Urdu Sentential Paraphrased Plagiarism Detection Using Large Language Models

HAFIZ RIZWAN IQBAL, Information Technology University (ITU), Arfa Software Technology Park, Pakistan
MUHAMMAD SHARJEEL, COMSATS University Islamabad (CUI), Lahore Campus, Pakistan
JAWAD SHAFI, COMSATS University Islamabad (CUI), Lahore Campus, Pakistan
USAMA MEHMOOD, Information Technology University (ITU), Arfa Software Technology Park, Pakistan
AGHA ALI RAZA, Lahore University of Management Sciences (LUMS), Pakistan

Plagiarism, the unauthorized reuse of text, fueled by the ease of access to online content, is a pressing concern for academia, publishers, and authors. Paraphrasing, a common tactic in textual plagiarism, compounds the problem further. The automatic detection of paraphrased plagiarism in text documents is a fundamental task in Natural Language Processing (NLP), crucial for maintaining academic integrity and authenticity. This paper presents an extensive investigation into Urdu sentential paraphrased plagiarism detection leveraging advanced Deep Neural Networks (DNNs) and Large Language Models (LLMs). The study builds upon the foundational work and proposes modifications to the Deep Text Reuse and Paraphrased Plagiarism Detection (D-TRaPPD) architecture to incorporate state-of-the-art pre-trained LLMs. The proposed approach, SELLM-D-TRaPPD, integrates various language models, including contextualized sentence embedding-based LLMs, language-agnostic and multilingual transformer-based LLMs, and multilingual knowledge-distilled transformer-based LLMs. We evaluated these models against three benchmark Urdu sentential paraphrase corpora—Urdu Sentential Paraphrase Corpus, Urdu Short Text Reuse Corpus, and Semi-automatic Urdu Sentential Paraphrase Corpus. The results demonstrate the effectiveness of SELLM-D-TRaPPD with LLMs, achieving F1 scores of 92.09%, 96.70%, and 98.23%, respectively. A comparative analysis with existing state-of-the-art methods shows significant performance improvements, establishing SELLM-D-TRaPPD as the new leading approach for Urdu sentential paraphrased plagiarism detection. These findings highlight the value of leveraging advanced neural network architectures and pre-trained LLMs in improving the accuracy and effectiveness of paraphrased plagiarism detection in Urdu, addressing a crucial gap in Urdu NLP research.

CCS Concepts: • **Computing methodologies** → **Neural networks**; • **Applied computing** → *Document analysis.*

Additional Key Words and Phrases: Automatic paraphrase detection, Sentential Paraphrased Plagiarism Detection, Large Language Models, Urdu NLP, DNNs, LLMs, Natural Language Processing

ACM Reference Format:
Hafiz Rizwan Iqbal, Muhammad Sharjeel, Jawad Shafi, Usama Mehmood, and Agha Ali Raza. 2024. Urdu Sentential Paraphrased Plagiarism Detection Using Large Language Models. 1, 1 (December 2024), 20 pages. https://doi.org/XXXXXXX.XXXXXXX

Authors' Contact Information: Hafiz Rizwan Iqbal, rizwan.iqbal@itu.edu.pk, Information Technology University (ITU), Arfa Software Technology Park, Lahore, Punjab, Pakistan; Muhammad Sharjeel, COMSATS University Islamabad (CUI), Lahore Campus, Lahore, Pakistan, muhammadsharjeel@cuilahore.edu.pk; Jawad Shafi, COMSATS University Islamabad (CUI), Lahore Campus, Lahore, Pakistan, jawadshafi@cuilahore.edu.pk; Usama Mehmood, Information Technology University (ITU), Arfa Software Technology Park, Lahore, Punjab, Pakistan, usamamehmood@itu.edu.pk; Agha Ali Raza, Lahore University of Management Sciences (LUMS), Lahore, Punjab, Pakistan, agha.ali.raza@lums.edu.pk.

Manuscript submitted to ACM

1 Introduction

Paraphrasing is the re-statement of a text with different expressions while maintaining the same meaning. Automatic paraphrase detection is a core NLP task that aims to determine whether two texts (sentences) are in a paraphrase relationship [8]. If the two sentences convey the same semantic meaning, they are categorized as "paraphrased"; otherwise, they are categorized as "non-paraphrased." Automatic paraphrase detection has been widely explored and has multiple applications, including duplicate question matching on social media [43], natural language generation [70], question-answering [27], and plagiarism detection [62].

In recent years, text plagiarism—the unacknowledged reuse of text—has become increasingly prevalent due to the abundance of freely accessible online resources. Plagiarism is present in published academic articles, raising deep concerns within the academic community [30]. A survey reveals that up to 90% of students have engaged in plagiarism [10], with 50% doing so in their assignment submissions [46]. In Germany, over 200 cases of academic plagiarism were identified through a crowd-sourcing initiative [30]. A recent report on cyber plagiarism[1] found that 66% of student work from 31 top-ranked U.S. universities was plagiarized. Additionally, AI-generated content is becoming a common practice in academia [15]. The consequences of undetected plagiarism are severe, including inflated publication records, compromised academic assessments, and unjust career advancements [30].

Paraphrasing is a common method of text plagiarism, where existing text is rephrased while retaining its original meaning [8]. Paraphrasing involves a range of text-altering operations, such as word reordering, synonym replacement, changing voice (active/passive), and summarization [16]. From an NLP perspective, several paraphrase typologies have been proposed [7, 50] to capture the various mechanisms of paraphrasing.

Detecting paraphrased cases of plagiarism is challenging, largely due to the abundance of digital text and the availability of AI-powered content creation tools (e.g., ChatGPT, YouChat, Chatsonic) [12]. While much research on paraphrase detection focuses on English [1, 9, 17, 22, 24] and European languages [20, 49], there is limited work on Southeast Asian languages, particularly Urdu. Urdu is an Indo-Aryan language [59], spoken by 231 million people in the Indian subcontinent. Urdu features a free word order and is influenced by Turkish, Arabic, and Persian languages [61]. Despite its wide use, Urdu lacks resources for core NLP tasks [60].

Recent efforts in the Urdu NLP community have led to the development of resources for automatic paraphrase detection. However, these studies mostly employ surface-level approaches and have not thoroughly explored DNNs and LLMs [36, 58, 62, 63].

Iqbal et al. [41] pioneered the use of DNNs for Urdu paraphrased plagiarism detection by introducing a hybrid architecture, D-TRaPPD, utilizing traditional word embeddings like FastText [35]. Although their work showed significant improvements, they did not explore contextual word embeddings (CON-WE) like Embeddings from Language Models (ELMo) [53] or transformer-based LLMs (e.g., BERT [21], RoBERTA [44], XLM [57]).

In this paper, we address these gaps by extending Iqbal et al. [41] work with modifications to the D-TRaPPD architecture, incorporating state-of-the-art pre-trained LLMs. Our major contribution is the enhancement of the D-TRaPPD architecture for sentence-level Urdu paraphrased plagiarism detection. This includes using non-contextualized word embedding-based language models (NC-WELM) such as Urdu-Word2Vec [54] and FastText [35], CWE like ELMo [53], contextualized sentence embedding-based LLMs (CON-SELLMs) including language-agnostic and multilingual transformer-based LLMs (LAM-TLLMs) like Language-agnostic BERT Sentence Embedding (LaBSE) [29] and XLM-R

[1] https://www.checkforplagiarism.net/cyber-plagiarism, Last visited: 20-Feb-2024

[57], and multilingual knowledge-distilled transformer-based LLMs (MKD-TLLMs) such as mUSE, XLM-R, and Distil-mBERT [57]. We conducted a comprehensive evaluation of these models on three benchmark Urdu paraphrase corpora: the Urdu Sentential Paraphrase Corpus [36], the Urdu Short Text Reuse Corpus [58], and the Semi-automatic Urdu Sentential Paraphrase Corpus [41]. Furthermore, we performed a detailed comparison of all proposed approaches (WELM-D-TRaPPD-W2V, WELM-D-TRaPPD-FastText, SELLM-D-TRaPPD-ELMO, SELLM-D-TRaPPD-BERT, SELLM-D-TRaPPD-LaBSE, SELLM-D-TRaPPD-XLM-R, SELLM-D-TRaPPD-Distil-mBERT, and SELLM-D-TRaPPD-USE) with the state-of-the-art.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the approaches used for Urdu paraphrase detection. Section 4 explains the experimental setup and evaluation measures. Section 5 presents results and analysis. Section 6 concludes the study. Finally, Section 7 outlines the limitations of this study and discusses future work.

2 Literature Review

In the scope of paraphrased plagiarism detection, numerous approaches have been proposed in the past [1, 3, 8, 22, 24, 31]. These approaches typically fall into three main categories: (i) surface-level, (ii) semantics-level, and (iii) idea-level (beyond the scope of this research). While surface-level approaches have gained significant attention for Urdu paraphrased plagiarism detection [14, 36, 39, 58, 62, 63], the exploration of semantics-level approaches remains limited [37, 41]. Notably, the DNN-based methods and LLMs remain largely unexplored in existing literature, constituting the primary focus of this research endeavor. To the best of our knowledge, no one has explored this way of research (i.e., leverage hybrid DNNs in conjunction with pre-trained LLMs) in Urdu NLP for sentence-level paraphrased plagiarism detection.

The evolution of machine learning to deep learning and neural networks has significantly supported semantics-based methods for paraphrased plagiarism detection. Esteki et al. [26] introduced the Index Word Replacement (IWR) approach, leveraging Support Vector Machine (SVM), DICE similarity, Longest Common Subsequence (LCS), and Levenshtein distance to identify paraphrased plagiarism in Persian texts. Their experiments on PAN-PC Persian corpora reported a PlagDet score of 0.80. Similarly, Hussain et al. [40] proposed a semantic similarity-based plagiarism detection method employing Nearest Neighbour (NN) and SVM, demonstrating competitive performance across four popular corpora.

In terms of word/phrase/sentence embedding approaches, Wieting et al. [67] proposed learning paraphrastic sentence embeddings by averaging word embeddings from a paraphrased pair database [32], showcasing superior performance. Arora et al. [5] trained unsupervised word embeddings on Wikipedia's articles, yielding significant improvements. Another notable approach by Wieting et al. [68] introduced the Gated Recurrent Averaging Network (GRAN), utilizing sentence pairs for sequence representation. However, Wieting et al.'s initial approach [67] outperformed these methods.

DNN-based semantic textual similarity approaches fall into three main categories: CNN-based, LSTM-based, and hybrid approaches. Hybrid models, which integrate CNNs and LSTMs, have shown better performance compared to standalone approaches for semantics-based plagiarism detection. Hybrid architectures combine the strengths of CNNs and LSTMs to capture both local and global features of text, compensating for each other's limitations. This trend is evident in the literature, where hybrid approaches consistently outperform standalone models. For instance, Agarwal et al. [1] introduced DeepParaphrase, a CNN-LSTM-based architecture for paraphrase detection. By combining local CNN features with LSTM's ability to capture long-term dependencies, DeepParaphrase achieved state-of-the-art performance on SemEval 2015 Twitter and MRPC corpora, with F1 scores of 0.75 and 0.84, respectively. Hambi et.al [23] introduced a plagiarism detection framework for English texts, incorporating Doc2Vec, Siamese-LSTM, and CNN models to build

preprocessing, learning, and detection layers, respectively. Comparative analysis with existing tools highlights the proposed approach's effectiveness in detecting various plagiarism types, achieving a notable accuracy of 98.33%.

Transformers, the latest advancements in NLP, have transformed the landscape of paraphrase detection by integrating attention mechanisms and context into word embeddings. These models utilize attention to identify crucial parts of input sequences at each step. Notable examples include Generative Pre-Trained Transformers (GPT, GPT-2, GPT-3) [55] and BERT [21], both of which are well-known pre-trained language models based on transformers. OpenAI's GPT trains language models unsupervised on extensive textual data, similar to ELMo's approach. However, GPT and ELMo differ in architecture and how they use contextualized embeddings. ELMo utilizes two separate LSTMs (left-to-right and right-to-left), combining their representations with shallow concatenation, while GPT predicts future tokens solely in one direction using multi-layer transformers [65]. GPT has undergone extensive evaluation across various NLP tasks, including semantic similarity and paraphrase detection [15].

The BERT, a cutting-edge LLM, undergoes training on vast volumes of raw text and fine-tune for specific tasks. Unlike GPT, BERT's architecture employs a bidirectional Transformer encoder, enabling it to analyze both left-to-right and right-to-left contexts during training. BERT undergoes training through two tasks: (i) Masked Language Model (MLM), where 15% of tokens in a sequence are randomly masked for the model to predict the missing words, and (ii) Next Sentence Prediction (NSP), a binary classification task determining if a sentence follows another. BERT has undergone evaluation across various NLP tasks, including paraphrase detection, showcasing its effectiveness in capturing semantic and syntactic characteristics by considering the context surrounding a word. The BERT results for paraphrase detection [4] have outperformed and have shown a better representation of context around the word rather than just after the word in the context of capturing syntactic and semantic properties of the word [21].

To overcome the constraints of monolingual sentence embedding models, there is a rising demand for multilingual or language-agnostic sentence embedding models. Language Agnostic Sentence Representations (LASER) [6] is at the forefront, providing versatile multilingual vector representations for sentences spanning 93 languages. Another contender, mUSE [69], is a pre-trained multilingual and multitask sentence embedding model, leveraging translation-based bridge tasks and two prominent transformers [65], along with CNN-based multilingual models. Multilingual BERT (mBERT), adopting the BERT architecture and training methodology [21], is trained on concatenated raw text from Wikipedia's 104 languages, without explicit cross-lingual signals alignment. Language-agnostic BERT (LaBERT) [29], a derivative of mBERT [21], generates a shared embedding space for 109 languages, using a unique training approach merging two pre-trained encoders: MLM and Translation Language Model (TLM). Cross-lingual Language Model (XLM-R), with 'R' representing Robustly Optimized BERT pre-training Approach (RoBERTa) [18], is a transformer-based multilingual large language model, trained on cleaned text from 100 languages. Multilingual models exhibit promise across various NLP tasks, particularly in areas like semantic similarity and paraphrase detection.

Multilingual Knowledge Distillation (MKD) [57] offers a straightforward and effective method to broaden existing pre-trained monolingual sentence embedding models to encompass new languages. MLD operates on the principle that a translated sentence occupies the same position as the source sentence in a shared embedding space. To implement MLD, a teacher model, already trained on monolingual English text (such as Sentence-BERT (SBERT) [56]), and a collection of translated (parallel) sentences in other languages are required. Subsequently, a new student model is trained in a multilingual setting to distill the knowledge from the teacher model. The student model learns multilingual sentence embeddings by aligning identical sentences (vector spaces) across different languages, thereby transferring the vector characteristics of the source language to diverse target languages. MLD has been evaluated across various text classification tasks, including multilingual paraphrase detection and semantic textual similarity (using the multilingual

STS17 corpus [11]), bitext mining (employing BUCC mining corpora [72]), and Tatoeba similarity search (over the Tatoeba test set [6]). Its efficacy has been demonstrated for over 50 languages from different language families.

While our primary focus is on Urdu, it's important to consider recent advancements in paraphrased plagiarism detection in other languages, as these approaches may offer valuable insights for multilingual applications or adaptation to Urdu.

In the Arabic domain, Al et al. [2] applied ELMo to paraphrase detection in Modern Standard Arabic (MSA) text, trained on a corpus containing MSA and 24 other Arabic dialects, surpassing traditional static word embedding methods. Additionally, Vrbanec et al. [66] compared the performance of various context vector-based word representation models, demonstrating the superiority of deep learning-based methods for semantic sentence similarity and paraphrase detection tasks. Moreover, Mahmoud et al. [45] investigated the efficacy of BERT-based models for paraphrase detection. Their proposed framework leverages AraBERT to extract contextualized embeddings, followed by a Siamese LSTM architecture for modeling sentence pairs. Evaluated on OSAC and SemEval corpora, as well as a custom-built paraphrased corpus, their approach achieved impressive F1-scores of 87.31% and 83.97% on OSAC and SemEval, respectively. This work demonstrates the potential of combining pre-trained language models with specialized architectures for paraphrase detection. Further advancing Arabic NLP, Hamza et al. [38] pioneered the use of ELMo embeddings to capture semantic and syntactic word relationships for question classification. Their exploration of various deep learning architectures, including CNNs, RNNs, and hybrid models, provided valuable insights into the task. However, their reliance on concatenation for feature fusion and the use of a softmax classifier might limit the model's ability to capture complex interactions between features.

In the Persian language, which shares some linguistic features with Urdu, Emami et al. [25] addressed the challenges of semantic textual similarity. They developed a corpus of 35,266 sentence pairs from movie and TV show subtitles and proposed a neural network-based approach using word vector representations. Their method achieved an impressive F-measure of 98.87% with binary classification on their corpus and 75.98% with 4-class classification on the PAN2016 dataset. Building upon this work, Zareshahi et al. [71] utilized the ParsBERT language model, which is specifically trained on Persian data, to convert tokenized sentences into meaningful vectors. By applying pooling layers and cosine similarity, they achieved a Pearson correlation coefficient of approximately 0.82, surpassing previous models that used combinations of FastText embeddings and CNN architectures.

These advancements in Arabic and Persian paraphrase detection and semantic similarity assessment offer valuable insights for our work on Urdu. The success of language-specific BERT models (AraBERT, ParsBERT) suggests that developing or fine-tuning a BERT model for Urdu could potentially enhance performance. Additionally, the effectiveness of Siamese architectures and the use of cosine similarity on pooled embeddings present promising directions for future research in Urdu paraphrased plagiarism detection.

2.1 Urdu Text Reuse and Plagiarism Detection

Initially, Urdu text reuse and plagiarism detection involved applying surface-level approaches for measuring similarity between the original and reused texts. A pioneering effort was made by Sharjeel et al. [63] in developing the COUNTER corpus, which contains real examples of text reuse from the field of journalism. It includes 600 manually annotated document pairs categorized as Wholly Derived (WD, 135), Partially Derived (PD, 288), or Non-Derived (ND, 177). In the experiments performed, various feature extraction approaches were employed, including Word N-gram Overlap (WNO), Vector Space Model (VSM), Greedy String Tiling (GST), LCS, Stop Word-based N-gram and Sentence/Token ratio, using a Naïve Bayes (NB) classifier. Optimal performance was achieved with WNO (F1 = 81.00) for a binary classification task.

Sameen et al. [58] proposed the Urdu Short Text Reuse Corpus (USTRC), a gold standard corpus comprising 2,684 sentence pairs extracted from journalistic sources. The corpus is annotated with three levels of reuse: verbatim (496 pairs), paraphrased (1,329 pairs), and independently written (859 pairs). The authors applied a range of approaches, including VSM, WNO, Character N-gram Overlap (CNO), LCS, Local Alignment (LA), Global Alignment (GA), Type Token Ratio (TTR), and Token Ratio (TR), to evaluate the corpus. Their findings indicate optimal F1 scores of 77.60 for binary using WNO.

Hafeez et al. [36] introduced a large-scale, manually annotated Urdu Sentential Paraphrases Corpus (USPC). The corpus comprises 4,900 sentence pairs labeled as paraphrased (2941) or non-paraphrased (1959). They developed and compared multiple approaches using NCWE like Word2Vec, Sentence Transformers (ST), and a Feature Fusion (FF) approach. Cosine similarity scores were computed between the embedding vectors generated by individual STs. Their findings demonstrated the superiority of the FF approach in enhancing classification performance, achieving an F1-score of 85.56 on the USPC, showcasing its effectiveness at the sentence level.

A recent effort made by Mehak et al. [47] is the development of a benchmark Urdu Text Reuse Detection at Phrasal level (UTRD-Phr-23) corpus, manually annotated, containing 25,001 text pairs categorized as derived (15,105) or non-derived (9896). To evaluate the corpus, a comparative analysis of NCWE and ST-based approaches was performed, calculating cosine similarity scores between embedding vectors from each ST. Experimental results achieved F1 = 63.00 using a combination of embeddings extracted from eight different ST-based models and a Random Forest (RF) classifier.

Noreen et.al [52] proposed an Urdu text reuse corpus at the lexical level comprising 22,184 text pairs, manually annotated as derived (8,660) or non-derived (13,524). To investigate Urdu text reuse detection at the lexical level, NCWE, ST, and FF approaches were employed. To determine the text reuse, cosine similarity between the embedding vectors produced by each ST. The experiments performed demonstrated the dominance of the feature fusion approach, achieving an F_1-score of 70.00.

Recently, Iqbal et.al [41] developed a Semi-automatic Urdu Sentential Paraphrase Corpus (SUSPC) using a semi-automatic pipeline. It consists of 3,147 sentence pairs manually labeled as paraphrased (854) or non-paraphrased (2293). Moreover, the authors proposed two novel Deep Learning models, Word Embeddings-based N-gram Overlap (WENGO) and D-TRaPPD, for paraphrase detection, and text reuse and plagiarism detection. Word embedding vectors were extracted from FastText [35], a pre-trained NCWEM for Urdu. Experimental results showed D-TRaPPDs performed better than WENGO in both tasks, achieving F1-scores of 96.80 for paraphrase detection UPPC, 88.90 on USTRC, and 96.80 for text reuse and plagiarism detection on SUSPC, respectively.

Table 1. Comparison of Urdu Text Reuse and Plagiarism Detection Approaches

Reference	Approach(s)	Corpus/Classifier/F1	Sem	CNN-LSTM	CWE	NCWE	mSE	mLLMs
Sharjeel et al. [63]	WNO, VSM, GST, LCS	COUNTER/NB/81.00	✖	✖	✖	✖	✖	✖
Sameen et al. [58]	VSM, WNO, CNO, LCS, LA, GA, TTR, TR	USTRC/J48/77.60	✖	✖	✖	✖	✖	✖
Hafeez et al. [36]	FF, NCWE, ST	USP/RF/85.56	✔	✖	✖	✔	✖	✔
Mehak et al. [47]	NCWE, ST	UTRD-Phr-23/RF/63.25	✔	✖	✖	✔	✖	✔
Noreen et al. [52]	FF, NCWE, ST	UTRD-Lex-23/RF/70.60	✔	✖	✖	✔	✖	✔
Iqbal et al. [41]	DNN-based WENGO and D-TRaPPD	UPPC/DNN/84.74, USTRC/DNN/87.85, SUSPC/DNN/96.80	✔	✔	✖	✔	✖	✖
Our Study	DNN-based WELM-D-TRaPPD and SELLM-D-TRaPPD	USTRC/DNN/92.09, USPC/DNN/96.70, SUSPC/DNN/98.23	✔	✔	✔	✔	✔	✔

Legend: **Sem**: Semantic-level; A ✔indicates the inclusion of the feature, while an ✖indicates that the feature is not addressed in the paper.

To conclude, previous research on Urdu paraphrase and text reuse detection primarily centered on developing standard evaluation corpora (mostly using manual approaches) and employing simple surface-level and machine learning-based approaches. These approaches offer several advantages, including simplicity and effectiveness in identifying textual alterations. Approaches like LCS and GST are useful for detecting small modifications and block moves, while VSM and WNG have been applied successfully at the document level. Sequence alignment methods like GA and LA help in aligning sequences of different lengths, and style-based methods like TTR capture distinctive writing styles. These methods provide a solid foundation for detecting various forms of text reuse and plagiarism in a resource-poor language Urdu.

Despite their utility, these approaches have notable limitations. Many struggle to capture semantic nuances, long-term dependencies, and vocabulary gaps. LCS struggles with block moves, and methods like VSM, WNG, and GST perform poorly in handling unseen instances or semantic changes. Sequence alignment methods underperform when there are significant length differences, and style-based approaches fail in document and sentence-level detection tasks, particularly when text segments vary in length. These weaknesses suggest that while surface-level methods are effective for basic detection, they lack the sophistication required for deeper semantic analysis.

To capture semantic nuances, various research efforts for Urdu text reuse and plagiarism detection [36, 47, 52] have incorporated traditional word embeddings like Word2Vec and more recent LLM-based embeddings. Table 2 highlights the key differences between traditional embedding methods and more recent LLM-based approaches, showcasing the strengths and limitations of each. While these embeddings-based approaches detected text reuse and plagiarism at a semantic level, they did not achieve significant improvements. This is largely because the pre-trained embedding models were used as-is to extract embeddings, and textual similarity between text pairs was calculated using simple measures like cosine and Jaccard similarity. Additionally, no task-specific fine-tuning was performed for the Urdu text reuse and plagiarism detection tasks, resulting in embedding vectors that were unable to fully capture the deep patterns and nuances of text reuse in the datasets. Another key limitation was the lack of adaptation to the specific linguistic features of Urdu, such as orthographic variations and context-dependent meanings, which require fine-tuning for the embeddings to be truly effective.

Table 2. Comparison: Traditional Embeddings vs LLM-based Embeddings

Aspect	Traditional Embeddings	LLM-based Embeddings
Approach	Use statistical methods or shallow neural networks	Leverage deep learning and transformer architectures
Context Understanding	Limited context understanding	Rich contextual understanding
Dimensionality	Often lower dimensional (e.g., 100-300 dimensions)	Higher dimensional (e.g., 768-1024 dimensions)
Training Data	Usually domain-specific corpora	Trained on vast and diverse datasets
Semantic Capture	Captures basic semantic relationships	Captures complex semantic and syntactic relationships
Computational Resources	Generally less resource-intensive	More computationally expensive
Adaptability	Less adaptable to new domains	More adaptable to various domains and tasks
Interpretability	Often more interpretable	Less interpretable due to complexity
Examples	Word2Vec, GloVe, FastText	BERT, GPT, RoBERTa embeddings
Handling of OOV words	Limited capability with out-of-vocabulary words	Better handling of rare or unseen words
Multilingual Support	Often language-specific	Can support multiple languages in a single model
Fine-tuning	Limited fine-tuning capabilities	Easily fine-tuned for specific tasks or domains
Contextual Variations	Static representations for words	Dynamic representations based on context
Task Performance	Good for specific NLP tasks	State-of-the-art performance on a wide range of NLP tasks
Integration Complexity	Simpler to integrate into existing systems	May require more complex integration and preprocessing

To address these issues, Iqbal et al. [41] made a pioneering effort by incorporating hybrid DNN approaches that are adept at capturing semantics for detecting Urdu paraphrased text reuse and plagiarism. By integrating Convolutional

Neural Networks (CNN) with Long Short-Term Memory (LSTM) networks, their models leveraged the strengths of both architectures for improved performance. However, their work has several limitations as it did not explore contextual CWE models like ELMo [53], any multilingual Sentence Encoder (mSE) like USE [57], or the more recently proposed transformer-based multilingual Large Language Models (mLLMs), such as LaBSE [29] and XLM-R [19]. These models could provide richer contextual representations and better adaptability to the nuances of Urdu, potentially enhancing the detection capabilities of paraphrased text reuse and plagiarism.

In response to the limitations of traditional approaches (i.e., surface-level and machine learning-based), advancements in DNN-based approaches, and the emergence of language-specific and language-agnostic transformer-based mLLMs, this study extends Iqbal et al.'s [41] work by introducing two novel D-TRaPPD variants: WELM-D-TRaPPD and SELLM-D-TRaPPD (Section 3). Furthermore, we leverage the capabilities of 2 NCWE models (W2Vec, FastText), one CWE model (ELMo), one mSE (Multilingual Universal Sentence Encoder (mUSE)), and state-of-the-art 4 transformer-based mLLMs (LaBSE, XLM-R, mBERT/XLM-R, Distill-mBERT) to address the challenges of Urdu paraphrased text reuse and plagiarism detection (Section 4.2.1). To evaluate these models, comprehensive experiments were conducted using the publicly available sentential corpora including USTRC, USPC, and SUSPC (Section 4.1). Notably, this research is a pioneering effort to leverage hybrid DNNs in conjunction with CWE, NCWE, mSE, and pre-trained mLLMs for the detection of Urdu paraphrased text reuse and plagiarism detection. Table 1 summarizes the above discussion and contrasts our proposed work with similar research and state-of-the-art methods in Urdu text reuse and plagiarism detection.

3 Urdu Sentential Paraphrased Plagiarism Detection

This section describes the novel DNN approach named D-TRaPPD [41], tailored specifically for detecting paraphrases in the Urdu language. D-TRaPPD is a hybrid DNN approach that comprises four key modules: (i) Text pre-processing and embedding extraction (ii) CNNs, (iii) LSTMs, and (iv) Semantic Similarity Estimation.

Figures 1 illustrate the architecture of the D-TRaPPD approach designed for Urdu paraphrase detection. At the start, Urdu text input undergoes essential preprocessing, involving the removal of numbers, punctuation, excessive white spaces, newlines, and any non-standard Urdu characters. Following this, the text is converted into embedding vectors. Using both word and sentence embedding models, embedding vectors for each input sentence pair (paraphrase/non-paraphrase pair) are generated using two separate methods.

In the case of a word embedding-based language model (WELM), the input text is first word tokenized. The WELM then generates an embedding vector for each word, stacked together to form an embedding matrix. For example, let's consider a sentence s consisting of n words, denoted as $w_1, w_2, w_3, ..., w_n$, where each w represents an individual word. These words are fed into the WELM, which returns the respective embedding vectors $v_1, v_2, v_3, ..., v_n$. The individual word vectors are merged to construct an embedding matrix M with dimensions $d \times n$. Henceforth, we shall refer to this architecture as WELM-D-TRaPPD (WELM-based D-TRaPPD).

In contrast, when using a sentence embedding-based LLM (SELLM) such as BERT, individual word embeddings are retrieved from the last hidden layer for each token in the input. Sentence embedding LLMs generate fixed-size vector representations for entire sentences, capturing their overall semantic meaning and context. However, since the D-TRaPPD architecture requires word-level embeddings, extracting embeddings directly from SELLMs is impractical. Therefore, we obtain word embeddings by accessing the last encoding layer of the LLM, which allows us to generate a 2D array suitable for input into a CNN.

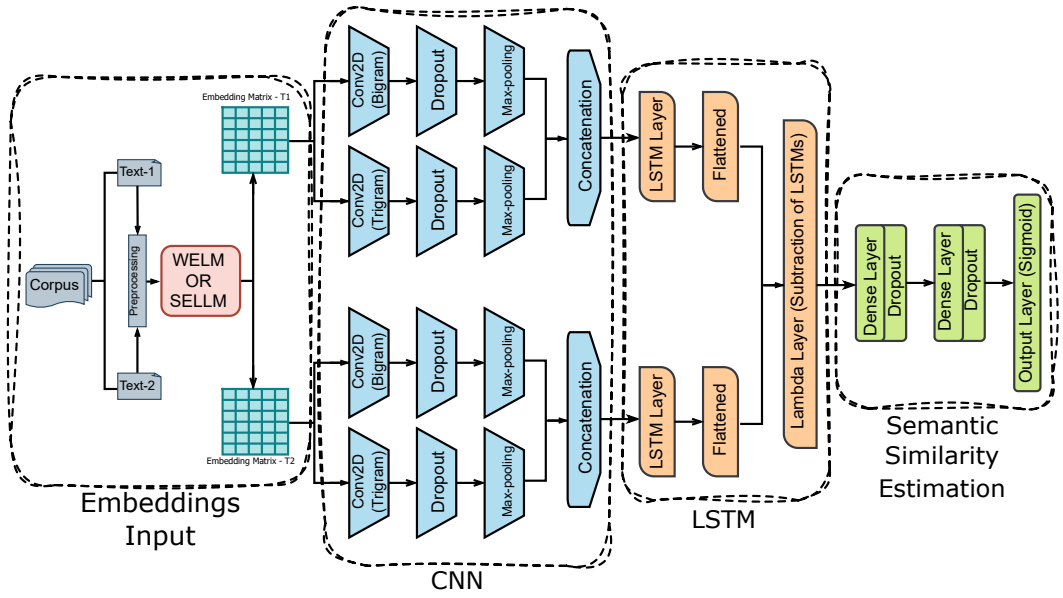

Fig. 1. Abstract level architecture of WELM-D-TRaPPD and SELLM-D-TRaPPD approaches for Urdu sentential paraphrased plagiarism detection

For example, let's consider a sentence s comprising n words. The sentence when goes through the SELLM, the word embeddings vectors $v_1, v_2, v_3, ..., v_n$ are fetched from its last hidden layer. Each of these word vectors is combined to form an embedding matrix M with dimensions $d \times n$. Henceforth, this architecture is referred to as SELLM-D-TRaPPD (SELLM-based D-TRaPPD).

In the next step, the WELM and SELLM-based matrices are used as input to a Siamese-adapted CNN module. Serving as a pivotal component, the CNN module is tasked with extracting meaningful structures from the input text [34]. By employing convolutional layers alongside pooling layers, CNNs are renowned for capturing robust local features of words or phrases, regardless of their positions [33]. The word-wise convolutions are conducted on the input embedding matrices using kernels of varying sizes, specifically 2 (referred to as bi-gram) and 3 (termed as tri-gram), with 128 filters, and a stride size of 1 for all spatial dimensions. Activation of the convolutional layers is achieved through Rectified Linear Unit (ReLU) [51]. To mitigate the risk of over-fitting, dropout layers [64] are thoughtfully incorporated, and max-pooling layers are employed to generate concise feature maps.

After creating compact feature maps via max-pooling, these maps are combined to prepare the input for the LSTM module. Within the LSTM module, the element-wise difference of the output vectors for both sentences is computed using a lambda layer. This resultant difference vector serves as the representative feature vector for the sentence pair, facilitating the learning of text similarity. To further regularize the proposed neural network, two fully connected layers are incorporated, each followed by dropout layers [42]. At the output layer, Sigmoid activation is utilized for binary classification, distinguishing between paraphrased or non-paraphrased sentences.

4 Experimental Set-up

This section outlines the corpora, approaches, evaluation methodology, hyper-parameter settings, and evaluation measures employed for detecting paraphrased plagiarism at the sentence level for the Urdu language.

4.1 Corpora

We evaluated our proposed approaches using three sentence-level Urdu corpora: the USTRC, the USPC, and the SUSPC.

- USTRC [58] is a benchmark Urdu corpus containing real cases of short text reuse from journalism. It consists of 2,684 short text pairs, manually extracted from 1200 news stories. These short texts are manually categorized into verbatim (496), paraphrased (1,329), and independently written (859). The verbatim (minor obfuscation) and paraphrased (major obfuscation) texts are the reworded version of the source text, whereas, the independently written cases are non-paraphrased (not derived from the source text).
- USPC [36] is a large-scale manually annotated sentence-level Urdu paraphrases corpus. It comprises 4,900 short texts (2,941 paraphrased and 1,959 non-paraphrased), meticulously sourced from Urdu newspapers.
- SUSPC [41] is a semi-automatically generated sentence-level paraphrased corpus for the Urdu language. It contains 3,147 sentence pairs marked as either paraphrased (854) or non-paraphrased (2,293).

4.2 Approaches

We implemented eight approaches based on the WELM-DTRaPPD and SELLM-D-TRaPPD architectures. These include: (i) WELM-D-TRaPPD-W2V, (ii) WELM-D-TRaPPD-FastText, (iii) SELLM-D-TRaPPD-ELMo, (iv) SELLM-D-TRaPPD-BERT, (v) SELLM-D-TRaPPD-LaBSE, (vi) SELLM-D-TRaPPD-XLM-R, (vii) SELLM-D-TRaPPD-Distil-mBERT, and (viii) SELLM-D-TRaPPD-USE.

4.2.1 Pre-trained Language Models. In our research, we utilized a diverse set of both non-contextualized and contextualized pre-trained word embedding models. Additionally, we incorporated sentence embedding models such as USE, along with transformer-based LLMs including BERT, RoBERTa, and others. These models contribute distinct features, crucial for enhancing the representation of input text in our Urdu sentential paraphrased plagiarism detection task.

The non-contextualized models consist of Urdu-Word2Vec [54], a mono-lingual Urdu Word2Vec with continuous bag-of-words [48] by Haider et al. [37], and FastText [35], character-based pre-trained word vectors for Urdu generated using the fastText API. FastText was further trained on additional Urdu text from various web pages and has output embeddings with a size of 300 dimensions. On the other hand, we utilized ELMo [13] as a contextualized embedding model. ELMO is a pre-trained contextualized mono-lingual word embedding model for Urdu by Fares et al. [28], trained on Wikipedia and Common Crawl data, with output embeddings of size 1024 dimensions.

We have also employed a variety of sentence embedding models, extracting individual word embeddings to enrich the input text representation. These models include: 1) mUSE [57] - a knowledge-distilled version of the multilingual Universal Sentence Encoder trained for 50+ languages providing word and sentence embedding vectors with dimensions 768 and 512, respectively, 2) LaBSE [29] - a BERT-based dual-encoder transformer trained on 6 billion translated text pairs for 109 languages delivering word and sentence embedding vectors of 768 dimensions;, 3) XLM-R [57], a multilingual transformer-based distilled version of XLM-RoBERTA trained on parallel data of 50 languages with output embeddings of 768 dimensions, 4) mBERT/XLM-R Mean [57] - a joint model pooling outputs from multilingual BERT and XLM-RoBERTA, generating 768-dimensional vectors for both words and sentences; and 5) Distil-mBERT

[57] - a knowledge-distilled multilingual version of Distil-BERT fine-tuned on parallel data in 50 languages, returning 768-dimensional embeddings vectors for words or sentences.

4.3 Fine-Tuning and Evaluation Strategy

The primary goal of the experiments performed is to find out whether a sentence pair is paraphrased or not. As we have labeled data, the task is treated as a supervised binary classification task. SUSPC and USPC corpora contain binary classification (paraphrased and non-paraphrased) whereas USTRC corpus incorporates three classes, i.e., verbatim, paraphrased, and independently written. For binary classification, verbatim and paraphrased classes are merged into a single class termed *paraphrased*, and independently written is designated as *non-paraphrased*. During encoding, paraphrased and non-paraphrased are represented by 0 and 1, respectively.

We implemented all approaches using Keras [2], a widely adopted deep learning API in Python. We experimented with various hyper-parameter combinations and selected those yielding superior performance. Consistent hyper-parameter settings across experiments on all the three corpora include: (i) stratified 10-fold cross-validation, (ii) *validation split* (0.1), (iii) *dropout rate* (0.5), (iv) *optimizer* (adam), (v) *kernel regularizer* (L_2), and (vi) *epochs* (50, 125, 250, 500, 1000, 2000). Smaller *batch sizes* (16, 32, 64, 128) were employed considering the number of instances in the corpora. The *loss* for binary classification is set as 'binary cross-entropy'. Table 3 outlines the hyperparameter settings across all three corpora, ensuring consistency in the experimental process.

Table 3. Summary of Hyper-parameters used in the experiments

Hyper-parameter	**Setting**
Cross-Validation	Stratified 10-fold cross-validation
Validation Split	0.1
Dropout Rate	0.5
Optimizer	Adam
Kernel Regularizer	L_2
Epochs	50, 125, 250, 500, 1000, 2000
Batch Sizes	16, 32, 64, 128
Loss Function	Binary Cross-Entropy (for binary classification)

We evaluated the approaches using standard evaluation measures commonly employed in prior Urdu paraphrased plagiarism detection studies [58, 63]. These include *Precision*, *Recall*, and F_1 score. In terms of classification, True Positives (TP) denote accurately classified relevant text pairs in the non-paraphrased class, while True Negatives (TN) are text pairs accurately classified as paraphrased.

5 Results and Discussion

In this section, we discuss the results of experiments performed for the Urdu paraphrased plagiarism detection task described in Section 4.

Tables 4, 5, and 6 demonstrate the detailed results for binary classification using the WELM-D-TRaPPD and SELLM-D-TRaPPD approaches 3. The variants of WELM-D-TRaPPD are WELM-D-TRaPPD-W2V and WELM-D-TRaPPD-FastText, and the variants of SELLM-DRaPPD are SELLM-D-TRaPPD-ELMO, SELLM-D-TRaPPD-BERT, SELLM-D-TRaPPD-LaBSE, SELLM-D-TRaPPD-XLM-R, SELLM-D-TRaPPD-Distil-mBERT, and SELLM-D-TRaPPD-USE. All of these variants are

[2]https://keras.io/ Last visited: 19-Nov-2024

compared across key metrics. It is important to note that the tables enlist only the highest F_1 scores [3] and their respective Precision and Recall, among the various combinations epochs, batch sizes, and the filter sizes. For each corpus, the best results along with their embedding model and embedding unit used, are shown in bold in the tables.

5.1 Results on USTRC

Table 4 shows the outcomes of approaches on the USTRC corpus, focusing on binary classification. Precision ranges from 67.07 to 96.39, with SELLM-D-TRaPPD-USE achieving the highest. Recall varies between 76.52 and 88.29, with SELLM-D-TRaPPD-USE leading again. The F_1 score spans from 71.32 to 92.09, with SELLM-D-TRaPPD-USE excelling.

In terms of performance ranking, SELLM-D-TRaPPD-USE emerges as the top performer across all metrics, while SELLM-D-TRaPPD-XLM-R demonstrates comparatively lower performance. SELLM-D-TRaPPD-USE's success is attributed to its utilization of the Universal Sentence Encoder, providing a comprehensive understanding of semantic nuances. On the other hand, SELLM-D-TRaPPD-XLM-R may face challenges in capturing intricate semantics or require further optimization due to its relatively lower performance. These insights offer a nuanced evaluation of each variant's strengths and weaknesses, guiding potential refinements for enhanced Urdu paraphrased plagiarism detection tasks.

Table 4. USPPD: Results on USTRC

Approach	Precision	Recall	F1
WELM-D-TRaPPD-W2V	94.19	83.96	87.88
WELM-D-TRaPPD-FastText	92.53	83.64	87.86
SELLM-D-TRaPPD-ELMO	93.89	87.20	90.42
SELLM-D-TRaPPD-mBERT/XLM-R	96.36	87.43	91.68
SELLM-D-TRaPPD-LaBSE	**96.39**	87.45	91.70
SELLM-D-TRaPPD-XLM-R	95.95	87.83	91.71
SELLM-D-TRaPPD-Distil-mBERT	67.07	76.52	71.32
SELLM-D-TRaPPD-USE	96.23	**88.29**	**92.09**

* **Bold** numbers in the tables represents the best-performing scores.

5.2 Results on USPC

Table 5 shows the results of approaches on the USPC corpus. Precision values range from 84.60 to 97.22, with SELLM-D-TRaPPD-USE achieving the highest precision. Similarly, recall rates vary between 71.72 and 96.19, again with SELLM-D-TRaPPD-USE. Moreover, the F_1 scores span from 77.37 to 96.70, with SELLM-D-TRaPPD-USE demonstrating the highest score.

When examining performance rankings, similar to the USTRC results, SELLM-D-TRaPPD-USE emerges as the top performer across all metrics, showcasing its effectiveness in detecting paraphrased plagiarism for Urdu. Conversely, SELLM-D-TRaPPD-Distil-mBERT exhibits relatively lower performance compared to other approaches, especially in terms of precision and recall. The exceptional performance of SELLM-D-TRaPPD-USE can be attributed to its utilization of the Universal Sentence Encoder, which enhances its ability to capture semantic nuances and similarities effectively.

An intriguing finding from the analysis is the impressive performance of WELM-D-TRaPPD-FastText, achieving a precision of 96.16. This indicates that the non-contextual word embedding model based on FastText outperforms

[3] All scores reported in the manuscript are expressed in percentages.

many state-of-the-art LLMs, falling short only in comparison to SELLM-D-TRaPPD-USE. The competitive performance of WELM-D-TRaPPD-FastText suggests that for languages like Urdu, which may have limited linguistic resources, leveraging pre-trained models like FastText, specifically trained on Urdu text, can yield highly effective results in detecting paraphrased plagiarism. Conversely, the relatively lower performance of other modern LLMs, which are either language-agnostic or trained on multilingual parallel corpora, highlights the importance of considering the linguistic characteristics and resource availability of the target language when selecting appropriate models for paraphrased plagiarism detection tasks. These findings underscore the potential efficacy of utilizing pre-trained language models specifically tailored to low-resource languages like Urdu for the accurate detection of real-world instances of paraphrased plagiarism.

Table 5. USPPD: Results on USPC

Approach	Precision	Recall	F1
WELM-D-TRaPPD-W2V	92.38	93.91	94.14
WELM-D-TRaPPD-FastText	96.18	96.13	96.16
SELLM-D-TRaPPD-ELMO	95.13	95.02	95.08
SELLM-D-TRaPPD-mBERT/XLM-R	95.41	95.43	95.42
SELLM-D-TRaPPD-LaBSE	96.25	95.56	95.90
SELLM-D-TRaPPD-XLM-R	95.91	95.54	95.73
SELLM-D-TRaPPD-Distil-mBERT	84.60	71.72	77.37
SELLM-D-TRaPPD-USE	**97.22**	**96.19**	**96.70**

5.3 USPPD: Results on SUSPC

Table 6 show the results of various approaches on the SUSPC corpus. Similar to the results of USTRC and USPS, SELLM-D-TRaPPD-USE outperformed all the other approaches used in the experiments. The reported precision ranged from 90.45 to 98.60, and recall varied between 89.31 and 98.23. It depicts the models' ability to accurately classify paraphrased and non-paraphrased instances. The F_1 score, spanning from 89.74 to 98.23, provides a comprehensive measure of overall performance.

Table 6. USPPD: Results on SUSPC

Approach	Precision	Recall	F1
WELM-D-TRaPPD-W2V	96.88	97.50	97.19
WELM-D-TRaPPD-FastText	96.92	96.69	96.80
SELLM-D-TRaPPD-ELMO	98.19	97.17	97.68
SELLM-D-TRaPPD-mBERT/XLM-R	98.59	**97.73**	98.16
SELLM-D-TRaPPD-LaBSE	98.23	97.58	97.91
SELLM-D-TRaPPD-XLM-R	98.60	97.54	98.07
SELLM-D-TRaPPD-Distil-mBERT	90.45	89.31	89.74
SELLM-D-TRaPPD-USE	**98.75**	97.72	**98.23**

5.4 Best Results Comparison

Table 7 indicate notable differences in performance metrics across the three corpora: USTRC, USPC, and SUSPC. The findings demonstrate that SUSPC yields the highest performance (F_1 = 98.23), while USTRC exhibits comparatively

lower scores (F_1 = 92.09) across all three corpora. Furthermore, it is noteworthy that the discrepancy between the best and worst results is nearly equivalent in all three corpora. Remarkably, the SELLM-D-TRaPPD-USE approach consistently outperforms other approaches across all corpora, suggesting its robustness in detecting Urdu paraphrased plagiarism.

USPPD approaches achieve intermediary results on the USPC compared to the USTRC and SUSPC. This could be attributed to the larger size of the USPC corpus, as DNN-based approaches often perform better with larger datasets. Additionally, the USPC dataset comprises curated paraphrased and non-paraphrased sentence pairs, while the USTRC merges verbatim and paraphrased pairs into a single "paraphrased" class, potentially leading to misclassifications.

In comparison with SUSPC, the competitive performance of USPPD on USPC suggests differences in the types of real and semi-automatically generated paraphrased pairs between the two corpora. The higher class imbalance in SUSPC could limit the extent of performance improvements compared to USPC, even if the corpus size were equalized.

Table 7. Comparison of Top Results in USPPD across USTRC, USPC, and SUSPC

Corpus	Approach	Precision	Recall	F1
USTRC	SELLM-D-TRaPPD-USE	96.23	88.29	92.09
USPC	SELLM-D-TRaPPD-USE	97.22	96.19	96.70
SUSPC	SELLM-D-TRaPPD-USE	**98.75**	**97.72**	**98.23**

5.5 USPPD: Results Comparison with the State-of-the-art for USTRC, USPC, and SUSPC

In Table 8, we present a detailed comparison of our proposed SELLM-D-TRaPPD-USE model with the existing state-of-the-art approaches, including D-TRaPPD and Feature Fusion, across three benchmark corpora: USTRC, USPC, and SUSPC. SELLM-D-TRaPPD-USE demonstrates significant improvements in precision, recall, and F1 scores. For instance, it achieves an F1 score of 92.09 on the USTRC corpus, compared to 87.85 by D-TRaPPD, highlighting a notable performance enhancement. Similarly, on the SUSPC corpus, SELLM-D-TRaPPD-USE surpasses D-TRaPPD with an F1 score of 98.23, showcasing its superior ability to capture nuanced paraphrasing patterns.

The USPC corpus results further underscore the effectiveness of our approach. SELLM-D-TRaPPD-USE achieves an F1 score of 96.70, outperforming the Feature Fusion method (85.56). While precision and recall values for Feature Fusion are unavailable, the substantial gap in F1 scores indicates the clear advantage of leveraging complex hybrid architectures over simpler, traditional methods. These findings emphasize the critical role of advanced contextual embeddings and hybrid DNN architectures in improving performance for paraphrased plagiarism detection tasks.

5.5.1 Critical Analysis of Findings. The performance improvements demonstrated by SELLM-D-TRaPPD-USE are primarily attributed to its ability to effectively combine the strengths of CNN-LSTM architectures with transformer-based LLMs. These architectures, when combined with contextual embeddings from LLMs, allow SELLM-D-TRaPPD-USE to excel in capturing both shallow and deep semantic relationships within paraphrased text.

However, the results reveal that the highest performance is achieved on the SUSPC corpus (F1 = 98.23), while lower scores are observed for USTRC (F1 = 92.09) and USPC (F1 = 96.70). This discrepancy can be attributed to the nature of the corpora. The paraphrasing cases in SUSPC are semi-automatically generated, whereas USTRC and USPC contain real-world paraphrased cases. This suggests that the proposed model struggles more with detecting real paraphrased plagiarism, which tends to involve complex and nuanced linguistic transformations. Real-world cases often exhibit greater variability in sentence structure, semantic shifts, and creative rephrasing, making them inherently

more challenging for automated detection systems. Addressing this limitation would require further fine-tuning of the model and exploring linguistic rules specific to real-world paraphrasing scenarios.

Another limitation lies in the reliance on pre-trained transformer-based LLMs, such as USE and LaBSE, which require significant computational resources. While the high precision (98.75) and recall (97.72) on the SUSPC corpus underscore the model's capability, the results also point to potential challenges in generalizing to low-resource environments or domains with different paraphrasing characteristics.

5.5.2 Broader Implications. The implications of these findings are significant for both academic and practical applications. By addressing the limitations of surface-level and traditional embedding methods, SELLM-D-TRaPPD-USE provides a robust solution for paraphrased plagiarism detection in Urdu. However, the struggle with real-world cases highlights the need for more nuanced evaluation corpora and further model optimization. Future work could focus on annotating real-world corpora with detailed paraphrasing types and incorporating error analysis to identify and address specific weaknesses. Additionally, efforts to develop Urdu-specific pre-trained LLMs and optimize the architecture for computational efficiency would enhance the applicability of the proposed model.

Table 8. Comparison of State-of-the-Art with SELLM-D-TRaPPD Approaches

Corpus	Approach	Precision	Recall	F1
Existing state-of-the-art				
USTRC	D-TRAPPD	92.52	83.64	87.85
USPC	Feature Fusion	–	–	85.56
SUSPC	D-TRaPPD	96.91	96.68	96.80
New state-of-the-art				
USTRC	SELLM-D-TRaPPD-USE	96.23	88.29	92.09
USPC	SELLM-D-TRaPPD-USE	97.22	96.19	96.70
SUSPC	SELLM-D-TRaPPD-USE	**98.75**	**97.72**	**98.23**

6 Conclusion

Our study delved into the prevalent issue of text plagiarism, underscored by the rampant reuse of digital content and the emergence of easy-to-use AI-generated tools. Paraphrasing, a key tactic in text plagiarism, exacerbates these concerns, posing challenges for academia, publishers, and authors alike. Through an exploration of automatic paraphrase detection, we addressed the critical need to distinguish between paraphrased and non-paraphrased text, a fundamental task in NLP. We note that prior research on plagiarism detection has primarily focused on English and European languages.

This research fills a crucial gap by investigating the detection of sentential paraphrased plagiarism in Urdu, an Indo-Aryan language spoken by millions. Building upon the foundational work of Iqbal et al. [41], our study proposes enhancements to the D-TRaPPD architecture, leveraging advanced DNNs and LLMs to detect sentence-level paraphrased plagiarism detection for Urdu. Our evaluation of various LLMs on Urdu paraphrase corpora demonstrates a substantial improvement in detecting sentential paraphrased plagiarism in Urdu compared to previous benchmarks.

The results collectively establish SELLM-D-TRaPPD-USE as the new state-of-the-art approach, demonstrating its superior performance in detecting sentential paraphrased plagiarism in Urdu compared to existing benchmarks. Moreover, the significant performance enhancement shown by SELLM-D-TRaPPD-USE over the current state-of-the-art (D-TRaPPD) emphasizes the effectiveness of advanced neural network architectures and semantic understanding facilitated by LLMs in enhancing the accuracy of Urdu paraphrased text detection tasks.

7 Limitations and Future work

Despite the promising results achieved in this study, our research is not without limitations. In this section, we identify the general and Urdu-specific constraints that may affect the applicability of our findings. Additionally, we propose potential directions for future research to overcome these limitations and enhance the effectiveness of our approach.

7.1 General Limitations

Common challenges with hybrid DNNs and pre-trained LLMs include:

- Dependence on Pre-Trained Models: The effectiveness of our model relies heavily on pre-trained LLMs, which may not be fully optimized for the nuances of the Urdu language.
- High Computational Requirements: The hybrid CNN-LSTM architecture with LLMs demands significant computational resources, which may not be feasible in all real-world scenarios, especially in resource-limited environments.
- Handling Real-World Data Variability: Our experiments have been conducted on two gold-standard corpora, but real-world data often presents more noise, variation, and domain-specific challenges that could affect model performance.

7.2 Urdu-Specific Limitations

Our research faces certain challenges unique to the Urdu language. These include:

- Lack of Detailed Error Analysis: A key limitation of our current work lies in the inability to conduct a detailed error analysis based on specific paraphrasing types. The corpora employed in our experiments (USTRC, USPC, and SUSPC) classify sentence pairs only as "paraphrased" or "non-paraphrased" without providing further annotations or labels regarding the types of paraphrasing involved, such as inflectional, derivational, or discourse-based variations. This lack of paraphrasing-type information limits our ability to examine the model's performance on specific linguistic transformations. Conducting such an analysis would require linguistic expertise to manually classify each text pair into specific paraphrasing categories, which is both labor-intensive and time-consuming, especially for large-scale corpora.
- Limited Availability of High-Quality Urdu Datasets: The proposed approach relies on high-quality annotated datasets, and while we used three gold-standard corpora, Urdu still lacks the diverse, large-scale datasets needed to fully train and fine-tune models for paraphrase detection. This limitation affects the generalizability of the model to unseen real-world data.
- Inconsistent Handling of Complex Morphology: Urdu is a morphologically rich language, with complex grammatical rules and structures. LLMs and other models pre-trained on multilingual datasets may not be fully optimized for Urdu's specific morphological nuances, which could lead to incorrect paraphrase detection.
- Lack of Pre-trained LLMs for Urdu: The pre-trained LLMs used in our approach, such as BERT or RoBERTa, are trained in a mix of languages, and Urdu may not have received sufficient representation during training. This poses a limitation in how well the models can capture Urdu-specific linguistic properties, leading to potential performance degradation.

7.3 Future Work

Below are some specific future work directions aimed at overcoming these challenges and enhancing the overall performance of the models.

- Detailed Error Analysis for Paraphrasing Types: To address the lack of detailed error analysis, manual annotation of the corpora with paraphrasing type information (e.g., inflectional, derivational, discourse-based) is required. Tools like Prodigy[4] or BRAT[5] could facilitate this annotation process. Collaborating with linguists to conduct this annotation and develop automated paraphrase classification techniques for Urdu would enhance error analysis and improve model adaptability.
- Expanding Availability of High-Quality Urdu Datasets: The scarcity of large-scale, diverse datasets for Urdu remains a significant challenge. Future efforts could focus on creating more comprehensive datasets by leveraging crowd-sourcing or collaborating with academic institutions. Techniques like self-training, contrastive learning, or back-translation could be explored to augment existing datasets. Additionally, cross-lingual transfer learning could leverage resources from similar languages like Hindi and Punjabi to enrich Urdu datasets.
- Improving Morphological Handling: To better handle Urdu's complex morphology, future work could explore the integration of morphological analyzers specific to Urdu. These tools can be used to pre-process the text to account for rich inflectional and derivational variations. Additionally, incorporating linguistic rules into the deep learning models and experimenting with morphologically-aware embeddings could enhance performance.
- Developing Urdu-Specific Pre-trained LLMs: The reliance on pre-trained multilingual LLMs presents a limitation due to insufficient representation of Urdu in these models. Future work could focus on developing Urdu-specific pre-trained LLMs by training on large-scale Urdu corpora. Transfer learning techniques could adapt existing multilingual models to better understand Urdu linguistic features. Additionally, multimodal models incorporating text and image data could address complex cases of plagiarism involving multimedia elements.
- Improving Explainability and Real-World Deployment: Enhancing the interpretability of the models through explainable AI (XAI) techniques would provide users with insights into the decision-making process of the models. Deploying these models in real-world applications, such as plagiarism detection platforms, could involve addressing challenges related to scalability, user interfaces, and robustness to noisy input data.

Acknowledgments

I extend my heartfelt gratitude to my esteemed supervisor (the late *Dr. Saeed-Ul-Hassan*), whose unwavering guidance and support were indispensable to the completion of this research. His wisdom and encouragement will be forever cherished.

Further, we gratefully acknowledge the financial support of the Higher Education Commission (HEC) of Pakistan through the National Research Program for Universities (NRPU) grant (Project ID: 9854/Punjab/NRPU/RD/HEC/2017). We also thank the Information Technology University, Lahore, Pakistan for their support. The authors declare no conflict of interest.

References

[1] Basant Agarwal, Heri Ramampiaro, Helge Langseth, and Massimiliano Ruocco. 2018. A deep network model for paraphrase detection in short text messages. *Information Processing & Management* 54, 6 (2018), 922–937.

[4] https://prodi.gy Last visited: 22-Nov-2024
[5] https://brat.nlplab.org Last visited: 22-Nov-2024

[2] Hesham Al-Bataineh, Wael Farhan, Ahmad Mustafa, Haitham Seelawi, and Hussein T Al-Natsheh. 2019. Deep Contextualized Pairwise Semantic Similarity for Arabic Language Questions. In *IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA*. 1586–1591.

[3] Salha M Alzahrani, Naomie Salim, and Ajith Abraham. 2011. Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 2 (2011), 133–149.

[4] Yuki Arase and Junichi Tsujii. 2021. Transfer fine-tuning of BERT with phrasal paraphrases. *Computer Speech & Language* 66, 1 (2021), 101–164.

[5] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *5th International Conference on Learning Representations (ICLR), Toulon, France*. 1–16.

[6] Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics* 7, 1 (2019), 597–610.

[7] Alberto Barrón-Cedeno. 2012. On the Mono-and Cross-Language Detection of Text Re-Use and Plagiarism (Thesis). *Departmento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia* (2012).

[8] Alberto Barrón-Cedeño, Marta Vila, M Antònia Martí, and Paolo Rosso. 2013. Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. *Computational Linguistics* 39, 4 (2013), 917–947.

[9] Steven Burrows, Martin Potthast, and Benno Stein. 2013. Paraphrase acquisition via crowdsourcing and machine learning. *ACM Transactions on Intelligent Systems and Technology (TIST)* 4 (2013).

[10] Sergey Butakov and Vladislav Scherbinin. 2009. The toolbox for local and global plagiarism detection. *Computers & Education* 52, 4 (2009), 781–788.

[11] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, Canada*. 1–14.

[12] Chaka Chaka. 2023. Detecting AI content in responses generated by ChatGPT, YouChat, and Chatsonic: The case of five AI content detection tools. *Journal of Applied Learning and Teaching* 6, 2 (2023), 1–11.

[13] Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards Better UD Parsing: Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Brussels, Belgium*. 55–64.

[14] Hussain A Chowdhury and Dhruba K Bhattacharyya. 2018. Plagiarism: Taxonomy, tools and detection techniques. *In Proceedings of the 19th National Convention on Knowledge, Library and Information Networking (NACLIN), Visakhapatnam, India* (2018), 1–17.

[15] Ho Chui Chui. 2023. ChatGPT as a Tool for Developing Paraphrasing Skills Among ESL Learners. *Journal of Creative Practices in Language Learning and Teaching (CPLT)* 11, 2 (2023), 85–105.

[16] Paul Clough and Robert Gaizauskas. 2009. Corpora and text re-use. *Handbook of corpus linguistics, handbooks of linguistics and communication science* (2009), 1249–1271.

[17] Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Computational Linguistics* 34, 4 (2008), 597–614.

[18] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), Online*. 8440–8451.

[19] Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems* 32, 1 (2019), 1–10.

[20] Seniz Demir, Ilknur Durgar El-Kahlout, Erdem Unal, and Hamza Kaya. 2012. Turkish Paraphrase Corpus.. In *In Proceedings of the the Eighth International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey*. 4087–4091.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, USA*. 4171–4186.

[22] Mohamed I El Desouki, Wael H Gomaa, and Hawaf Abdalhakim. 2019. A hybrid model for paraphrase detection combines pros of text similarity with deep learning. *Int. J. Comput. Appl* 975, 8887 (2019), 1–06.

[23] Faouzia Benabbou El Mostafa Hambi. 2020. A new online plagiarism detection system based on deep learning. *International Journal of Advanced Computer Sciences and Applications* 11, 9 (2020), 470–478.

[24] Mohamed A El-Rashidy, Ramy G Mohamed, Nawal A El-Fishawy, and Marwa A Shouman. 2024. An effective text plagiarism detection system based on feature selection and SVM techniques. *Multimedia Tools and Applications* 83, 1 (2024), 2609–2646.

[25] Zahra Sadat Hosseini Moghadam Emami, Shohreh Tabatabayiseifi, Mohammad Izadi, and Mohammad Tavakoli. 2021. Designing a deep neural network model for finding semantic similarity between short persian texts using a parallel corpus. In *Proceedings of the International Conference on Web Research (ICWR), Tehran, Iran*. 91–96.

[26] Fezeh Esteki and Faramarz Safi Esfahani. 2016. A Plagiarism Detection Approach Based on SVM for Persian Texts.. In *In Proceedings of the Forum for Information Retrieval Evaluation (FIRE) (Working Notes), Kolkata, India*. 149–153.

[27] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria*. Association for Computational Linguistics, 1608–1618.

[28] Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, Gothenburg, Sweden*. 271–276.

[29] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT Sentence Embedding. *arXiv preprint arXiv:2007.01852* (2020).

[30] Tomáš Foltỳnek, Norman Meuschke, and Bela Gipp. 2019. Academic plagiarism detection: a systematic literature review. *Comput. Surveys* 52, 1 (2019), 1–42.

[31] Tomáš Foltỳnek, Terry Ruas, Philipp Scharpf, Norman Meuschke, Moritz Schubotz, William Grosky, and Bela Gipp. 2020. Detecting machine-obfuscated plagiarism. In *Proceedings of the iConference, 15th International Conference, iConference 2020, Boras, Sweden*. 816–827.

[32] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Atlanta, Georgia*. 758–764.

[33] Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research* 57 (2016), 345–420.

[34] Yoav Goldberg and Graeme Hirst. [n. d.]. Neural Network Methods in Natural Language Processing. Morgan & Claypool Publishers(2017). *9781627052986 (zitiert auf Seite 69)* ([n. d.]).

[35] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *In Proceedings of the International Conference on Language Resources and Evaluation (LREC), Miyazaki, Japan*. 3483–3487.

[36] Hamza Hafeez, Iqra Muneer, Muhammad Sharjeel, Muhammad Adnan Ashraf, and Rao Muhammad Adeel Nawab. 2023. Urdu Short Paraphrase Detection at Sentence Level. *ACM Transactions on Asian and Low-Resource Language Information Processing* 22, 4 (2023), 1–20.

[37] Samar Haider. 2018. Urdu word embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC), Miyazaki, Japan*. 964–968.

[38] Alami Hamza, Noureddine En-Nahnahi, and Said El Alaoui Ouatik. 2020. Contextual word representation and deep neural networks-based method for Arabic question classification. *Advances in Science, Technology and Engineering Systems Journal* 5, 5 (2020), 478–484.

[39] Israr Hanif, Rao Muhammad Adeel Nawab, Affiffa Arbab, Huma Jamshed, Sara Riaz, and Ehsan Ullah Munir. 2015. Cross-language Urdu–English (clue) text alignment corpus. *Cross-Language Urdu-English (CLUE) Text Alignment Corpus: Notebook for PAN at CLEF, Toulouse, France* (2015), 1–09.

[40] Syed Fawad Hussain and Asif Suryani. 2015. On retrieving intelligently plagiarized documents using semantic similarity. *Engineering Applications of Artificial Intelligence* 45, 1 (2015), 246–258.

[41] Hafiz Rizwan Iqbal, Rashad Maqsood, Agha Ali Raza, and Saeed-Ul Hassan. 2023. Urdu paraphrase detection: A novel DNN-based implementation using a semi-automatically generated corpus. *Natural Language Engineering* 30, 1 (2023), 354–384.

[42] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

[43] Huong T Le, Dung T Cao, Trung H Bui, Long T Luong, and Huy Q Nguyen. 2021. Improve Quora Question Pair Dataset for Question Similarity Task. In *In Proceedings of the 2021 RIVF International Conference on Computing and Communication Technologies (RIVF), Hanoi, Vietnam*. IEEE, 1–5.

[44] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).

[45] Adnen Mahmoud and Mounir Zrigui. 2022. Siamese AraBERT-LSTM Model based Approach for Arabic Paraphrase Detection. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC), Manila, Philippines*. 545–553.

[46] Hermann A Maurer, Frank Kappe, and Bilal Zaka. 2006. Plagiarism-A survey. *J. UCS* 12, 8 (2006), 1050–1084.

[47] Gull Mehak, Iqra Muneer, and Rao Muhammad Adeel Nawab. 2023. Urdu Text Reuse Detection at Phrasal level using Sentence Transformer-based approach. *Expert Systems with Applications* 234, 1 (2023), 121–163.

[48] Tomas Mikolov, Kai Chen, G. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *In Proceedings of the International Conference on Learning Representations (ICLR), Scottsdale, Arizona*. 1–12.

[49] Yusuke Mori, Hiroaki Yamane, Yusuke Mukuta, and Tatsuya Harada. 2020. Finding and generating a missing part for story completion. In *In Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, Online*. 156–166.

[50] Sharjeel Muhammad. 2020. *Mono-and cross-lingual paraphrased text reuse and extrinsic plagiarism detection (Thesis)*. Ph. D. Dissertation. Lancaster University, U.K.

[51] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*.

[52] Ayesha Noreen, Iqra Muneer, and Rao Muhammad Adeel Nawab. 2024. Mono-lingual text reuse detection for the Urdu language at lexical level. *Engineering Applications of Artificial Intelligence* 136, 1 (2024), 109–123.

[53] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, USA*. 2227–2237.

[54] Namoos Hayat Qasmi, Haris Bin Zia, Awais Athar, and Agha Ali Raza. 2020. SimplifyUR: Unsupervised Lexical Text Simplification for Urdu. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC), Marseille, France*. 3484–3489.

[55] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018), Preprint. 1–12.

[56] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China.* 3982–3992.

[57] Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online.* 4512–4525.

[58] Sara Sameen, Muhammad Sharjeel, Rao Muhammad Adeel Nawab, Paul Rayson, and Iqra Muneer. 2017. Measuring short text reuse for the Urdu language. *IEEE Access* 6, 1 (2017), 7412–7421.

[59] Jawad Shafi. 2019. *An Urdu Semantic Tagger-Lexicons, Corpora, Methods and Tools (Thesis).* The Lancaster University, United Kingdom.

[60] Jawad Shafi, Rao Muhammad Adeel Nawab, and Paul Rayson. 2023. Semantic Tagging for the Urdu Language: Annotated Corpus and Multi-Target Classification Methods. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 22, 6 (2023), 1–32.

[61] Jawad Shafi, Hafiz Rizwan Iqbal, Rao Muhammad Adeel Nawab, and Paul Rayson. 2022. UNLT: Urdu Natural Language Toolkit. *Natural Language Engineering* (2022), 1–36.

[62] Muhammad Sharjeel. 2020. *Mono-and cross-lingual paraphrased text reuse and extrinsic plagiarism detection (Thesis).* Lancaster University, U.K.

[63] Muhammad Sharjeel, Rao Muhammad Adeel Nawab, and Paul Rayson. 2017. COUNTER: corpus of Urdu news text reuse. *Language Resources and Evaluation* 51, 3 (2017), 777–803.

[64] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS), Long Beach California, USA.* 6000–6010.

[66] Tedo Vrbanec and Ana Meštrović. 2020. Corpus-based paraphrase detection experiments and review. *Information* 11, 5 (2020), 1–24.

[67] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards Universal Paraphrastic Sentence Embeddings. In *4th International Conference on Learning Representations, (ICLR), San Juan, Puerto Rico.* 1–19.

[68] John Wieting and Kevin Gimpel. 2017. Revisiting recurrent networks for paraphrastic sentence embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL), Vancouver, Canada.* 2078–2088.

[69] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Online.* 87–94.

[70] Rohola Zandie and Mohammad H Mahoor. 2023. Topical language generation using transformers. *Natural Language Engineering* 29, 2 (2023), 337–359.

[71] Ali Zareshahi, MohammadAli Javadzade, and Esmaeel Bastami. 2024. Measuring Semantic Similarity of Persian Sentences Using ParsBERT Model. In *Proceedings of the International Conference on Artificial Intelligence and Robotics (QICAR), Singapore.* 316–321.

[72] Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2017. Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora, Vancouver, Canada.* 60–67.

Received 30 June 2024; revised 23 October 2024; revised 26 November 2024; accepted 02 December 2024