ParaChecker: Elevating Paraphrase Detection Using Deep Learning

Yuti Khamker
Department of Applied Sciences
San Jose State University
San Jose, USA
yutiashwin.khamker@sjsu.edu

Swetha Neha Kutty Sivakumar
Department of Applied Sciences
San Jose State University
San Jose, USA
swethaneha.kuttysivakumar@sjsu.edu

Sourab Saklecha
Department of Applied Sciences
San Jose State University
San Jose, USA
sourabrajendra.saklecha@sjsu.edu

Kashish Thakur
Department of Applied Sciences
San Jose State University
San Jose, USA
kashish.thakur@sjsu.edu

***Abstract*—Quora is one of the popular question-answer platforms for sharing knowledge and engaging in discussions on various topics. An enormous number of users on the Quora website makes it inevitable to have multiple questions from different users with similar intent. Question and answering social media platforms like Quora is exceptionally significant to ensure quality content is presented to users. Identifying semantically identical questions based on the intent of the question will enhance the overall user experience. This project contributes to finding pairs of questions that are semantically identical using deep learning models to tackle a binary classification problem. Before embedding, the dataset is preprocessed using a variety of Natural Language processing techniques to increase consistency. Followed by training the deep learning models, Gated Recurrent Unit (GRU), and Long-Short Term Memory (LSTM) model in Siamese architecture using GLoVe (Global Vectors for word representation) and Word2Vec (Word to Vector) embedding methods respectively. The LSTM model with Word2Vec achieved an accuracy of 83.30% surpassing the GRU model.**

Keywords— Natural Language processing, GRU, LSTM, Deep Learning, Siamese Network

I. Introduction

Improving user accessibility and efficiency requires Question and Answering (Q&A) sites like Quora to identify questions with related intents. However, identifying paraphrased questions is difficult due to the expressive and diverse nature of natural language, where the same meaning can be expressed using different words and sentence structures. This complexity is often too complex for traditional natural language processing approaches as it often fails to capture the nuanced semantic similarity. To overcome the limitations, deep learning approaches will be incorporated that have been proven to be more effective at finding semantic similarities between texts, which streamlines information access and increases the readership for writers, addressing the practical demand for effective knowledge-sharing platforms and contributing to preserving the reliability and standard of the cQA sites. Before embedding, the dataset is preprocessed using a variety of techniques to address misspellings, tokenize sentences to individual words, eliminate stop words, and lemmatize to increase consistency. The embedding method GLoVe (Global Vectors for word representation) and Word2Vec (Word to Vector) will be used after the preprocessing to process contextualized embeddings to the context in which the words appear. After preprocessing, 80% of the dataset will be split into a Training set and the remaining 20% for testing. By applying two deep learning models this research contributes to LSTM and GRU deep neural network models consisting of embedding layer, LSTM or GRU layer, dropout, and dense layers. The best model achieved 83.30% accuracy by extracting the intricate features to determine the similarity score of the question pairs.

II. Related Work

There is a need for large amounts of labeled question pairs while training neural models to detect duplicate questions in community Question Answering (cQA) platforms. Two novel methods, first, an automatic generation of duplicate questions and second, a weak supervision technique using just the title and body of a question are proposed, utilizing a publicly available question-answer pair data, mostly sourced from StackExchange. Employing BERT for fine-tuning was a novel approach that helped them better benchmark results. They used f1 scores and P-values to evaluate their models [2].

The next study works on a similar dataset, Sprint FAQ, and Quora. Utilized Bi-LSTM model with an encoder that operates on 300-dimensional GloVe word embeddings has three components-A question encoder, a similarity function, and a domain adaptation component. Evaluating their model using Area under the curve (AUC) on positive pairs vs negative ones [3]. The following work expanded the Qoura question pairs (QQP) dataset to the Answer Enhanced Quora question pair (AeQQP) and used that along with the CQADupStack benchmark dataset to feed their adaptive multi-attention network (AMAN) model. Models were compared using precision, recall, f1, and accuracy scores, with architecture outperforming SOTA benchmarks [4].

According to [5], incorporating BiLSTM encoder to encode two sentences, P and Q. By using a "matching-aggregation" structure, the model matches the sentences in both directions from different angles. Using common benchmark datasets, they assess the BiMPM model on three tasks: answering sentence selection, natural language inference, and paraphrase detection. The proposed model achieves state-of-the-art performance across all tasks. The study proposed a model architecture where a Bidirectional Transformer Encoder is combined with a Convolutional Neural Network to identify the Question Paraphrase. It includes two different inference setups: Siamese and Matched Aggregation. Out of which the Matched Aggregation setup produces better results, which gives an accuracy of 90.80% and an F1 Score of 0.9022 [6].

Working on Quora question pairs dataset using bag of words (BoW) to perform Paraphrase Identification, using LSTM and BERT. Countvectorizer is used for feature extraction, and term-frequency documents along with a unigram for their XGbooster and CatBooster model [7]. Similarly utilizing the same dataset, a novel Siamese LSTM model was introduced incorporating Manhattan distance, outperforming existing models with an accuracy of 91.14%. Three different word embedding techniques were explored—Google News Vector, FastText Crawl, and FastText Crawl Subword—to vectorize questions and train the proposed model [8]. The next approach introduces a framework that extends Word Mover's Distance (WMD) by presenting text documents as a normal distribution instead of a bag of embedded words. The suggested framework employs the Wasserstein 2 distance metric and utilizes a variational siamese network to learn semantic similarity discriminatively. Evaluation of Wasserstein 2 distance is done on Quora dev and test data and is reported using Area under curve (AUC) [9]. The study utilizes the BERT model with Global Vector embeddings (GLOVE) and contextualized embeddings from Language models (ELMO) to identify the paraphrased sentences using the Microsoft Research Paraphrase(MSRP), and Quora Question Pairs(Quora) datasets. The fine-tuned BERT model shows better performance with ELMo embedding and achieves the identification rate on the two datasets, MSRP and Quora 86.51% and 94.32% [10]. It proposes a framework using the weighted Fine-Tuned BERT Feature extraction with the Siamese Bi-LSTM model. Worked on a Quora dataset using GLOVE word-based embedding for retaining semantic text-similarity. The proposed model outperformed the CNN and Multi-Layer Perceptron(MLP) models, attaining an efficient detection rate of 90% [11]. The following approach worked on an ensemble model using GloVe embedding, LSTM, Convolution, Max pooling, Dense layers, Batch normalization, and Activation function to detect the semantically duplicate questions[12].

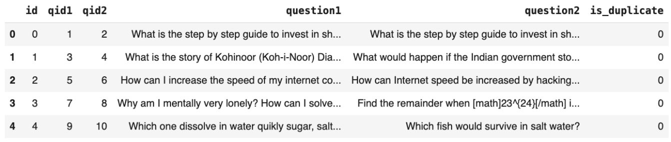
The next study worked on the Stack Overflow dataset to tackle the problem of duplicate question detection, by employing deep learning techniques like Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) in conjunction with Word2Vec for semantic understanding. The results show that WV-CNN and WV-LSTM outperform baseline and machine learning techniques like SVM, Logic Regression, Random Forest, and eXtreme Gradient Boosting in terms of recall rates [13]. The following study utilized LSTM and BiLSTM models, along with Word2Vec and pre-trained fastText word embeddings to classify whether the question pair is duplicate or not [14].

The study proposes the use of a light weight unified model to simultaneously classify whether a sentence is a paraphrase of the other and to generate the paraphrased sentence of a given input sentence. The model is assessed against the evaluation metrics such as BiLingual Evaluation Understudy(BLEU), Recall Oriented Understudy for Gisting Evaluation(ROUGE), Word Error Rate(WER), and Google BLEU(GLEU) for paraphrase generation and is evaluated against performance metrics such as accuracy, Precision, and Recall. It achieves better results for the Quora dataset, where it achieves an accuracy of 87.17%, precision of 78.9%, and recall of 93.75% [15]. The next study proposed Conditional Variational AutoEncode(CVAE) and multi-task VAE framework for Question Paraphrase Identification (QPI) using the corpora's dataset. The models are evaluated against the transformed models such as BERT, RoBERTa, and ERNIE, using Accuracy and F1-score metrics. The approach outperformed the existing models without reducing overall efficiency [16].

III. Methodology

A. *Dataset Description*

For this project we use The Quora question pair dataset which comprises 400,000 records featuring pairs of questions from the platform. Each record includes binary labels indicating whether the questions in the pair are semantically corresponding or not. The dataset is designed to address potential duplicate question pairs, providing a binary value to signify their presence. Each record contains the complete text of both questions along with unique IDs for each question in the pair. Since it is never possible to know for sure what a sentence really means, ground truth labels are by their very nature subjective. Arguably, human labeling is a 'noisy' process as well. Because of this, it is important to note that the ground truth labels on this dataset are 'informed,' rather than 100% accurate, and may contain inaccurate labeling. Although we think the labels, generally speaking, reflect a decent consensus, this may not always be the case for specific items within the collection. The data was sourced from Kaggle. Figure 1 below depicts the initial raw dataset collected from the data source.



	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0

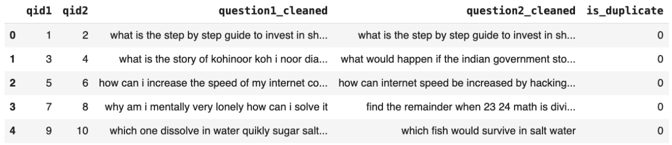
Fig. 1. Initial Raw Dataset

B. *Dataset Splits*

To provide a reliable assessment of our deep learning model, we carefully partitioned the dataset for our project. We preserved the class distribution of duplicate and non-duplicate questions across the training, validation, and test sets by using stratified sampling. This approach ensured that the model is tested on a representative sample, prevented data leakage, and maintained the dataset's natural balance. The dataset was specifically divided into three categories: Training (80%), Validation (10% of the Training data), and Testing (20%). To obtain accurate and impartial performance measurements, this meticulous separation is essential.

C. *Data Preprocessing*

A number of crucial procedures were involved in preparing our dataset in order to get the text data ready for efficient model training. To maintain data integrity, we first dealt with null values and eliminated duplicate entries. The content was then cleaned up by removing emoticons and HTML tags. After that, we capitalized every word in the text, eliminated any punctuation, and changed it to lowercase. Words were standardized to their base forms using lemmatization, and stop words were eliminated to cut down on noise. These actions were crucial in converting the unprocessed text into a format that could be used for the creation of significant word embeddings and the ensuing model training. Figure 2 illustrates the cleaned dataset after performing the different preprocessing steps.



	qid1	qid2	question1_cleaned	question2_cleaned	is_duplicate
0	1	2	what is the step by step guide to invest in sh...	what is the step by step guide to invest in sh...	0
1	3	4	what is the story of kohinoor koh i noor dia...	what would happen if the indian government sto...	0
2	5	6	how can i increase the speed of my internet co...	how can internet speed be increased by hacking...	0
3	7	8	why am i mentally very lonely how can i solve it	find the remainder when 23 24 math is divi...	0
4	9	10	which one dissolve in water quikly sugar salt...	which fish would survive in salt water	0

Fig. 2. Cleaned Dataset

IV. Model Architecture

A. *Siamese Network Architecture*

The project incorporates Siamese Network Architecture to identify whether the pair of questions are paraphrased or not. The architecture is explicitly developed to perform the tasks requiring similarity learning. It incorporates twin subnetworks, where each network shares the same weights and architecture. The sharing of weights plays a crucial role in the network architecture when it comes to tasks such as identifying the difference between two similar and dissimilar inputs. Feature Extraction is consistent when each input in the two identical subnetworks is processed with shared weights. This further aids in fair and consistent comparison among the inputs. The concept of contrastive learning integrated with the Siamese network helps in improving the capability of the model to distinguish between similar and non-similar inputs, which builds a robust and discriminative feature representation. Following the feature extraction, the corresponding feature vectors are extracted. The output of the network is a single value representing the probability score. The aim of the Loss function is to minimize the error in the obtained output [1].

In the project, we have used two different variations of Siamese Network Architecture.

1) Siamese LSTM Model, using Pre-trained Word2Vec embedding.

2) Siamese GRU Model, using Pre-trained Glove embedding.

Figure 3 illustrates the Generic Siamese Network Architecture.

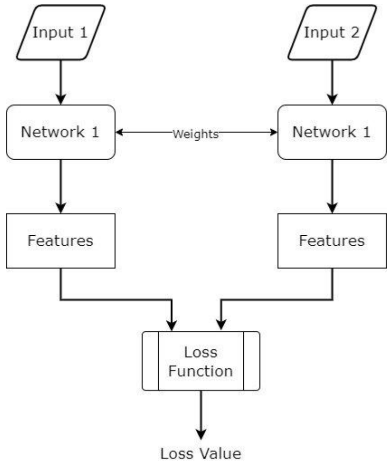


Fig. 3. Siamese Network Architecture

The cleaned text inputs are first converted to string values to ensure all the values are of the type string. The text inputs are then converted into sequences of integers using Keras Tokenizer, and the size of the vocabulary is determined. The vocabulary size is computed by identifying the unique words in the combined tokenized text, along with an additional one to include the padding token. The tokenized sequences are then padded to ensure that both inputs are of equal length. The word embedding used for the given models are a pre-trained Word2Vec model and GLoVE model, GoogleNews-vectors, and Glove vectors respectively. The dataset is split into a Train, and Test set, where a Validation Set is provided during the Model Training.

- Training Set: 80% of data
- Testing Set: 20% of data
- Validation Set(within training): 10% of the training data

1) *Siamese LSTM Model Architecture*

Figure 4 depicts the architecture of the proposed Siamese LSTM Model, including all the layers, and components used.

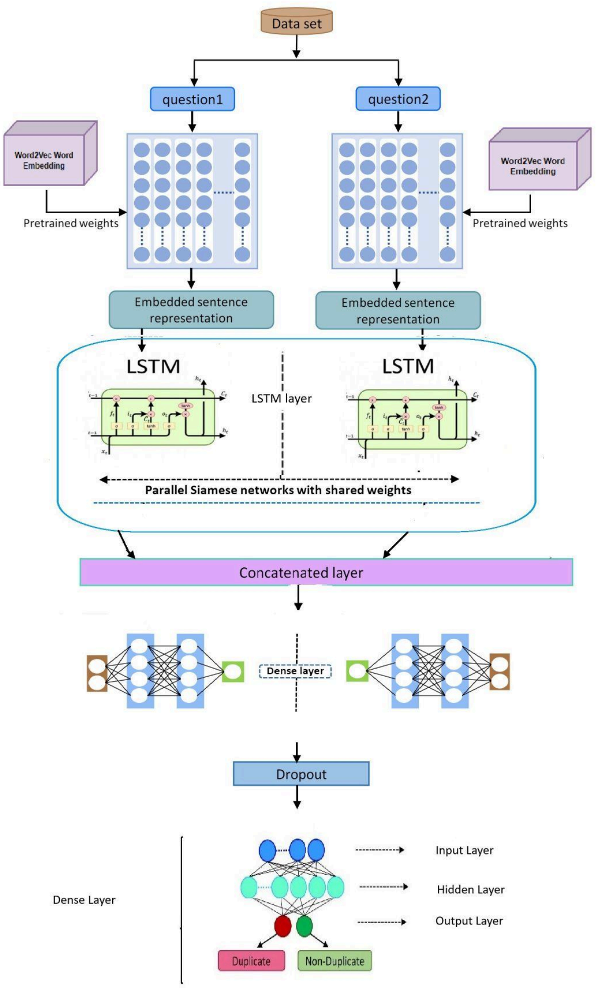


Fig. 4. Siamese LSTM Model Architecture

The Siamese neural network takes two separate inputs, where each input represents the questions to be compared, which are represented as input_q1, and input_q2. The maximum length of each input is set to 50. This represents the Input Layer of the Siamese Network. The input layer is followed by the Embedding Layer, where both inputs go through identical neural network layers, sharing the same weights and architecture, ensuring consistent and fair comparison between the two inputs. The embedding layer transforms the input sequences into corresponding word vectors based on the given embedding size. The size of the word2vec embedding is set to 300. The output of the embedding layer is then fed to LSTM layers, where the embedded sequences are processed to produce an output tensor containing the hidden state units, having the shape, (None, lstm_hidden_units).

The LSTM hidden units are set to 128. The output of the LSTM layer represents the summary of the whole processed sequence, capturing the semantic meaning, and context from the input sequences. The LSTM layer and Embedding layers are considered Shared Layers, the overall purpose of the two layers is to extract the most relevant features from the input sequences. Once the inputs are processed, through the Embedded and the LSTM layers, their outputs are merged or concatenated, which is performed by the Concatenated Layer. The concatenation layer basically combines the features extracted from both inputs into a single representation. Following the Concatenation Layer we have additional Discriminative Layers like Dense and Drop layers. The purpose of the Dense layers is to understand the discriminative features from the combined representation, while the Dropout layer is added to overcome overfitting. These layers are used to transform the combined representation into a form suitable for classification. Finally, we have a single Output Layer, using a Sigmoid Activation Function which produces the probability score. The loss function used is Binary Cross Entropy. Figure 5 illustrates the summary of the Siamese LSTM model.

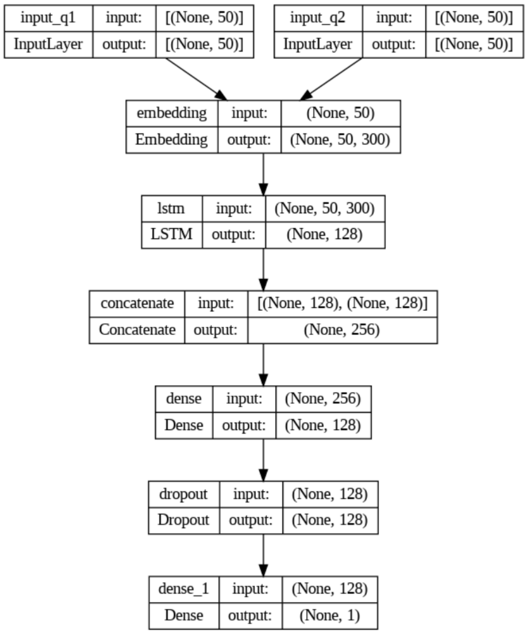


Fig. 5. Siamese LSTM Model Summary

2) Siamese GRU Model

Similar to the previous model, the proposed model architecture for duplicate detection on Quora Question pairs also employs the Siamese Network structure but with the GLoVE word embeddings. The purpose of using GRU here is to mitigate the vanishing gradient problem. Figure 6 shows the architecture of the Siamese GRU Model with the GRU layer and other components.

The inputs given here consist of question pairs that are question 1 and question 2 (q1 and q2). These inputs are then passed to the embedding layers through an identical network. Embedding layers used here are the pre-trained GLoVE word embeddings. The embedding layers convert input tokens into dense vector representations which provide rich semantic information. The size of GLoVE embedding is set to 100. These vector representations generated as an output from the embedding layers are fed into parallel GRU layers(Gated Recurrent Units) which capture the sequential dependencies and long-term relationships. Components of GRU consist of i) Updated Gate, ii) Reset Gate, iii) Candidate Hidden State, and iv) Final Hidden State. The general working of GRU calculates the Updated and Reset Gate based on the current input received and the prior hidden state at each time step. The amount of the hidden state that should impact the candidate hidden state is then calculated using the reset gate. Finally, the candidate hidden state and previous hidden state are combined to update the final hidden state according to the updated gate. The question pairs are processed with shared weights to ensure that both representations learned are comparable. The outputs from the GRU layer are passed to a concatenated layer which combines the representation or the features extracted from the question pair by concatenating the final hidden states of the GRU layers. Similar to the previous model architecture, after concatenated layers, the dense and dropout layers are added to learn higher-level interactions and to prevent overfitting respectively. The dropout layer helps in regularization as it randomly drops a fraction of neurons during training. Then the probability score is generated as an output value ranging between 0 to 1 from the final dense layer which uses the sigmoid activation function, indicating the probability that the two questions are the same. Figure 7 illustrates the summary of the Siamese GRU Model.

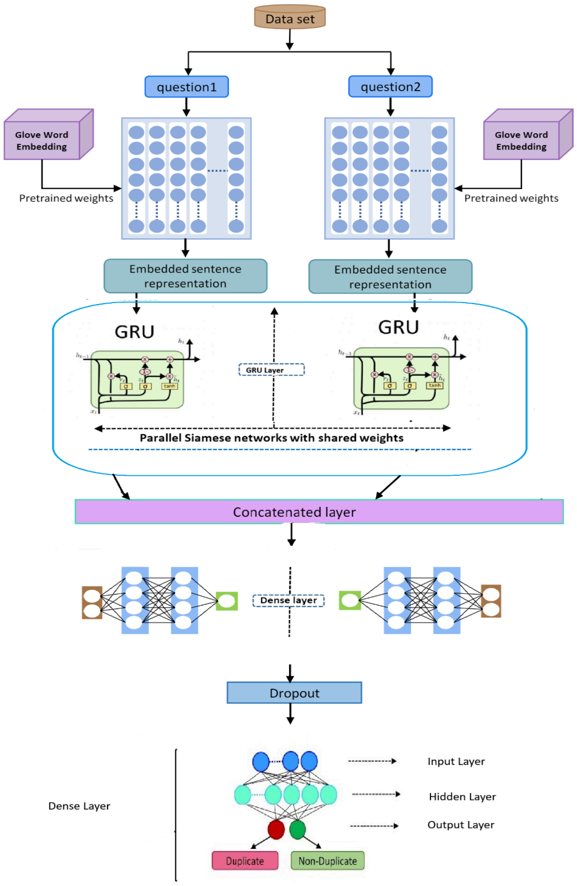


Fig. 6. Siamese GRU Model Architecture

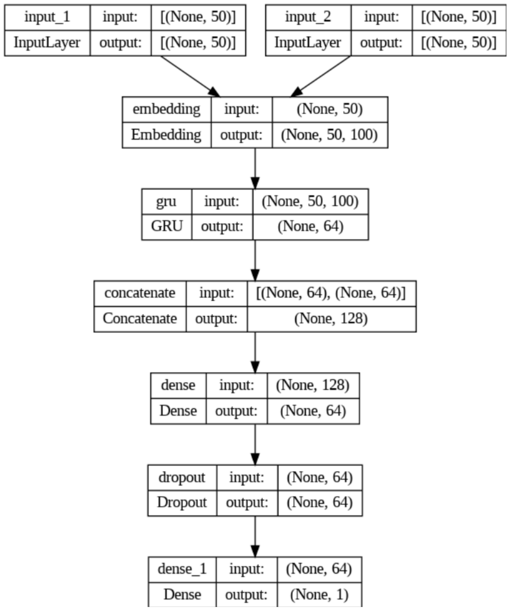


Fig. 7. Siamese GRU Model Summary

V. Experimental Setup

A number of hyperparameters are used for the two variations of the Siamese Network Model. The purpose and explanation of each is explained below. Both local and cloud-based resources are utilized to execute the train and evaluate the models for question paraphrase detection

1) Local Resources- Apple Macbook M2, with 8GB RAM, and macOS Sonoma 14.4.1

2) Cloud-based Resources- Google Colab Pro with Nvidia T4-GPU with High RAM runtime

A. *Siamese LSTM and Siamese GRU Model*

The maximum length of the input word is set to 50, as during the data exploration the majority words are within the range 0 to 50. The learning rate for the model is set using the Adam Optimizer, which is an adaptive learning rate mechanism that updates the learning rate based on the first and second moments of gradients. The batch size of the model is decided by considering the memory usage and training speed. The value selected maintains the balance between the two. For the Activation functions, since the project is using LSTM and GRU units, the default activation functions are used for it, which are tanh for cell gate, and sigmoid for gates. The Dense layers use ReLU activation functions, which effectively handle the non-linearities. The output layer uses the Sigmoid activation function, as the project is performing binary classification.

The word embedding used for the Siamese LSTM model is the Google News Word2Vec model, which is a pre-trained model extensively trained on a large and diverse corpus of 100 billion words from the Google News dataset. The words are represented using a 300-dimensional vector, as the pre-trained model has the same dimensionality, which is able to effectively capture the semantic meaning and syntax roles. The purpose of using a pre-trained word2Vec model is to incorporate the high-quality word embeddings that are already trained on a huge corpus and are able to capture deep semantic relationships between words. The word embedding used for the Siamese GRU model is from Glove and was trained on 6 billion tokens and words are represented by 100 dimensional vectors. The reason for choosing 100 dimensions is that many benchmark datasets have performed competitively on this dimension size. To decide the optimal number of neurons for LSTM and GRU, model complexity and its ability to capture patterns in text are taken into account. Setting the value of the LSTM and GRU unit to 128, and 64 respectively where the models are able to capture the patterns, without being too complex. The dropout rate is used for the LSTM layer as well as for GRU layer and the dense layer as the regularizers to avoid overfitting. The recurrent dropout rate is used as the regularizer for the recurrent connections. The Loss function used for the model is Binary Cross Entropy, which is suitable for tasks of Binary Classification. Table 1 illustrates the list of hyperparameters used with their corresponding values.

TABLE I. HYPERPARAMETER USED

Hyperparameter	Value (Siamese LSTM Model)	Value (Siamese GRU Model)
Learning Rate	0.001	0.001
Batch Size	64	64
Number of Epochs	20	20
Activation Function	ReLU(Dense layer) and Sigmoid (Output)	ReLU(Dense layer) and Sigmoid(Output)
Embedding Dimension	300	100
Units	128	64
Dropout Rate	0.2	0.5
Recurrent Dropout	0.2	0.5
Optimizer	Adam	Adam
Loss Function	Binary Cross Entropy	Binary Cross Entropy

Table 2 illustrates the software libraries and frameworks used in the model.

TABLE II. SOFTWARE LIBRARIES AND FRAMEWORKS USED

Library/ Framework	Version	Purpose
TensorFlow	2.11.0	Backend library for deep learning, used for building and training the model
Keras	2.11.0	Provides tools for text processing, contains functions for working with sequence data, to construct and train deep learning models, provides various neural network layers such as Input defines the input layer, Embedding is used for word embeddings, LSTM, GRU, for Long Short-Term Memory networks, Concatenate for merging layers, Dense for fully connected layers, and Dropout for regularization, Reloading saved models for inference or further training
Gensim	4.1.2	Used for loading and working with pre-trained word embeddings (Word2Vec), Glove embedding
Sci-kit learn	1.0.2	Provides utility functions for preprocessing data. LabelEncoder is used to convert categorical labels into numeric form.
Pandas	1.4.2	Used for loading, processing, and transforming the dataset.
NumPy	1.22.4	Provides support for large, multi-dimensional arrays and matrices, along with mathematical functions
Matplotlib	3.5.2	Used for visualizing the model performance and data analysis results.
Python	3.9.2	The programming language used for scripting and implementing the entire analysis, pickle is a standard library module used for serializing and deserializing Python objects, allowing you to save and load model artifacts.

VI. RESULTS AND ANALYSIS

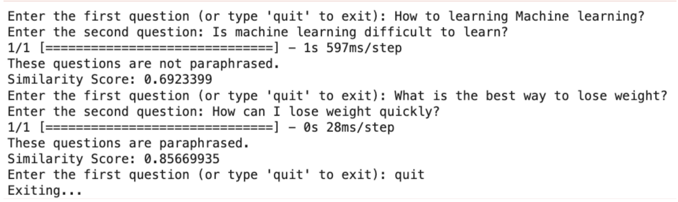
A. Results

The table below shows the comparison results of the GRU and LSTM models using the Siamese Network. The performance of the models is evaluated using Accuracy, Precision, Recall, and F1 score. Based on the quantitative values, LSTM using Word2VEc embedding outperformed GRU using GLoVE embedding across all evaluation metrics. The LSTM model achieved an Accuracy of 83.30%, a precision of 84%, a Recall of 83%, and an F1 score of 83%. On the other hand, GRU with Glove achieved 77.34% accuracy, precision of 78%, recall of 77% and F1 score of 78%. Based on the evaluation metric, the LSTM model is effective in identifying duplicate questions.

TABLE III. MODEL COMPARISON

Models	Embedding	Accuracy	Precision	Recall	F1 Score
LSTM with Siamese network	Word2Vec	83.30%	84%	83%	83%
GRU with Siamese network	GloVe	77.34%	78%	77%	78%

Table 3 illustrates the model comparison based on different embeddings used, along with the evaluation metrics.



```
Enter the first question (or type 'quit' to exit): How to learning Machine learning?
Enter the second question: Is machine learning difficult to learn?
1/1 [==============================] - 1s 597ms/step
These questions are not paraphrased.
Similarity Score: 0.6923399
Enter the first question (or type 'quit' to exit): What is the best way to lose weight?
Enter the second question: How can I lose weight quickly?
1/1 [==============================] - 0s 28ms/step
These questions are paraphrased.
Similarity Score: 0.85669935
Enter the first question (or type 'quit' to exit): quit
Exiting...
```

Fig. 8. Models Prediction Results.

Figure 8 depicts the results received for the predictions made by the models. Each model outputs a probability score indicating the similarity between the two input questions also known as similarity score. The similarity closer to 1 indicates high similarity, whereas the value closer to 0 indicates a low similarity. The value of the similarity scores is compared with a threshold value to decide whether the questions are paraphrased or not.

B. *Analysis*

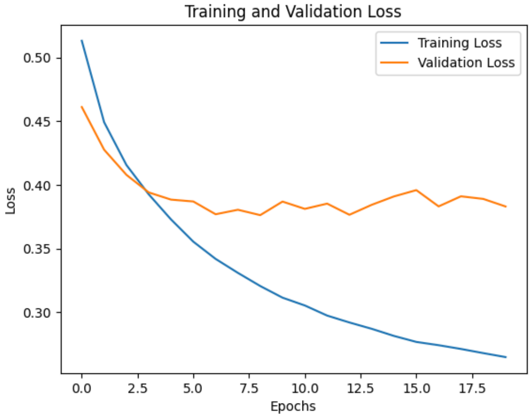


Fig. 9. Training and Validation Loss Curve for Siamese LSTM Model

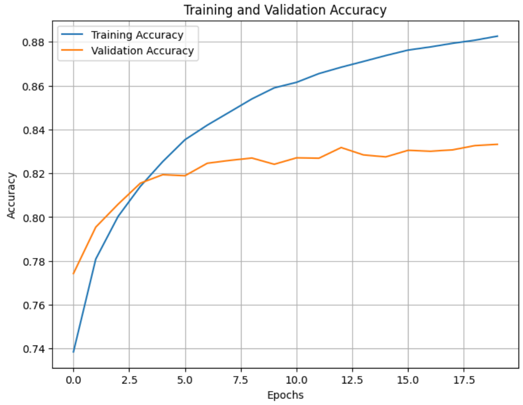


Fig. 10 Training and Validation Accuracy Curve for Siamese LSTM Model

The above figures 9 and 10 show the accuracy and loss curves of the Siamese LSTM model during the training and validation phases. The higher validation loss and lower accuracy during the validation phase determine the overfitting of the training data. The model has not effectively learned the pattern and its generalization can be improved.

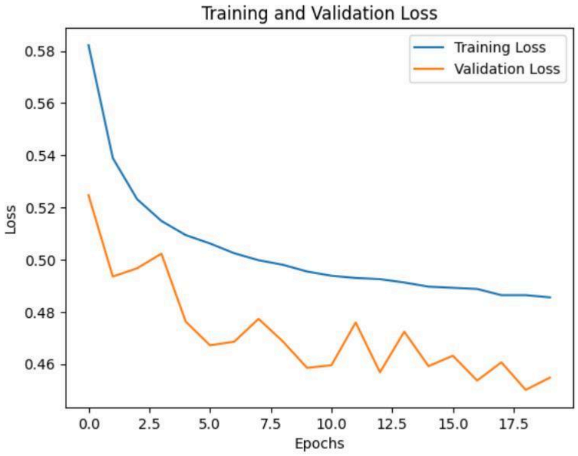


Fig. 11. Training and Validation Loss Curve of Siamese GRU Model

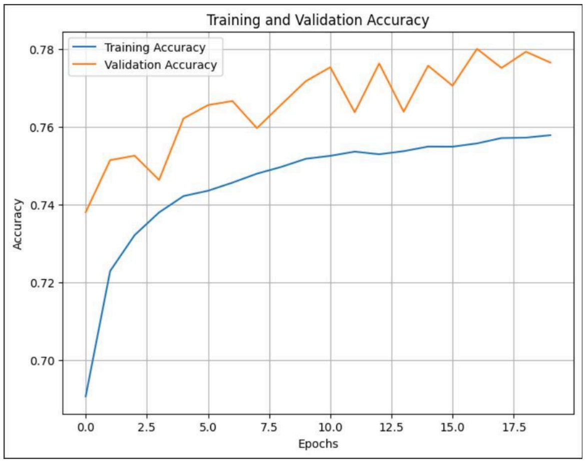


Fig. 12. Training and Validation accuracy Curve of Siamese GRU Model

The above Figures 11 and 12 show the accuracy and loss curves of the Siamese GRU model during the training and validation phases. The above graph illustrates the increased accuracy and decreased loss during the validation phase, demonstrating that the model is correctly predicting values that are close to the true values, which in turn indicates that the model has successfully learned the underlying pattern.

VII. Conclusion

This project's main contribution is to efficiently find question pairs that are semantically related on Q&A sites such as Quora, improving information accessibility and user experience. The limitations of conventional natural language processing techniques in capturing subtle semantic similarities were overcome by using Glove and Word2Vec embeddings and deep learning models, namely GRU and LSTM within a Siamese network architecture. With an accuracy of 83.3%, the Siamese LSTM Model with word2vec embedding outperformed the Siamese GRU Model with GLoVE embedding. These results are noteworthy because they show how deep learning techniques might boost the effectiveness and dependability of knowledge-sharing platforms. The improved model performance implies that users should anticipate more precise identification of questions that have been paraphrased, which will improve user happiness and the flow of knowledge.

In order to further enhance performance, future research may investigate the usage of more sophisticated models, such as Transformer-based architectures. More complex question classification might also be possible by combining datasets with different goal classes. The model's capacity to recognize questions that are semantically linked may also be improved by including answer text data as an additional information source. Lastly, more gains in model robustness and accuracy may be possible by merging and fine-tuning several word embeddings.

References

[1] B.-R. Cha and B. Vaidya, “Enhancing Human Activity Recognition with Siamese Networks: A Comparative Study of Contrastive and Triplet Learning Approaches,” *Electronics*, vol. 13, no. 9, pp. 1739–1739, May 2024, doi:https://doi.org/10.3390/electronics13091739.

[2] A. Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych, “Neural Duplicate Question Detection without Labeled Training Data,” *White Rose Research Online (University of Leeds, The University of Sheffield, University of York)*, Jan. 2019, doi:https://doi.org/10.18653/v1/d19-1171.

[3] D. J. Shah, T. Lei, Alessandro Moschitti, S. Romeo, and Preslav Nakov, “Adversarial Domain Adaptation for Duplicate Question Detection,” *arXiv (Cornell University)*, Jan. 2018, doi: https://doi.org/10.18653/v1/d18-1131.

[4] D. Liang *et al.*, “Adaptive Multi-Attention Network Incorporating Answer Information for Duplicate Question Detection,” *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR’19*, 2019, doi: https://doi.org/10.1145/3331184.3331228.

[5] L. Wang, L. Zhang, and J. Jiang, “Detecting Duplicate Questions in Stack Overflow via Deep Learning Approaches,” Dec. 2019, doi: https://doi.org/10.1109/apsec48747.2019.00074.

[6] H. Sakhrani, S. Parekh, and P. Ratadiya, “Contextualized Embeddings based Convolutional Neural Networks for Duplicate Question Identification.” Accessed: May 18, 2024. [Online]. Available: http://arxiv.org/pdf/2109.01560.pdf

[7] A. Chandra and R. Stefanus, “Experiments on Paraphrase Identification Using Quora Question Pairs Dataset,” *arXiv (Cornell University)*, Jan. 2020, doi: https://doi.org/10.48550/arxiv.2006.02648.

[8] Z. Imtiaz, M. Umer, M. Ahmad, S. Ullah, G. S. Choi, and A. Mehmood, “Duplicate Questions Pair Detection Using Siamese MaLSTM,” *IEEE Access*, vol. 8, pp. 21932–21942, 2020, doi: https://doi.org/10.1109/access.2020.2969041.

[9] M. Deudon, "Learning semantic similarity in a continuous space, Advances in Neural Information Processing Systems, vol. 31, 2018.

[10] A. Razaq, Z. Halim, Atta Ur Rahman, and Kholla Sikandar, “Identification of paraphrased text in research articles through improved embeddings and fine-tuned BERT model,” *Multimedia Tools and Applications*, Feb. 2024, doi: https://doi.org/10.1007/s11042-024-18359-w.

[11] D. Viji and S. Revathy, “A hybrid approach of Weighted Fine-Tuned BERT extraction with deep Siamese Bi – LSTM model for semantic text similarity identification,” *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 6131–6157, Jan. 2022, doi: https://doi.org/10.1007/s11042-021-11771-6.

[12] N. Ansari and R. Sharma, “Identifying Semantically Duplicate Questions Using Data Science Approach: A Quora Case Study,” *arXiv (Cornell University)*, Apr. 2020, doi: https://doi.org/10.48550/arxiv.2004.11694.

[13] Z. Wang, W. Hamza, and R. Florian, “Bilateral Multi-Perspective Matching for Natural Language Sentences,” Feb. 2017, doi: https://doi.org/10.48550/arxiv.1702.03814.

[14] Maksuda Bilkis Baby, Bushra Ankhari, Md Shajalal, Md. Atabuzzaman, Fazle Rabbi, and Masud Ibn Afjal, “Identifying Duplicate Questions Leveraging Recurrent Neural Network,” *Lecture notes in networks and systems*, pp. 331–341, Jan. 2023, doi: https://doi.org/10.1007/978-981-19-9483-8_28.

[15] H. Palivela, “Optimization of paraphrase generation and identification using language models in natural language processing,” *International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100025, Nov. 2021, doi: https://doi.org/10.1016/j.jjimei.2021.100025.

[16] Z. Jin, Y. Hong, R. Peng, J. Yao, and G. Zhou, “Intention-Aware Neural Networks for Question Paraphrase Identification,” *Lecture notes in computer science*, pp. 474–488, Jan. 2023, doi: https://doi.org/10.1007/978-3-031-28244-7_30.