Islamic University Journal of Applied Sciences (IUJAS)
https://journals.iu.edu.sa/jesc

Volume VIII, Issue II, December 2026, Pages 22-38

A Multi-Similarity Neural Network for Paraphrase Detection

Emad Nabil*

Computer Science Department, Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah 42351, KSA, emadnabil@iu.edu.sa

AI Center, Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah 42351, KSA

*Corresponding Author: (Emad Nabil), *Email Address*: Emadnabil@iu.edu.sa

Abstract

This study introduces a multi-similarity neural network framework for paraphrase detection, an important task in natural language processing that identifies whether two sentences convey the same meaning using different expressions. The proposed method combines various similarity measures, such as string-based similarity, semantic similarity, and embedding-based similarity, with a deep learning classifier. The framework is structured as a three-phase pipeline: preprocessing, extraction of multiple similarity features, and classification through a neural network. It employs more than 168 string similarity algorithms, semantic measures derived from WordNet, and several pre-trained embedding models to compute similarity scores. These features are aggregated and supplied to a deep neural network to determine whether sentence pairs are paraphrases. The model was evaluated on the Microsoft Research Paraphrase Corpus (MSRP) using accuracy and F1-score as performance metrics. The experimental results indicate that the proposed framework achieves 81.74% accuracy and an F1 Score of 86.6%, surpassing several existing approaches. Overall, the results suggest that integrating diverse similarity measures with neural networks enhances the identification of both explicit and nuanced paraphrases, thereby supporting advancements in text analysis and plagiarism detection systems.

Keywords: Paraphrase Detection; Similarity Measures; Deep Learning; Sentence Embeddings

https://doi.org/10.63070/jesc.2026.032
Received 06 April 2026; Revised 17 May 2026; Accepted 04 June 2026; Available online 15 July 2026.
Published by Islamic University of Madinah on behalf of ***Islamic University Journal of Applied Sciences.***
This is an open access article published under the Creative Commons Attribution 4.0 International license (CC BY 4.0):https://creativecommons.org/licenses/by/4.0/

1. Introduction

The principal aim of Natural Language Processing (NLP) is to equip computers with the ability to comprehend and produce human language. This technology forms the backbone of numerous everyday automated systems, from conversational agents to tools that ensure academic integrity. A critical element within this field is Paraphrase Detection (PD), which focuses on identifying whether two separate text passages express the same core meaning. Recognizing this semantic equivalence is fundamental to more complex operations, such as data mining, document retrieval, and plagiarism detection.

Applications like Turnitin depend significantly on this functionality. They assist educators in confirming the authenticity of student work by highlighting content that may have been dishonestly copied from peers or online sources, even when the original wording has been modified. Because individuals naturally convey identical thoughts through a diverse range of words and sentence structures, identifying these altered versions remains the most crucial and difficult task in plagiarism analysis. Consequently, improving an AI's capability to link varied expressions of the same idea is vital for attaining advanced linguistic proficiency.

To tackle this problem, this study introduces a hybrid approach for detecting semantic similarities. The proposed framework combines artificial neural networks with a multi-level evaluation of text comparability, specifically examining structural strings, meaning-based relations, and vector embeddings.

The proposed framework combines multiple similarity techniques, including string similarity, semantic similarity, and embedding-based similarity, within a unified neural network architecture for paraphrase detection. The framework follows a three-stage pipeline consisting of preprocessing, similarity feature extraction, and neural-network-based classification.

The main contributions of this study are summarized as follows: (1) integrating a large set of heterogeneous similarity measures into a single framework, (2) investigating the impact of different preprocessing strategies on paraphrase detection performance, (3) evaluating several deep learning architectures to determine the optimal model configuration, and (4) achieving competitive performance on the MSRP benchmark dataset.

The remainder of this paper is structured as follows: Section 2 provides an overview of existing research on semantic analysis. Section 3 describes the architecture of our newly developed model. Section 4

explains the experimental design and presents the resulting data. Section 5 discusses the implications of these empirical findings.

2. Literature Review

This literature review focuses specifically on traditional paraphrase detection approaches that are closely related to the proposed framework, particularly supervised and unsupervised methods based on similarity measures, machine learning, and deep learning techniques. The literature review spans prior work on paraphrase detection (PD), mainly categorized into **unsupervised** and **supervised** approaches. Most studies rely on the **Microsoft Research Paraphrase Corpus (MSRP)** dataset. Unsupervised methods typically employ similarity measures, while supervised methods integrate similarity features with machine learning and deep learning models. Figure 1 illustrates the main PD approaches.

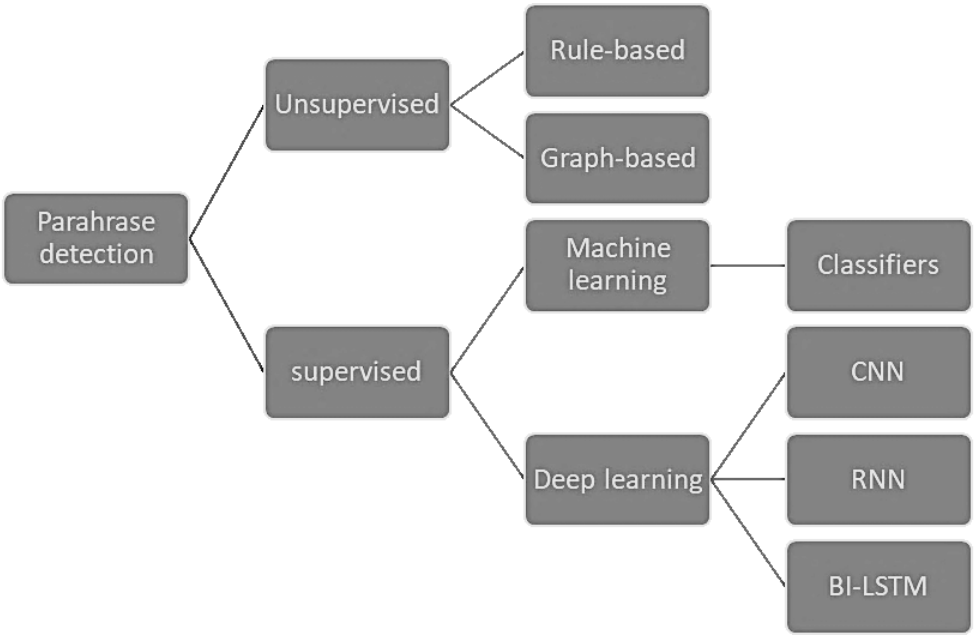


Figure 1. Paraphrase detection techniques

2.1. Unsupervised Methods

In [1], unsupervised techniques such as **Latent Semantic Analysis (LSA)**, WORDNET-based similarities (Lesk, Lin, JCN), semantic measures (Resnik, J&C, L&C), cosine similarity, and TF-IDF were applied. In [2], similarity was computed using Wikipedia-based encyclopedic knowledge, latent semantic space,

salient semantic space, and explicit semantic space methods. In [3], sentences were converted into graphs through tokenization, lemmatization, and parsing. Dependency graphs were generated and compared using Graph Subsumption to determine paraphrasing.

In [4], a Semantic Text Similarity (STS) model combined string-based methods such as Longest Common Subsequence (LCS) and Normalized LCS (NLCS) with semantic similarity matrices. In [5], sentences were represented as vectors and compared using cosine similarity against a predefined threshold. In [6], matrix similarity techniques were applied using various WordNet-based measures (JCN, LCH, LESK, LIN, RES, WUP). Table 1 summarizes these unsupervised approaches, reporting reference, method, accuracy, and F1-score.

Table 1. Different Approaches of Unsupervised Methods

Ref No.	Methods	Rule-based	Graph-based	Accuracy	F-1 score
[1]	Cosine similarity with TF-IDF weighting	✓		65.4%	75.3%
	Combination of several word similarity measures			70.3%	81.3%
[2]	Explicit semantic space	✓		67.0%	79.3%
	Latent semantic space			68.8%	79.9%
	Salient semantic space			72.5%	81.4%
[3]	Converting text to a graph to utilize the graph Subsumption method		✓	70.6%	80.5%
[4]	Combination of semantic and string similarity	✓		72.6%	81.3%
[5]	Additive composition of vectors and cosine distance	✓		73.0%	82.0%
[6]	JCN WordNet similarity with matrix	✓		74.1%	82.4%

2.2. Supervised Methods

Supervised approaches extract features and apply machine learning or deep learning classifiers. In [7], string (n-grams, skip-grams, LCS) and WORDNET-based semantic features were extracted and classified using SVM, KNN, and MaxEnt. In [8], sentence similarity and dissimilarity were combined using predicate extraction (Charniak parser), thesaurus-based similarity, and a dissimilarity classifier. In [9], POS tagging and stop-word removal were followed by LCS and WORDNET feature extraction, then classified using logistic regression. In [10], three vector representations were used and evaluated with distributional models (DM, NLM, RAE, SDS), where SDS achieved the best results. In [11], a TF-KLD weighting scheme with fine-grained features (unigram, bigram similarity) was classified using SVM. In [12], string and skip-thought similarities were combined and evaluated in Weka; VotedPerceptron

achieved the best performance. In [13], machine translation metrics (WER, BLEU, PER, NIST) were used as features for SVM classification. In [14], 17 features (n-gram, surface, dependency, tree-edit distance) were tested with multiple classifiers; SVM outperformed others.

In [15], recursive autoencoders were combined with dynamic pooling and softmax classification. In [16], machine translation evaluation metrics (BLEU, NIST, TER, METEOR, etc.) were classified using Logistic Regression, SMO, and KNN. In [17], CNN-based sentence modeling with pooling was combined with similarity measurement layers. In [18], kernel-based methods (NSPDK, PTK, SPTK) were applied to tree, graph, and feature representations. In [19], RecNN and RNN were integrated into a Dynamic Disambiguation Model and fed into a Siamese network. In [20], BI-LSTM outputs were combined with handcrafted features and classified via dense and softmax layers. In [21], the DPIM-ISS model used CNNs to extract paraphrase matching patterns and joint semantic-syntactic representations. In [22], preprocessed sentences were evaluated using three similarity measures and classified with Linear SVC. Table 2 summarizes these supervised studies, including reference, method, accuracy, and F1-score. Overall, unsupervised approaches focus primarily on similarity computation, whereas supervised methods integrate similarity features with advanced machine learning and deep learning models, achieving improved performance.

Although previous studies achieved promising results using either similarity-based approaches or deep learning models, several limitations remain. Many existing methods rely on a limited number of similarity features or focus on a single representation technique. Furthermore, insufficient attention has been given to combining diverse similarity measures with optimized neural architectures in a unified framework. These limitations motivate the proposed approach, which integrates multiple complementary similarity techniques with deep learning to improve paraphrase detection performance.

3. Approach

Based on the comprehensive analysis of the existing literature, it is evident that machine learning (ML) techniques, particularly when bolstered by proficient feature selection methodologies, have demonstrated superior performance outcomes. Building upon this precedent, this research endeavors to investigate the synergistic amalgamation of feature extraction strategies with the potency inherent in deep learning (DL) paradigms. This investigation is grounded in the premise that DL, as

substantiated by numerous empirical instances, possesses the capacity to surpass the performance benchmarks established by ML. Additionally, the perceived augmentative influence of meticulous data preprocessing on performance enhancement further underscores our research objectives. In light of these considerations, the conceptual framework presented in Figure 2 is posited as our proposed approach.

Table 2. Different models of supervised methods for paraphrasing detection

Ref. No.	Method	ML	DL	Accuracy	F-1 score
[7]	combination of lexical and semantic features	✓		76.6%	79.6%
[8]	sentence dissimilarity classification	✓		72.0%	81.6%
[9]	PI using semantic heuristic features	✓		74.7%	81.8%
[10]	simple distributional semantic space		✓	73.0%	82.3%
[11]	Matrix factorization with supervised reweighting	✓		80.4%	85.9%
[12]	Seven similarity techniques with the Voted-Perceptron classifier	✓		76.6%	83.5%
[13]	combination of MT evaluation measures as features	✓		75.0%	82.7%
[14]	dependency-based features	✓		75.6%	83.0%
[15]	recursive autoencoder with dynamic pooling		✓	76.8%	83.6%
[16]	combination of eight machine translation metrics	✓		77.4%	84.1%
[17]	Multi-perspective Convolutional NNs and structured similarity layer		✓	78.6%	84.7%
[18]	Combination of Convolution Kernels and similarity scores		✓	79.1%	85.2%
[19]	Recursive NNs using syntax-aware multi- sense word embeddings		✓	78.6%	85.3%
[20]	BI-LSTM and Hand-Crafted features		✓	79.2%	85.4%
[21]	CNN with semantic and string features		✓	73.57%	83.55%
[22]	Classical machine learning classifiers with text similarity scores	✓		75.78%	83.01%

The proposed method for paraphrase detection comprises a sequential process delineated in Figure 2. The initial phase entails data preprocessing, although, in some instances, this step was omitted. In the subsequent step, the processed or raw data undergoes diverse similarity techniques to derive corresponding similarity scores. Subsequently, these scores are utilized as inputs for deep learning classifiers to discern the paraphrastic nature of the sentences.

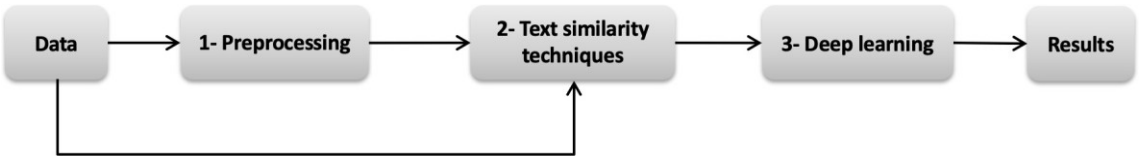


Figure 2. The proposed approach pipeline

The initial step, designated as step one, encompasses preprocessing operations. This encompasses the conversion of sentences to lowercase, followed by the application of stemming and lemmatization techniques. After this, the elimination of stop words is performed. Stemming and lemmatization are instrumental in reverting the words within the sentence to their original root forms.

During the second step, the preprocessed or raw sentences are subjected to similarity methodologies. Our exploration encompassed a diverse array of similarity techniques sourced from various repositories. Specifically, we examined the Abydos Python library, which boasts an assemblage of more than 200 string similarity methodologies.

Moreover, scrutiny extended to the NLTK and Spacy libraries [23], wherein provisions for semantic similarity methodologies such as WUP [24] and LIN [25] , alongside supplementary semantic similarity techniques, are made available. Moreover, we employed the Sentence Transformers libraries, incorporating multiple pre-trained models designed for embedding similarity analysis. Notable among these models are BERT-BASE-NLI-MEAN-TOKENS [26] and ALL- MPNET-BASE-V2, alongside other variants [27].

The used pre-trained models are described in Table 3. These models are engineered to transform pairs of sentences into respective vector representations, subsequently facilitating comparative assessments for similarity. The resultant similarity scores are bounded within the range of -1 to 1, wherein a score of 1 signifies utmost similarity, while a score of -1 signifies maximal dissimilarity. The conclusive step, denoted as step three, involves the aggregation of the computed similarity scores, which are subsequently channeled into deep learning classifiers to report if two sentences are paraphrased or not.

The embedding models presented in Table 3 differ in their training objectives, architectural designs, and vector dimensionalities, all of which influence their ability to capture semantic relationships between sentence pairs. Some models generate lower-dimensional embeddings that offer faster computation and lower memory requirements, while higher-dimensional embeddings generally provide richer semantic representations at the cost of increased computational complexity. In this study, multiple embedding models with varying vector dimensions were evaluated to ensure diversity in contextual

feature extraction. The integration of these embeddings with string-based and semantic similarity measures enhanced the framework's capability to capture both lexical and contextual similarities between sentences.

Table 3. Embedding similarity pre-trained models

Model Name	Output Vector Size	Language	Training data
bert-base-nli-mean-tokens [23]	768	English	Not mentioned
all-mpnet-base-v2 [23]	768	English	1B+ training pairs.
multi-qa-mpnet-base-dot-v1 [23]	768	multilingual	215M (question, answer) pairs from diverse sources.
all-distilroberta-v1 [23]	768	English	1B+ training pairs.
all-MiniLM-L12-v2 [23]	384	English	1B+ training pairs.
multi-qa-distilbert-cos-v1 [23]	768	multilingual	215M (question, answer) pairs from diverse sources
all-MiniLM-L6-v2 [23]	384	English	1B+ training pairs
paraphrase-xlm-r- multilingual-v1 [23], [24]	768	multilingual	data for 50+ languages
LaBSE [23], [24], [25]	768	multilingual	109 languages
paraphrase-mpnet-base-v2 [23]	768	English	12 datasets
paraphrase-multilingual- MiniLM-L12-v2 [23], [24]	384	multilingual	50+ languages
paraphrase-MiniLM-L6-v2 [23]	384	English	12 datasets
multi-qa-MiniLM-L6-cos-v1 [23]	384	multilingual	215M (question, answer) pairs from diverse sources.
stsb-roberta-large [23]	1024	English	Not mentioned
Universal Sentence Encoder	512	English	Not mentioned
fasttext cc.en.300.bin [26],[27],[28]	300	English	4 datasets

4. Results

The same dataset was used throughout all the experiments. As per the best practices in ML/DL, we applied a train-test validation method with 70% training and 30% validation, and this is given in the dataset itself as a training file and a testing file. Evaluation Metrics We used the accuracy and the F-1 score as they are standard evaluation metrics applied to the paraphrase detection problem. The accuracy score, explained in Eq. 1, represents the number of occurrences of the data that were properly categorized, whereas the F-1 score [28], as explained in Eq. 2, represents the correctness of the test. The F1-score is calculated from the precision (Eq. 3) and the recall (Eq. 4)

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad \text{Eq.1}$$

$$F1 = \frac{2\times Percision \times Recall}{Percision + Recal} \qquad \text{Eq. 2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad \text{Eq. 3}$$

$$\text{Recall} = \frac{TP}{TP+FN} \qquad \text{Eq. 4}$$

a. Dataset

This study utilizes the **Microsoft Research Paraphrase Corpus (MSRP)** [29] dataset. The MSRP contains 5,801 sentence pairs extracted from online news articles, with one sentence selected from each article. Among these pairs, 3,900 (67%) are annotated as paraphrases. Each sentence pair is labeled by human annotators to indicate whether the two sentences are semantically equivalent.As shown in Table 4, the dataset includes two text fields (#1 string and #2 string) covering diverse topics such as economics and legal cases, ensuring lexical and contextual variety. The dataset also includes the source article ID, which is excluded in this study as it is not required, and a final label indicating whether the sentence pair is paraphrased.

Table 4 presents a sample from the Microsoft Research Paraphrase (MSRP) dataset.

#1 String	#2 String	#1 ID	#2 ID	Quality
Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence.	Referring to him as only "the witness", Amrozi accused his brother of deliberately distorting his evidence.	702876	702977	1
Yucaipa owned Do- minick's before selling the chain to Safeway in 1998 for $2.5 billion.	Yucaipa bought Dominick's in 1995 for $693 million and sold it to Safeway for $1.8 billion in 1998.	2108705	2108831	0

b. Experiments

This research investigates paraphrase detection using a multifaceted approach that combines various text similarity techniques with deep learning models of different architectures. To comprehensively evaluate the effectiveness of the proposed framework, experiments were conducted using multiple preprocessing settings, including lowercasing, stemming, lemmatization, stop-word removal, as well as scenarios without any preprocessing. The experimental results presented in the following tables demonstrate the impact of these different settings on paraphrase detection performance.

The experiments followed a common three-stage paraphrase detection pipeline. First, the text data undergoes a preprocessing stage depending on the selected experimental configuration. Next, the

processed or raw sentence pairs are evaluated using three main categories of similarity techniques: string similarity, semantic similarity, and embedding similarity. These techniques generate similarity scores that capture lexical, semantic, and contextual relationships between the sentence pairs. Finally, the extracted similarity scores are aggregated and used as input features for neural network classifiers to determine whether the sentence pairs are paraphrases.

To identify the optimal model architecture, several neural network configurations were evaluated using different combinations of similarity features and preprocessing strategies. The experiments involved varying the number of layers, the number of neurons, and the structure of the neural network models. The final selected architecture achieved the best overall performance in terms of accuracy and F1-score while maintaining effective feature representation and classification capability. The results indicate that integrating diverse similarity measures within parallel dense layers enhances the model's ability to capture both lexical and semantic relationships between sentence pairs.

4.1.The First Experimental Approach

Table 5 shows the preprocessing with lemmatization, removing the stop words, and lower casing the entire data, and the combination of the text similarity techniques will be as follows: 168 string similarity techniques, 8 pre-trained BERT models, and lastly Wordnet semantic similarity. The models were used from the TensorFlow library in Python. Table 5 consists of four columns: accuracy, F-1 score, number of parameters, and Layers

Table 5. Results of preprocessing (lemmatization, removing the stop words, and lower casing)

Accuracy	F-1 score	Parameters number	Layers (number of neurons, activation function)
0.76058	0.82941	57,921	2 Dense and 1 concatenate (256,relu) (64,sigmoid)
0.75942	0.82744	92,673	2 Dense and 1 concatenate (256,relu) (256,sigmoid)
0.76174	0.82724	185,345	2 Dense and 1 concatenate (512,relu) (512,sigmoid)

Table 6 shows the preprocessing with stemming, removing the stop words, and lower casing the entire data, and the combination of the text similarity techniques will be as follows: 168 string similarity techniques and 8 pre-trained BERT models, and the Models were used from the TensorFlow library in Python. Table 6 consists of four columns, namely, accuracy, F-1 score, number of parameters, and Layers. From the results in Tables 5 and 6 , it is noticed that the first approach did not achieve better results.

Table 6. Results of preprocessing (stemming, removing the stop words and lower casing)

Accuracy	F-1 score	Parameters number	Layers (number of neurons, activation function)
0.75362	0.82717	56,961	2 Dense and 1 concatenate (256,relu) (64,sig- moid)
0.75246	0.82521	91,137	2 Dense and 1 concatenate (256,relu) (256,sigmoid)
0.75362	0.81657	182,273	2 Dense and 1 concatenate (512,relu) (512,sigmoid)

4.2. The Second Experimental Approach

Table 7 shows the results without any preprocessing done on the dataset, and the text similarity combination will be as follows: 168 string similarity techniques, 14 pre-trained BERT models, Spacy semantic similarity, Universal sentence encoder pre-trained model, and lastly the fasttext pre-trained model. Table 7 consists of four columns: accuracy, F-1 score, number of parameters, and Layers

Table 7. Results of the no-preprocessing approach

Accuracy	F-1 score	Parameters number	Layers (number of neurons, activation function)
0.81391	0.86381	59,841	2 Dense and 1 concatenate (256,relu) (64,sigmoid)
0.81043	0.86347	95,745	2 Dense and 1 concatenate (256,relu) (256,sigmoid)
0.81739	0.86613	191,489	2 Dense and 1 concatenate (512,relu) (512,sigmoid)

In the previous section of experiments, we viewed most of the results of approaches 282 from no preprocessing DL as well as preprocessing and DL, we mentioned the ones that 283 were worth mentioning in the mentioned approaches to view the difference when changing 284 any of the experiment settings. Through experiments, we tried to test the effect of using ensemble models of text similarity techniques as input to our deep learning model, and our highest results were by the pipeline that can be seen in Figure 3. The pipeline goes as inputting the raw data in parallel into the 168 string similarity techniques, 16 pre-trained embedding models, and 1 semantic similarity technique, and after this stage, one vector consisting of 186 similarity scores is produced for each pair of sentences. Then this vector is input into two 2 Dense layers parallel to each other to extract features. After that, the two extracted feature sets are concatenated to produce one final feature set that will be used to predict whether the pair of sentences is plagiarized or not. This pipeline yielded the best results with an accuracy of 81.74% and an F-1 score of 86.6%. And as shown in Figure 4, the best Neural Network.

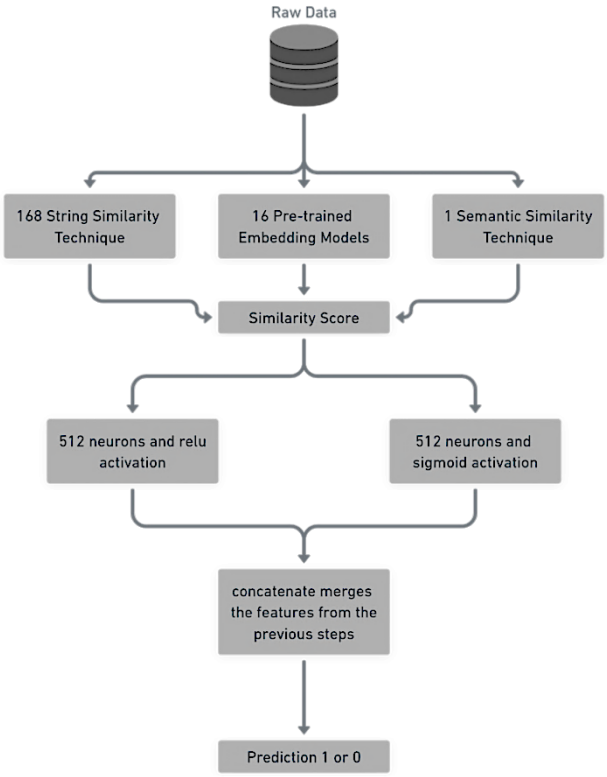


Figure 3. Proposed paraphrase detection pipeline of the best-performing model.

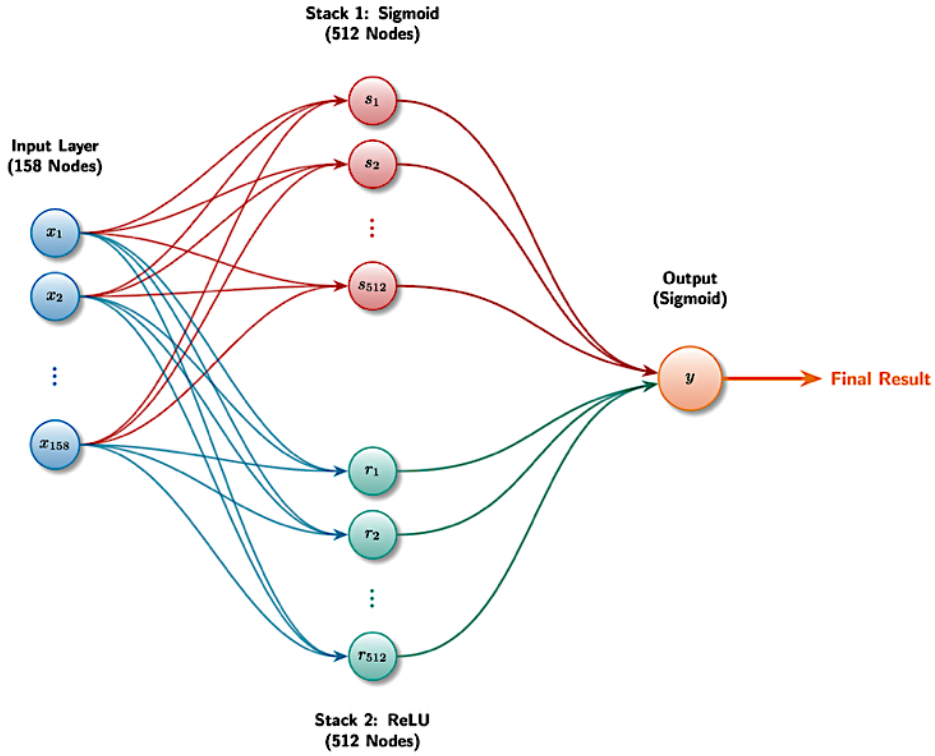


Figure 4. Neural Network Architecture of the best-performing paraphrase detection model.

In Table 8, we compare the proposed approach with the different old researchers' approaches. After examining the results in Table 8, it is deduced that the proposed approach outperforms the previous researchers' works.

Table 8. Comparison between the proposed model and previous works

Ref.	Name	Accuracy	F-1 Score
[11]	TF-KLD	80.4%	85.9%
[20]	BI-LSTM and Hand-Crafted features	79.2%	85.4%
[18]	REL-TK	79.1%	85.2%
[17]	Multi-Perspective CNN	78.6%	84.7%
[19]	SAMS-RecNN	78.6%	85.3%
[16]	MTMETRICS	77.4%	84.1%
[15]	SHPNM	76.8%	83.6%
[7]	KM	76.6%	79.6%
[12]	CombineBest	76.6%	83.5%
[22]	Classical machine learning classifiers with text similarity scores	75.78%	83.01%
[14]	WDDP	75.6%	83.0%
[13]	FHS	75.0%	82.7%
[9]	ParaDetect	74.7%	81.8%
[6]	matrixJcn	74.1%	82.4%
[21]	DPIM-ISS	73.57%	83.55%
[5]	MSR addition	73.0%	82.0%
[10]	SDS	73.0%	82.3%
[4]	STS	72.6%	82.0%
[8]	QKC	72.0%	81.6%
[3]	RMLMG	70.6%	81.3%
[2]	ESA	67.0%	79.3%
	LSA	68.8%	79.9%
	SSA	72.5%	81.4%
[1]	Vector base similarity (Baseline)	65.4%	75.3%
	MCS	70.3%	81.3%
Proposed Model		**81.74%**	**86.6%**

5. Discussion

It is preserved from the results that inputting the data directly to the text similarity techniques gives better results than preprocessing the data. It is thought that this happens because the preprocessing may remove some important information from the data, and that information helps the Neural Network to understand the sentences better. The integration of 5 neural networks with a diverse array of similarity techniques, including string, semantic, and embedding similarity, represents a new approach in paraphrase detection, resulting in state-of-the-art performance. This innovative strategy outperforms traditional methods by significantly extending the model's capabilities in identifying various forms of paraphrasing, ranging from direct copies to nuanced rephrasing. Neural networks

enhance accuracy by deciphering intricate linguistic patterns and semantic nuances. The incorporation of contextual awareness, facilitated by string, semantic, and embedding techniques, allows for the detection of subtle paraphrased instances, enabling adaptability in an ever-evolving landscape of textual variations. Furthermore, its versatility positions it as an effective tool for deployment across a multitude of domains, from NLP applications to content generation, acting as a robust guardian against the proliferation of paraphrased content.

6. Conclusions

The success of this research is indicative of the growing importance of effective paraphrase detection, as the digital age has witnessed an exponential increase in the generation and dissemination of textual content. With the prevalence of information-sharing platforms, content creation, and academic publications, the need for reliable mechanisms to discern paraphrased content from genuine originality has never been more pronounced. The achievement of an 81.74% accuracy rate and an F-1 score of 86.6% not only underscores the proficiency of this approach but also signifies a substantial step toward addressing a complex and pervasive issue in the world of text analysis. As digital communication continues to evolve, paraphrase detection plays a pivotal role in maintaining the integrity of information, academic work, and content creation. Moreover, this research contributes significantly to the broader context of natural language processing, as it enriches the toolbox of techniques available for text analysis. The integration of neural networks with diverse similarity measures demonstrates the potential of combining various methodologies to tackle multifaceted challenges in textual analysis. This not only serves the specific goal of paraphrase detection but also has implications for advancing the field's capabilities in understanding and processing textual data more comprehensively. In essence, the findings of this research showcase the capacity of innovative approaches to tackle intricate problems in natural language processing, offering a promising solution to a persistent issue while extending the horizons of text analysis methodologies.

For future work, the proposed framework can be extended and evaluated on additional benchmark datasets to further examine its robustness and generalizability across different domains and text styles. Moreover, applying the framework to the Arabic language represents a promising research direction due to the linguistic complexity and rich morphology of Arabic text. Future studies may also investigate the integration of recent Large Language Models (LLMs) and transformer-based architectures to further enhance paraphrase detection performance.

Funding: This work was supported by the Deanship of Scientific Research, Islamic University of Madinah, Saudi Arabia.

References

[1] Mihalcea, R.; Corley, C.; Strapparava, C. Corpus-based and knowledge-based measures of text semantic similarity. In Proceedings of the AAAI, July 2006, Vol. 6, pp. 775–780.

[2] Hassan, S. Measuring semantic relatedness using salient encyclopedic concepts. PhD thesis, University of North Texas, 2011.

[3] Rus, V.; McCarthy, P.M.; Lintean, M.C.; McNamara, D.S.; Graesser, A.C. Paraphrase Identification with Lexico-Syntactic Graph Subsumption. In Proceedings of the FLAIRS conference, May 2008, pp. 201–206.

[4] Islam, A.; Inkpen, D. Semantic similarity of short texts. Recent Advances in Natural Language Processing V 2009, 309, 227–236.

[5] Milajevs, D.; Kartsaklis, D.; Sadrzadeh, M.; Purver, M. Evaluating neural word representations in tensor-based compositional settings. arXiv preprint arXiv:1408.6179 2014.

[6] Fernando, S.; Stevenson, M. A semantic similarity approach to paraphrase detection. In Proceedings of the Proceedings of the 11th annual research colloquium of the UK special interest group for computational linguistics, March 2008, pp. 45–52.

[7] Kozareva, Z.; Montoyo, A. Paraphrase identification on the basis of supervised machine learning techniques. In Proceedings of the International conference on natural language processing (in Finland). Springer, August 2006, pp. 524–533.

[8] Qiu, L.; Kan, M.Y.; Chua, T.S. Paraphrase recognition via dissimilarity significance classification. In Proceedings of the Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, July 2006, pp. 18–26.

[9] Ul-Qayyum, Z.; Altaf, W. Paraphrase identification using semantic heuristic features. Research Journal of Applied Sciences, Engineering and Technology 2012, 4, 4894–4904.

[10] Blacoe, W.; Lapata, M. A comparison of vector-based representations for semantic composition. In Proceedings of the Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, July 2012, pp. 546–556.

[11] Ji, Y.; Eisenstein, J. Discriminative improvements to distributional sentence similarity. In 367 Proceedings of the Proceedings of the 2013 conference on empirical methods in natural language 368 processing, October 2013, pp. 891–896.

[12] El Desouki, M.I.; Gomaa, W.H.; Abdalhakim, H. A hybrid model for paraphrase detection combines pros of text similarity with deep learning. Int. J. Comput. Appl 2019, 975, 8887.

[13] Finch, A.; Hwang, Y.S.; Sumita, E. Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In Proceedings of the Proceedings of the third international workshop on paraphrasing (IWP2005), 2005.

[14] Wan, S.; Dras, M.; Dale, R.; Paris, C. Using dependency-based features to take the'para-farce'out of paraphrase. In Proceedings of the Proceedings of the Australasian language technology workshop 2006, November 2006, pp. 131–138.

[15] Socher, R.; Huang, E.H.; Pennington, J.; Ng, A.Y.; Manning, C.D. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In Proceedings of the NIPS, December 2011, Vol. 24, pp. 801–809.

[16] Madnani, N.; Tetreault, J.; Chodorow, M. Re-examining machine translation metrics for paraphrase identification. In Proceedings of the Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, June 2012, pp. 182–190.

[17] He, H.; Gimpel, K.; Lin, J. Multi-perspective sentence similarity modeling with convolutional neural networks. In Proceedings of the Proceedings of the 2015 conference on empirical methods in natural language processing, September 2015, pp. 1576–1586.

[18] Filice, S.; Da San Martino, G.; Moschitti, A. Structural representations for learning relations between pairs of texts. In Proceedings of the Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), July 2015, pp. 1003–1013.

[19] Cheng, J.; Kartsaklis, D. Syntax-aware multi-sense word embeddings for deep compositional models of meaning. arXiv preprint arXiv:1508.02354 2015.

[20] Shahmohammadi, H.; Dezfoulian, M.H.; Mansoorizadeh, M. Paraphrase detection using LSTM networks and handcrafted features. Multimedia Tools and Applications 2021, 80, 6479–6492.

[21] Kong, L.; Han, Z.; Han, Y.; Qi, H. A deep paraphrase identification model interacting semantics with syntax. Complexity 2020.

[22] Hany, M.; Gomaa, W.H. A Hybrid Approach to Paraphrase Detection Based on Text Similarities and Machine Learning Classifiers. In Proceedings of the 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC). IEEE, May 2022, pp. 343–348.

[23] Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 2019.

[24] Reimers, N.; Gurevych, I. Making monolingual sentence embeddings multilingual using knowledge distillation. arXiv preprint arXiv:2004.09813 2020.

[25] Feng, F.; Yang, Y.; Cer, D.; Arivazhagan, N.; Wang, W. Language-agnostic BERT sentence embedding. arXiv preprint arXiv:2007.01852 2020.

[26] Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. Transactions of the association for computational linguistics 2017, 5, 135–146.

[27] Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 2016.

[28] Joulin, A.; Grave, E.; Bojanowski, P.; Douze, M.; Jégou, H.; Mikolov, T. Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612.03651 2016.

[29] El Desouki, M.I.; Gomaa, W.H. Exploring the recent trends of paraphrase detection. International Journal of Computer Applications 2019, 975.