# Paraphrase Identification Using Deep Learning

**Lokesh Dammalapati; Thejaswi Mullapudi; Keerthi Kappera**

**Khaled Sayed, Ph.D. (Assistant Professor)**

---

**Abstract**

Paraphrase identification is a crucial task in Natural Language Processing (NLP), involving the determination of semantic equivalence between sentence pairs. This work evaluates the performance of four deep learning architectures Feedforward Neural Networks (FFN), Bi-LSTM with Gated Relevance Network (GRN), Siamese Networks, and Convolutional Neural Networks (CNN) on the Quora Question Pairs dataset. Key contributions include preprocessing with text cleaning and GloVe embeddings, comparative performance analysis, and detailed error analysis. Among the evaluated models, Bi-LSTM GRN achieved the best F1-Score of 62.4%, demonstrating its superior ability to capture sequential dependencies and semantic relationships. This paper provides a critical analysis of model limitations and offers suggestions for advancing paraphrase identification tasks.

---

## 1. Introduction

### 1.1 Motivation

Paraphrase identification underpins several NLP applications, including question-answering systems, plagiarism detection, and semantic search engines. The task involves identifying semantically equivalent sentences despite lexical and syntactic variations. For instance, the sentences, "How can I improve Python skills?" and "What are the best methods to learn Python?" are lexically distinct but convey the same intent. These linguistic variations make paraphrase identification a challenging problem in computational linguistics.

### 1.2 Challenges

The primary challenges in paraphrase identification are:

- Handling linguistic nuances, idiomatic expressions, and synonyms.
- Dealing with imbalanced datasets, where "Not Duplicate" cases often dominate.
- Effectively leveraging contextual information in model architectures.

### 1.3 Contributions

This paper addresses the above challenges through:

1. A comparative study of four distinct neural architectures for paraphrase detection.
2. Insights into preprocessing techniques that enhance model performance.
3. Error analysis to uncover and address limitations in current approaches.

## 2. Related Work

### 2.1 Early Methods

Traditional approaches to paraphrase identification relied on lexical similarity metrics like Jaccard similarity, cosine similarity, and handcrafted features such as part-of-speech tags or dependency parsing. However, these methods struggled with semantic equivalence due to their inability to capture deeper contextual relationships.

### 2.2 Modern Deep Learning Approaches

The advent of deep learning has transformed paraphrase detection:

- Bi-LSTM Models: These models are adept at capturing sequence-level dependencies and have been widely used in tasks like sentiment analysis, machine translation, and paraphrase identification.
- Siamese Networks: By employing twin subnetworks with shared weights, these models compare sentence embeddings and measure similarity using distance metrics.
- CNNs: Known for their ability to identify local patterns, CNNs work well in tasks requiring feature extraction at the n-gram level.
- Pre-trained Embeddings: Word2Vec, GloVe, and fastText have improved semantic representation by embedding words in high-dimensional vector spaces.

### 2.3 Quora Dataset Studies

The Quora Question Pairs dataset has become a benchmark for paraphrase detection tasks. Studies using this dataset highlight the challenges of handling noisy and imbalanced data, where a majority of pairs are labeled as "Not Duplicate."

---

## 3. Methodology

### 3.1 Dataset

The Quora Question Pairs dataset is a widely used benchmark comprising 404,000 sentence pairs labeled as "Duplicate" or "Not Duplicate." For this study, the dataset was split into 80% training and 20% testing subsets

### 3.2 Preprocessing

Preprocessing steps included:

1. Text Cleaning: Removal of stopwords, punctuation, and non-alphabetic characters, followed by lemmatization using spaCy.
2. Tokenization and Padding: Conversion of text into sequences of integers and padding to a uniform length of 128 tokens.
3. Embedding Layer: Utilization of pre-trained GloVe embeddings (200 dimensions) to capture semantic information.

### 3.3 Model Architectures

**1. Feedforward Neural Network (FFN):**

A straightforward neural network comprising an input layer, one or more hidden layers of fully connected neurons, and an output layer. This architecture serves as a baseline, processing input features through successive layers to produce an output, without accounting for sequential dependencies.

**2. Bidirectional Long Short-Term Memory with Gated Relevance Network (Bi-LSTM with GRN):**

This model integrates Bidirectional Long Short-Term Memory (Bi-LSTM) networks with a Gated Relevance Network (GRN). Bi-LSTMs process sequences in both forward and backward directions, capturing context from both past and future states, which is beneficial for understanding dependencies in sequences. The GRN component enhances this by focusing on relevant features within the sequence, effectively capturing semantic interactions between different parts of the input.

**3. Siamese Network:**

A Siamese Neural Network consists of two identical subnetworks that share the same architecture and weights. Each subnetwork processes one of the two input data points (e.g., question pairs) to generate embeddings. These embeddings are then compared using a similarity function to determine how alike the inputs are. This architecture is particularly useful for tasks like duplicate question detection, where the goal is to assess the similarity between two inputs.

**4. Convolutional Neural Network (CNN):**

Designed to automatically and adaptively learn spatial hierarchies of features from input data. In the context of text data, CNNs apply convolutional filters to capture local patterns such as n-grams within token embeddings. This allows the network to effectively extract meaningful features that contribute to understanding the structure and semantics of the text.

Each of these architectures offers unique strengths, making them suitable for various tasks in natural language processing and machine learning.

Figure 4.1. Architectures of Four Models: (a) Feedforward Neural Network, (b) Bi-LSTM with GRN, (c) Siamese Network, and (d) Convolutional Neural Network.

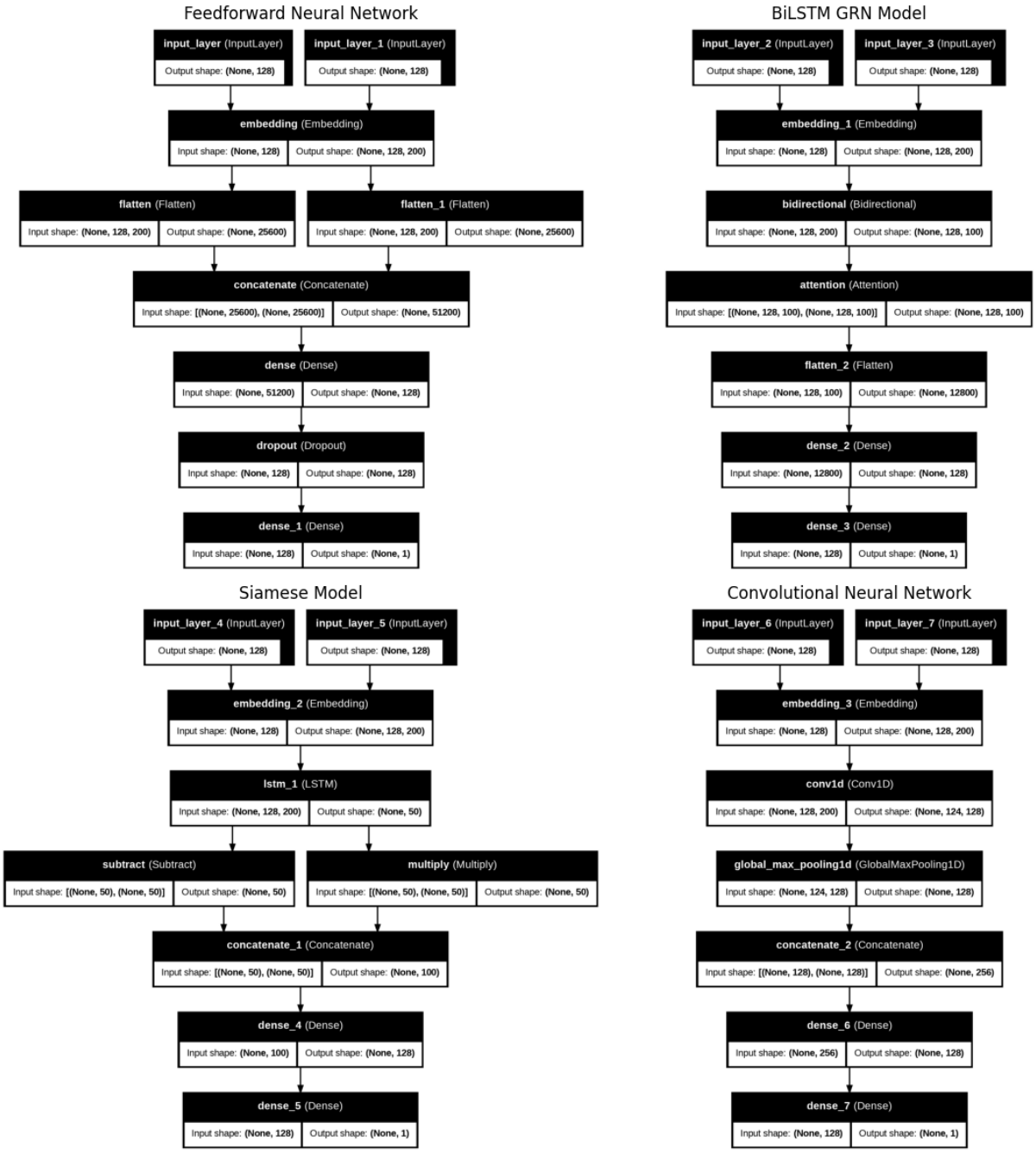


Figure 4.1

### 3.4 Training Setup

- **Optimizer**: Adam.
- **Loss Function**: Binary Crossentropy.
- **Batch Size**: 64.
- **Epochs**: 10 with early stopping.
- **Callbacks**: Model checkpointing, learning rate reduction, and early stopping based on validation loss.

## 4. Experiments and Results

### 4.1 Evaluation Metrics

The models were evaluated on the following metrics:

- **Accuracy**: Fraction of correct predictions.
- **Precision**: Fraction of true positives among all positive predictions.
- **Recall**: Fraction of true positives among all actual positives.
- **F1-Score**: Harmonic mean of precision and recall.

### 4.2 Results Table

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| **FFN** | 66.85% | 58.77% | 36.07% | 44.70% |
| **Bi-LSTM + GRN** | 71.20% | 60.58% | 64.33% | 62.40% |
| **Siamese Network** | 71.10% | 65.25% | 47.51% | 54.98% |
| **CNN** | 66.50% | 54.17% | 63.80% | 58.59% |

### 4.3 Visualization

Figure 4.3. Architectures and Performance Metrics of Four Models: (a) Feedforward Neural Network, (b) Bi-LSTM with GRN, (c) Siamese Network, and (d) Convolutional Neural Network. Each subfigure displays the model's structure alongside its accuracy and loss values.

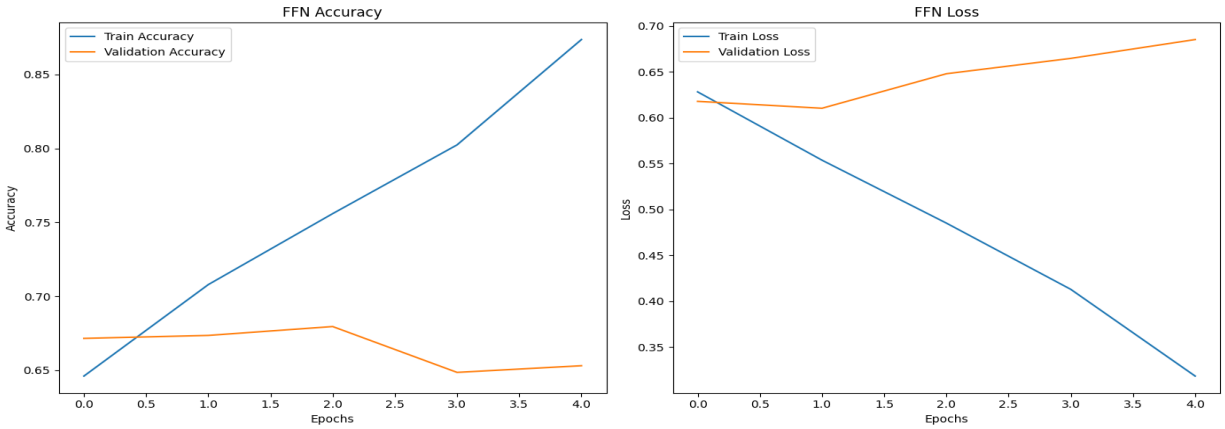


Figure 4.3(a)

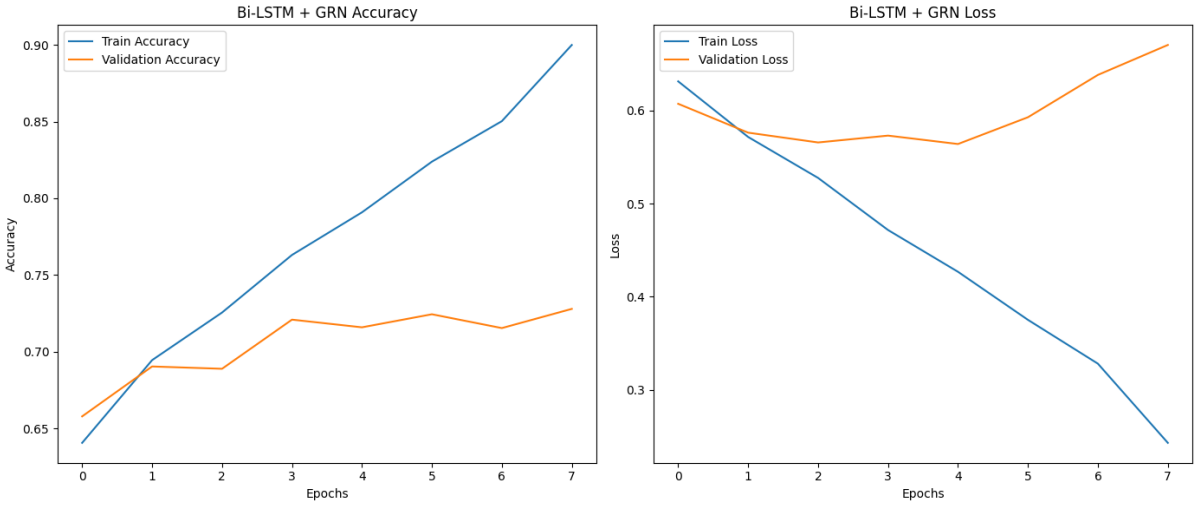


Figure 4.3(b)

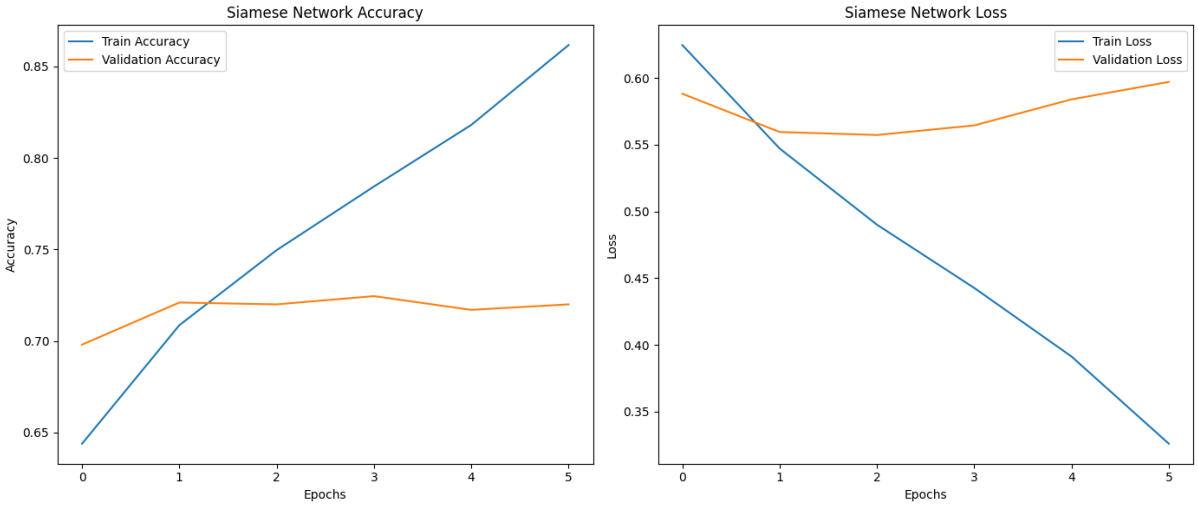


Figure 4.3(c)

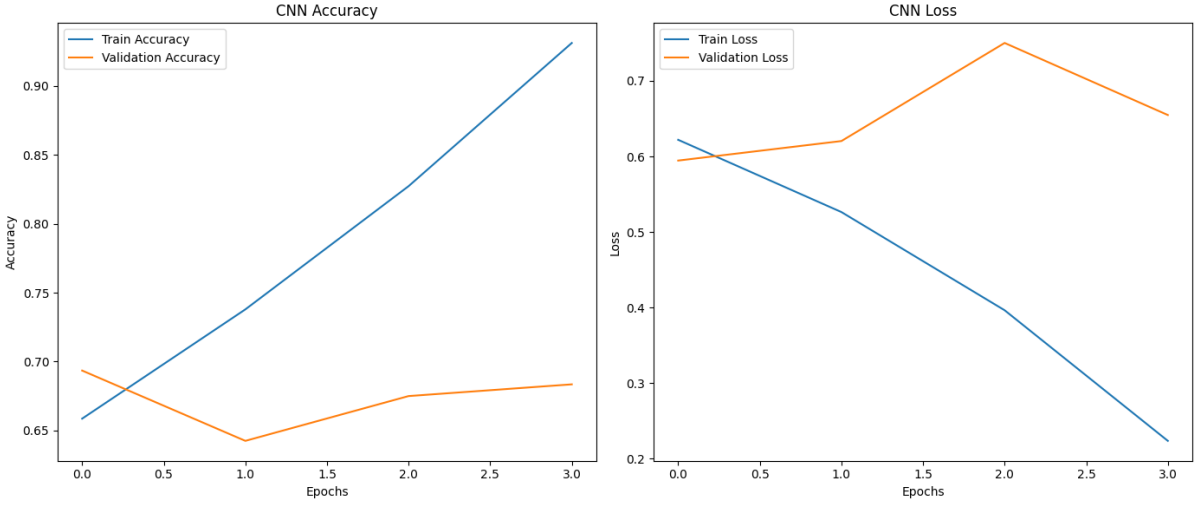


Figure 4.3(d)

---

## 5. Analysis and Discussion

### 5.1 Error Analysis

- **Common Errors**: Models often misclassified paraphrases involving idiomatic expressions and ambiguous phrases.
- **Impact of Imbalanced Data**: The dataset's imbalance affected the recall for the minority class ("Duplicate"), leading to overemphasis on the "Not Duplicate" class.

### 5.2 Model Insights

- **Bi-LSTM GRN**: Excelled in understanding sequential and contextual information.
- **Siamese Network**: Effective for comparing similar embeddings but struggled with subtle semantic differences.
- **CNN**: Captured structural patterns but lacked depth for semantic equivalence.
- **FFN**: Limited in capturing relationships beyond simple patterns.

---

## 6. Conclusion and Future Work

### 6.1 Conclusion

This study underscores the efficacy of the Bidirectional Long Short-Term Memory with Gated Relevance Network (Bi-LSTM GRN) in paraphrase identification, achieving an F1-Score of 62.4%. The model's ability to capture bidirectional contextual information and semantic interactions between text segments highlights the significance of context-aware architectures in understanding nuanced language patterns. Additionally, the implementation of a robust preprocessing pipeline has proven essential in enhancing model performance, emphasizing the critical role of data quality in natural language processing tasks.

### 6.2 Future Work

Future directions include:

To further enhance paraphrase identification, future research should focus on:

- **Implementing Transformer-Based Models:** Leveraging models like BERT can improve performance by effectively capturing contextual nuances.
- **Extending to Multilingual Datasets:** Applying current methodologies to datasets in various languages will assess model generalizability across linguistic contexts.
- **Developing Ensemble Models:** Combining the strengths of CNNs and Bi-LSTMs can lead to more robust paraphrase detection systems.

## 7. References


1. Zhou, C., Qiu, C., & Acuna, D. E. (2024). "Paraphrase Identification with Deep Learning: A Review of Datasets and Methods."

2. Yin, W., & Schutze, H. (2015). "Convolutional Neural Network for Paraphrase Identification." Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).

3. Lan, W., & Xu, W. (2018). "Neural Network Models for Paraphrase Identification, Semantic Textual Similarity, Natural Language Inference, and Question Answering."

4. Peinelt, N., Nguyen, D., & Liakata, M. (2020). "Better Early than Late: Fusing Topics with Word Embeddings for Neural Question Paraphrase Identification."

5. Vrbanec, T., & Mestrovic, A. (2021). "Corpus-Based Paraphrase Detection Experiments and Review.".