian, R. (2017). "Bilateral Multi-Perspective Matching for Natural Language Sentences." *CoRR, abs/1702.03814.* Retrieved from http://arxiv.org/abs/1702.03814

Werbos, P. (1990). "Backpropagation through time: what it does and how to do it." *Proceedings of the IEEE,* **78**, 1550-1560. doi:{10.1109/5.58337

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., . . . Brew, J. (2020). Transformers: State-of-the-art natural language processing. (pp. 38--45). Association for Computational Linguistics.

Xu, Z., & Yuan, H. (2020). "Forum Duplicate Question Detection by Domain Adaptive Semantic Matching." *IEEE Access,* 56029-56038. doi:10.1109/ACCESS.2020.2982268

Yenduri, G., M, R., G, C. S., Y, S., Srivastava, G., Maddikunta, P. K., . . . Gadekallu, T. R. (2023). Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions. Retrieved from https://arxiv.org/abs/2305.10435

Yu, L., Hermann, K. M., Blunsom, P., & Pulman, S. (2014). "Deep Learning for Answer Sentence Selection." *CoRR.*

Zhang, L., Xu, M., Wang, G., Shi, R. ,., & Yan, R. (2023). SiameseNet Based Fine-Grained Semantic Change Detection for High Resolution Remote Sensing Images. doi:https://doi.org/10.3390/rs15245631

Zhu, W., Yao, T., Ni, J., Wei, B., & Lu, Z. (2018, 03). "Dependency-based Siamese long short-term memory network for learning sentence representations." *PLOS ONE,* **13**, 1-14. doi:10.1371/journal.pone.0193919