Natural Language Engineering **1** (1): 1–40. Printed in the United Kingdom

Urdu Paraphrase Detection: A Novel DNN-based Implementation using a Semi-Automatically Generated Corpus

Hafiz Rizwan Iqbal[1], Rashad Maqsood[1] Agha Ali Raza[2], Saeed-Ul Hassan[3*]

[1]Information Technology University, Lahore, Pakistan
[2]Lahore University of Management Sciences, Lahore, Pakistan
[3]Manchester Metropolitan University, Manchester, United Kingdom
[*]Corresponding author email: agha.ali.raza@lums.edu.pk

(*Received 10 March 2022; revised xxxxxxx*)

Abstract

Automatic paraphrase detection is the task of measuring the semantic overlap between two given texts. A major hurdle in the development and evaluation of paraphrase detection approaches, particularly for South Asian languages like Urdu, is the inadequacy of standard evaluation resources. The very few available paraphrased corpora for these languages are manually created. As a result, they are constrained to smaller sizes and are not very feasible to evaluate mainstream data-driven and deep neural networks based approaches. Consequently, there is a need to develop semi or fully automated corpus generation approaches for the resource-scarce languages. There is currently no semi- or fully automatically generated sentence-level Urdu paraphrase corpus. Moreover, no study is available to localize and compare approaches for Urdu paraphrase detection that focus on various mainstream deep neural architectures and pre-trained language models.

This research study addresses this problem by presenting a semi-automatic pipeline for generating paraphrased corpora for Urdu. It also presents a corpus that is generated using the proposed approach. This corpus contains 3,147 semi-automatically extracted Urdu sentence pairs that are manually tagged as paraphrased (854) and non-paraphrased (2,293). Finally, this paper proposes two novel approaches based on deep neural networks for the task of paraphrase detection in Urdu text. These are Word Embeddings *n-gram* Overlap (henceforth called WENGO), and a modified approach, Deep Text Reuse and Paraphrase Plagiarism Detection (henceforth called D-TRAPPD). Both of these approaches have been evaluated on two related tasks: (i) paraphrase detection, and (ii) text reuse and plagiarism detection. The results from these evaluations revealed that D-TRAPPD ($F_1 = 96.80$ for paraphrase detection and $F_1 = 88.90$ for text reuse and plagiarism detection) outperformed WENGO ($F_1 = 81.64$ for paraphrase detection and $F_1 = 61.19$ for text reuse and plagiarism detection) as well as other state-of-the-art approaches for these two tasks. The corpus, models, and our implementations have been made available as free to download for the research community.

1 Introduction

Automatic paraphrase detection is the task of deciding whether two given text fragments have the same meaning or not (Wang *et al.* 2021). Paraphrase detection has a number of applications including question-answering (Noraset *et al.* 2021), natural language generation (Zandie *et al.* 2022; Paris *et al.* 2013), and intelligent tutoring systems (Forsythe *et al.* 2006). In question-answering, multiple paraphrased answers could be considered as evidence for the correctness of an answer (Noraset *et al.* 2021). For intelligent tutoring systems with natural language input (Forsythe *et al.* 2006), paraphrase detection (Agarwal *et al.* 2018) is useful to assess the match between expected answers and the answers provided by the students. In addition to these uses, paraphrase detection is also important for information extraction(Ji *et al.* 2020), machine translation (Farhan *et al.* 2020), information retrieval (Ehsan *et al.* 2016), automatic identification of copyright infringement (Clough *et al.* 2002; Jing *et al.* 2021), and text reuse and plagiarism detection. In recent years, the detection of paraphrased cases of plagiarism has also attracted the attention of the research community.

Text reuse can formally be defined as the conscious extraction of the selected text pieces from an existing text to produce a new one (Clough *et al.* 2002). Text reuse spectrum ranges from the simple scenarios of word-for-word (aka verbatim) copying, paraphrasing (insertion, deletion, substitution, word reordering), and reusing of ideas, to the more complex scenario in which the same event is written independently by two different authors belonging to the same language and context (Clough *et al.* 2003).

Text plagiarism (aka the unacknowledged reuse of text) is a counterpart to text reuse. In text plagiarism the author intentionally or unintentionally reuses the text from a single or multiple sources without acknowledgment of the original source (Barron *et al.* 2010; Cadeno *et al.* 2013; Nawab *et al.* 2012). In plagiarism, the writer can often change the surface form to keep the source(s) hidden from the reader (Clough *et al.* 2002).

It is not easy to differentiate between plagiarism and various types of text reuse. However, from the perspective of computational linguistics and natural language processing (NLP), both plagiarism and text reuse are similar tasks (Cadeno *et al.* 2013; Clough *et al.* 2003) because they share an almost identical authoring environment. For instance, in the journalism industry, an experienced plagiarizer is a person who is highly skilled in text editing. Eventually, nearly all types of rewritings (e.g., paraphrasing) in journalism and academia are quite similar (Barron *et al.* 2012). Therefore, we will consider both tasks as equivalent and will hereafter use both terms interchangeably or in combined form as "text reuse and plagiarism".

Paraphrasing is a linguistic technique that is employed in almost every text reuse and plagiarism case (Barron *et al.* 2012; Barrón-Cedeño *et al.* 2013). It occurs when someone generates new text from preexisting text while preserving its meaning (Burrows *et al.* 2013). It is performed over text using different text altering operations, including deletion (e.g. of repeating contexts as a result of syntactic modifications), lexical substitutions (e.g., replacing words with their synonyms),

structural changes (e.g., word reordering, switching between active and passive voice tenses), and summarizing (Clough *et al.* 2009). Moreover, from the NLP perspective, researchers have also proposed various paraphrase typologies (Barron *et al.* 2012; Muhammad2020) to cover different types of text alteration mechanisms used by author(s) rephrase the source text.

Text plagiarism is becoming very common due to the free and ready availability of large amounts of text online and this has become a cause of alarming for academics, publishers, and authors alike (Foltynek *et al.* 2019). Surveys in the past (Maurer *et al.* 2006; Butakov *et al.* 2009) reported that a majority of students were involved in some form of plagiarism, and most of them committed plagiarism in their assignments. According to a report on Cyber Plagiarism [1], 66% students out of a sample of 16,000, from 31 top-ranked U.S. universities, admitted to cheating. In Germany, more than 200 academic plagiarism cases were found in a crowd-sourcing project (Foltynek *et al.* 2019). In Pakistan, 20 researchers from various Pakistani universities were blocklisted in 2015 by the Higher Education Commission (HEC) of Pakistan for their plagiarized work, while the number of reported cases are were even higher than this. It can reasonably be assumed that, if plagiarism and illegal reuse of text remain undiscovered, the outcomes will be even more severe, which may include artificial inflation in publications, distorted competence among students, and undue career advancements and research grants (Foltynek *et al.* 2019).

Various studies (Potthast *et al.* 2010; Potthast *et al.* 2013; Barrón-Cedeño *et al.* 2013; Franco *et al.* 2026) by the research community have shown that detecting paraphrased plagiarism presents major challenges. A hindrance to research in automatic paraphrase detection, especially for Urdu and other South Asian languages, is the lack of large-scale labeled paraphrased corpora. The majority of the available resources for paraphrase detection are developed either for English (Dolan *et al.* 2005; Alvi *et al.* 2012; Barrón-Cedeño *et al.* 2013) or other resource-rich languages (Ganitkevitch *et al.* 2013; Xu *et al.* 2015; Al-Bataineh *et al.* 2019). However, there is a dearth of such resources for South Asian languages including Urdu.

Urdu is a widely spoken language with around 231 million speakers worldwide (mostly in the Indian sub-continent)[2]. It is a free word order language, derived from the Hindustani/Sanskrit language and influenced majorly by Turkish, Arabic, and Persian (Sharjeel et al. 2017). Urdu is a highly inflected and morphologically rich language because gender, case, number, and forms of verbs are expressed by morphology. Additionally, there are numerous multi-word expressions in Urdu and letters whose shapes can vary depending on the context (Shafi *et al.* 2021). Over the last decade, the digital footprint of Urdu has increased exponentially. However, the language lacks severely in terms of computational tools and standard evaluation resources (Daud *et al.* 2017).

Recently, Sharjeel et al. developed the first-ever paraphrased corpus for Urdu at the document-level, called the Urdu Paraphrase Plagiarism Corpus (Sharjeel *et al.*

[1] https://www.checkforplagiarism.net/cyber-plagiarism
[2] https://www.ethnologue.com/guides/ethnologue200

2016). Moreover, a handful of corpora have also been developed for the related task of text reuse and plagiarism detection (Sharjeel *et al.* 2017; Sameen *et al.* 2017; Haneef *et al.* 2019; Muneer *et al.* 2019) in Urdu. Furthermore, several state-of-the-art surface-level string-similarity-based approaches have been applied on these standard evaluation resources to show their usefulness in the task of text reuse and plagiarism detection in Urdu.

However, even these basic approaches have not been evaluated for the task of paraphrase detection in Urdu. These corpora have been created manually, which is both time consuming and labor-intensive. Although they provide a good baseline to further explore Urdu text reuse and plagiarism detection tasks, their limited size is a major drawback for their utilization in mainstream data-driven and deep neural networks (DNN) based approaches. As a result, the development of novel approaches for Urdu paraphrase detection, and text reuse and plagiarism detection tasks has been constrained. This highlights the fact that to create large-scale standard evaluation resources for Urdu (and similar resource-poor languages), it is important to develop semi- or fully automatic corpus generation approaches.

Therefore, it can be deduced that there is currently no semi- or fully automatically generated sentence-level Urdu paraphrase corpus with examples of paraphrased and non-paraphrased sentence pairs. Moreover, there is no research study to compare various DNN-based architectures, including CNNs and LSTMs, that use pre-trained embedding models for paraphrase detection, and text reuse and plagiarism detection in Urdu texts.

This research work focuses on answering the following research questions: (i) how to create a semi- or fully automatically generated corpus for paraphrase detection in Urdu; (ii) whether it is possible to differentiate between different levels of Urdu paraphrasing using the mainstream DNN-based approaches; and (iii) whether the DNN-based approaches perform better than the traditional approaches that measure surface-level similarity between two sentences for Urdu paraphrase detection, and text reuse and plagiarism detection.

In this paper, we present a semi-automatically generated sentence-level paraphrased corpus for Urdu. The "Semi-automatic Urdu Sentential Paraphrase Corpus" (henceforth called SUSPC) contains a total of 3,147 sentence-pairs marked as either paraphrased (854) or non-paraphrased (2,293). This is the first-ever semi-automatically created sentence-level paraphrased corpus developed for Urdu with manual annotations. The proposed corpus would benefit the Urdu NLP community in several ways: (i) it would reduce the scarcity of the publicly available corpora for Urdu paraphrased detection; (ii) it would present a less expensive and quick approach to creating a corpus for paraphrase detection; (iii) it would provide empirical evidence that an existing approach (Dolan *et al.* 2005) can be utilized to automatically generate a paraphrase corpus for Urdu; (iv) it would present an adequate number of semantically equivalent sentence pairs in natural Urdu; and (v) It would demonstrate using state-of-the-art supervised learning approaches for Urdu paraphrase detection.

As another contribution, we have proposed two DNN-based approaches: (i) a novel approach WENGO and (ii) a modified approach D-TRAPPD. Both ap-

proaches are evaluated on two related tasks: (i) Urdu paraphrase detection, and (ii) Urdu text reuse and plagiarism detection. Results show that the proposed D-TRAPPD approach has not only established a strong baseline for the paraphrase detection task in Urdu but also outperformed the state-of-the-art surface-level string similarity approaches (Sameen *et al.* 2017) for Urdu text reuse and plagiarism detection in both binary classification ($F_1 = 78.5$) and multi-classification ($F_1 = 88.90$) tasks.

We have made our corpus, models, and implementation freely available for the research community [3]. We believe that the SUSPC corpus and DNN-based approaches presented in this research work will help (i) analyze and develop efficient paraphrase detection systems, specifically for Urdu; (ii) provide a detailed comparison of the DNN-based approaches on a variety of tasks and corpora; and (iii) further motivate research in Urdu paraphrase, and text reuse and plagiarism detection.

The rest of the article is organized as follows. Section 2 describes the related work. Section 3 presents the newly proposed corpus creation process, its statistics, and standardization. Section 4 describes the details of the approaches used to detect Urdu paraphrases. Section 5 explains the experimental set-up, evaluation tasks, text prepossessing, and evaluation measures. Section 6 presents results and their analysis. Finally, Section 7 presents the conclusion.

2 Literature Review

This section presents the details of the corpora and the approaches developed for the task of automatic paraphrase detection in the past.

2.1 Corpora

Developing a large-scale standard evaluation resource manually to investigate paraphrase detection is a difficult task since since it is time consuming and labor intensive. There have been efforts made in the past to develop benchmark corpora for paraphrase detection. Several benchmark corpora have been developed for English (e.g. (Dolan *et al.* 2005; Alvi *et al.* 2012; Barrón-Cedeño *et al.* 2013; Nighojkar *et al.* 2021; Kadotani *et al.* 2021; Meng *et al.* 2021; Corbeil *et al.* 2021)) along with other languages (Ganitkevitch *et al.* 2013; Xu *et al.* 2015; Al-Bataineh *et al.* 2019). An in-depth discussion of all these corpora are beyond the scope of this study. This research work focuses on some of the most prominent studies concerning the sentence-level corpora for English and Urdu.

Microsoft Research Paraphrase Corpus (MRPC) (Dolan *et al.* 2005) was one of the pioneering efforts to generate sentence-level paraphrased corpora using automatic corpus generation approaches. MRPC was developed to foster the research and the development of automatic paraphrase detection systems for English. It contained 5,801 sentence pairs, each annotated manually as either paraphrased or

[3] https://www.dropbox.com/sh/kmxjuq170i66tx2/AACHWZXIkjpCnE44EvQcWxoCa?dl=0

non-paraphrased. Heuristic filters, along with a Support Vector Machine (SVM) classifier, were used to extract likely paraphrased sentence pairs from 32,408 news clusters gathered from the internet over a period of 2 years. Three human annotators manually annotated the resulting sentence pairs to classify them as either paraphrased or non-paraphrased. Out of the 5,801 extracted sentence pairs, 67% were classified as paraphrased while the other 33% were classified as non-paraphrased.

The PAN-PC corpora, an outcome of PAN (Plagiarism analysis, Authorship attribution, and Near-duplicate detection) [4] shared the different tasks (Sanchez *et al.* 2014) involved in plagiarism detection including paraphrased plagiarism detection. PAN-PC is a set of three benchmark corpora: PAN-PC-09 (Stein *et al.* 2009), PAN-PC-10 (Potthast *et al.* 2010), and PAN-PC-11 (Potthast *et al.* 2011). These corpora have various features, such as intrinsic and extrinsic plagiarism cases, translated cases of plagiarism from German and Spanish languages to English, and a variety of plagiarism types (verbatim, paraphrased, independently written) created artificially and manually.

In a related study, Barron et. al (Barrón-Cedeño *et al.* 2013) presented a P4P (Paraphrase for Plagiarism) corpus by extracting simulated paraphrasing plagiarism cases from the PAN-PC-10 corpus (Potthast *et al.* 2010). The P4P corpus was created by manually annotating a portion of the PAN-PC-10 corpus using a newly proposed paraphrasing typology and guidelines from MRPC. It contained 847 paraphrased sentence pairs, each containing a source and a plagiarised sentence, where the latter is created by applying different paraphrasing operations defined in the new paraphrasing typology. Moreover, each sentence contained 50 or fewer words in accordance with the guidelines from MRPC, which considers the average sentence to contain 28 words. Later, Alvis et al. (Alvi *et al.* 2012) extracted another sentence-level paraphrased corpus from the P4P corpus.

The recent trends in automatic paraphrase generation are not over-reliant only on the lexical and syntactic properties of the text pairs. Instead, researchers have used various methods like formality transfer (Kadotani *et al.* 2021), adversarial paraphrasing (Nighojkar *et al.* 2021), Seq2Seq paraphrase generation (Meng *et al.* 2021), transformer-based back translation (Corbeil *et al.* 2021), etc., to produce high-quality paraphrased text pairs. Moreover, these corpora have been extensively evaluated under supervised and unsupervised experimental environments to show their effectiveness in paraphrase detection.

Literature shows that the paraphrased plagiarism corpora developed for English and other resource-rich languages is responsible for creating a significant stumbling block in the way of research and development of less resourced languages like Urdu. Although limited gold standard corpora (Sharjeel *et al.* 2017; Sameen *et al.* 2017; Sharjeel *et al.* 2016) are available for Urdu, which cover document and passage-level plagiarism, particularly paraphrased plagiarism, these have all been created manually.

We found only one Urdu corpus in the literature for the task of paraphrased pla-

[4] https://pan.webis.de/shared-tasks.html

giarism detection, and this was the "Urdu Paraphrase Plagiarism Corpus" (UPPC)[5] (Sharjeel *et al.* 2016). The UPPC corpus (Sharjeel *et al.* 2016) was the pioneering attempt to promote research in Urdu paraphrased plagiarism detection, complete with simulated cases of paraphrased plagiarism. It is a document level corpus that contains 160 documents, among which 20 are the source, 75 are paraphrased plagiarised (PP), and 65 are non-plagiarised (NP). The corpus in total contains 2,711 sentences, 46,729 words, and 6,201 unique words. Wikipedia articles about 20 celebrities from different domains (historical, religious, and political) were used as source documents for this corpus. These were paraphrased by graduate-level university students to generate plagiarised documents. The plagiarised documents also contained typing mistakes (typos) to simulate real-world scenarios when plagiarists paraphrase texts. The non-plagiarised documents were created by consulting books and essays as sources. Although UPPC is a useful resource for Urdu paraphrased plagiarism detection, it has a number of limitations. Since it is manually created, it contains only a small number of document pairs. In addition, the size of the plagiarised documents (between 200 and 300 words) is also short as compared to real academic essays. The documents also only contain text about celebrities. Lastly, since the simulated cases were generated in a controlled environment using crowdsourcing approaches, they do not adequately demonstrate the practices followed by plagiarists in real-life scenarios.

Although, this paper focuses on the task of paraphrase detection, it is worthwhile to also include two Urdu corpora developed for text reuse and plagiarism detection. The two corpora are as follows: (i) COrpus of Urdu News TExt Reuse (COUNTER)[6] (Sharjeel *et al.* 2017) and (ii) Urdu Short Text Reuse Corpus (USTRC)[7] (Sameen *et al.* 2017).

The COUNTER corpus (Sharjeel *et al.* 2017) is a remarkable effort for the detection of monolingual text reuse and plagiarism in Urdu text. It has been developed using the guidelines of the well known MEasuring TExt Reuse (METER) corpus (Gaizauskas *et al.* 2001) of the English language. COUNTER is also a document-level corpus containing 600 document pairs that are manually annotated as Wholly Derived (WD, 135), Partially Derived (PD, 288), or Non-Derived (ND, 177). The corpus contains 10,841 sentences, 275,387 words, and 21,426 unique words. The largest source document contains 1,377 words while the largest derived document consists of 2,481 words. The average length of a source document is 227 words while for the derived documents the average length is 254 words. In this corpus, the source news articles have been collected from various news stories released by five Pakistani news agencies and included stories about business, showbiz, sports, and national and foreign affairs. The derived articles were taken from the same news stories published in 9 different top Urdu newspapers. COUNTER is a useful benchmark resource to design and evaluate automatic monolingual text reuse and

[5] http://ucrel.lancs.ac.uk/textreuse/uppc.php
[6] http://www.research.lancs.ac.uk/portal/en/datasets/corpus-of-urdu-news-text-reuse-counter(5b0be889-e0eb-4a9c-8441-d6723ecfb617).html
[7] http://ucrel.lancs.ac.uk/textreuse/ustrc.php

plagiarism detection systems for Urdu. However, the corpus contains only a small number of document pairs for each level of text reuse since it is difficult to create a corpus with real examples of text reuse and plagiarism because of confidentiality issues (Clough *et al.* 2003).

Another benchmark resource that consists of real cases of sentence/passage-level text reuse and plagiarism for Urdu is USTRC (Sameen *et al.* 2017). It contains 2,684 manually extracted short text pairs from 600 news document pairs (in which the news agency's text is treated as source and the newspaper text is considered as reused text). The annotators manually classified these short text pairs into Verbatim (V, 496), Paraphrased (P, 1,329), and Independently Written or Non-Paraphrased (I,859). The source (news agency) texts were taken from the Associated Press of Pakistan (APP[8]) while the derived texts were extracted from the top 4 newspapers in Pakistan. Both the source and the derived news texts were in Urdu and included stories from various news sections including politics, sports, technology, business, entertainment, and foreign and national affairs. None-the-less, confidentiality constraints in getting real cases of plagiarism and the labor-intensive nature of USTRC became reasons why this too remained only a small-size corpus.

Table 1 presents the summarized view of the available corpora developed for Urdu text reuse and plagiarism detection and their characteristics. It can be observed that all of the available corpora consist of real and simulated cases of plagiarism from either journalism or academia. The number of cases included in each corpora is also limited because (i) it is difficult to gather real cases of plagiarism from academia due to confidentiality and ethical issues, and (ii) the manual creation process itself is a labour-intensive and time consuming task.

Table 1: Existing corpora for Urdu paraphrase and text reuse and plagiarism detection. For document-level corpora, the size indicates the total number of documents, including both the source and the suspicious documents and is equal to the summation of source and suspicious. In the case of sentence-level corpora, the size indicates the number of pairs, where each pair consists of the source and the corresponding suspicious sentences.

Corpus	**Reuse Type**	**Text Length**	**Obfuscation Levels**	**Size(Source/Suspicious)**	**Text Domain**	**Free**
COUNTER	Real	Document	Wholly Derived, Partially Derived, Non-Derived	1200(600/600)	Journalism	Yes
UPPC	Simulated	Document	Paraphrased, Non-paraphrased	160(20/140)	Academic	Yes
USTRC	Real	Sentence	Verbatim, Paraphrased, Non-paraphrased	2684(2684/2684)	Journalism	Yes

To sum up, there are only a few corpora available that can be used for monolingual paraphrase detection in Urdu. Moreover, they are much smaller in size as compared to the corpora that are available for other popular languages such as English, the major reason being that they have been created manually, thus

[8] https://www.app.com.pk/

requiring time and labor. Therefore, it is the need of the hour to either adopt existing or to develop semi- or fully automatic corpus generation approaches for quick production of Urdu corpora for paraphrase detection and similar tasks. This research study presents a novel semi-automatically generated Urdu sentence-level paraphrase corpus (SUSPC), which consists of 3,147 semi-automatically extracted Urdu text pairs that are then manually tagged as either paraphrased (854) or non-paraphrased (2,293). To the best of our knowledge, the proposed corpus is novel, unique, semi-automatically generated, and the largest sentence-level paraphrased corpus ever developed for Urdu.

2.2 Approaches

Over the years, various monolingual paraphrase detection approaches have been proposed. These can be classified into: (i) surface, (ii) fuzzy, (iii) semantics, and (iv) deep neural network (DNN) based approaches (Alzahrani *et al.* 2011; Agarwal *et al.* 2018; Desouki *et al.* 2019; Muhammad2020). This research work only presents the DNN-based approaches for monolingual paraphrase detection. DNN-based approaches can be sub-categorized into (i) word/phrase/sentence embeddings, (ii) Convolutional Neural Networks (CNNs), (iii) Recurrent Neural Networks(RNNs)/Long Short-Term Memory(LSTM), and (iv) CNNs-RNNs/LSTM-based approaches.

Wieting et al. (Wieting *et al.* 2015) proposed an approach to learn the paraphrastic sentence embeddings by simply averaging the word embeddings learned from the Paraphrased Pair Database (PPDB) (Ganitkevitch *et al.* 2013). It has been observed that this does not perform well due to the crucial need for supervision from the PPDB dataset. In comparison, Arora et al. (Arora *et al.* 2017) trained word embeddings in an unsupervised way on unlabeled texts from Wikipedia's. The sentences were represented as weighted average vectors of all the words, leading to a 10% to 30% improvement in results. Wieting et al. (Wieting *et al.* 2017) also proposed Gated Recurrent Averaging Network (GRAN), under which, instead of training on phrase pairs, sentence pairs were used and their states were averaged with an aggressive regularization for sequences representation. However, the results for the paraphrase detection task outperformed the approach proposed by Wieting et al. (Wieting *et al.* 2015).

The inclusion of context in word embeddings has been proven to be a watershed idea in NLP as exemplified by Embeddings from Language Model (ELMO) (Peters *et al.* 2018). Its embeddings are context-sensitive because ELMO considers the context of the words and how they are used in the running text. This indicates that ELMO embeddings contain more information and thus probably increase performance. For paraphrase detection tasks, ELMO has outperformed the periphrastic and other non-contextual static word embeddings based approaches. Al-Bataineh et al. (Al-Bataineh *et al.* 2019) who presented paraphrase detection based on deep contextualized embeddings for Modern Standard Arabic (MSA), trained their contextualized word embeddings using ELMOon a corpus containing MSA and 24 other renowned Arabic dialects. In another study, Vrbanec et al. (Vrbanec *et al.* 2020)

reported a performance comparison of eight different vector-based word representation models. Their findings showed that the word representation models based on deep learning outperformed the conventional state-of-the-art models for semantic level sentence similarity and paraphrase detection tasks.

Transformers, the new state-of-the-art models in NLP, particularly in paraphrase detection, have demonstrated that incorporating attention along with the context in word embeddings is revolutionary. Transformers use attention mechanism to decide at each step which parts of the input sequence are important. Generative Pre-Trained Transformers (GPT, GPT-2, GPT-3) (Radford *et al.* 2018) and (BERT) (Devlin *et al.* 2018) are two renowned transformers-based pre-trained language models. OpenAI GPT is based on an idea similar to ELMO though it trains the language model in an unsupervised fashion and on a much larger collection of textual data. GPT differs from ELMO in two ways. Firstly, both models have different architectures. ELMO trains two independent LSTMs (left-to-right and right-to-left) and uses shallow concatenation to produce joint representation, while GPT, which is based on the renowned multi-layer transformers (Vaswani *et al.* 2017), predicts the future only in one direction, i.e., from left-to-right. Secondly, GPT and ELMO differ in their use of contextualized embeddings. GPT's empirical evaluation has been conducted on various NLP tasks including semantic similarity and paraphrase detection (Radford *et al.* 2018).

A contemporary of GPT is BERT, a language model trained on a huge collection of raw text and fine-tuned on specific tasks without customizing the underlying neural network. However, the bidirectional (left-to-right and right-to-left) training of BERT makes it different from GPT. BERT's architecture consists of a multi-layer bidirectional Transformer encoder. It is trained with two tasks: (i) Masked Language Model (MLM), which predicts the missing words in a sequence by randomly masking (i.e., replacing the selected tokens with placeholder [MASK]) 15% of its tokens, and (ii) Next Sentence Prediction (NSP), which is a binary classification task to decide whether a sentence follows another sentence. BERT has been evaluated on various NLP downstream tasks including paraphrase detection (Wang *et al.* 2018; Arase *et al.* 2021), and it has empirically shown that a representation that learns a context around a word rather than just after the word is better in capturing syntactic and semantic properties of the word (Devlin *et al.* 2018).

CNNs have established their worth in paraphrase detection and classification tasks with word embeddings representation (Kim2014). In their work, Wang et al. (Wang *et al.* 2016) have introduced a model that took into account both similarities and dissimilarities between a source-derived sentence pair. Similar and dissimilar components were computed for one sentence in relation with the other. These were fed to a single layer CNN model (Kim2014). The convolutional output gave feature representation for each input. This representation was absorbed by the similarity function, which gave a value for the prediction. This model produced outstanding results with respect to other state-of-the-art approaches. Yin et al. (Yin *et al.* 2016) reported paraphrase identification using Attention Based Convolutional Neural Networks (ABCNN). They conducted experiments for various paraphrase

identification tasks and showed that ABCNNs are much better than CNNs, which are without attention mechanisms.

Furthermore, Long Short-Term Memory (LSTMs — a special kind of Recurrent Neural Networks (RNNs)) (Hochreiter *et al.* 1997) have also been used widely for the task of paraphrase and textual semantic similarity detection. Mueller et al. (Mueller *et al.* 2016) used a Siamese adaptation of the LSTM model to get the hidden representation for sentence pairs. The similarity is predicted by the difference in the final representation. The work establishes that using a simple LSTM for extracting feature vectors easily exceeds the performance achieved by models that use carefully crafted features (Marelli *et al.* 2014). Similarly, Kleen et al. (Kleenankandy Nazeer *et al.* 2020) reported a relational gated LSTM architecture to model the relationship between two input sentences by controlling the input. They also proposed the Type Dependency Tree-LSTM model to embed sentence semantics into a dense vector by using sentence dependency type and parse structure. The proposed model achieved comparable scores to the other state-of-the-art paraphrase detection approaches.

For semantic level similarity and paraphrase detection tasks, CNNs have also been used with LSTMs. Kim et.al (Kim *et al.* 2015) proposed a neural language model for sentence semantics matching that takes a character as input but makes predictions at the word level. Over the characters, they used CNNs and a highway network for feature extraction, which is given as an input to the LSTMs. The results show that the proposed model is able to encode both semantics and orthographic information with only the input of the character. In addition to these, Wang et al. (Wang *et al.* 2017) also proposed a bilateral multi-perspective matching (BiMPM) model for two sentences. This model is based on Siamese-CNN, Multi-perspective CNN-LSTM, and BiLSTM. It encodes the two sentences in two directions and matches each time-step of a sentence with all the time-steps of the other sentence from multiple perspectives.

In another study, Agarwal et al. (Agarwal *et al.* 2018) reported a big neural architecture based on CNNs and LSTMs, along with surface-level string features to detect paraphrasing in clean and noisy text pairs. This CNN-LSTM-based model used CNNs to search local features, which were given as input to LSTMs to capture long-term dependencies. Moreover, a separate CNN that took a similarity matrix as input was also used. In addition to these, six different statistical features were also used that showed that the proposed approach outperformed the extant state-of-the-art approaches in terms of F_1 score for paraphrase detection. Finally, Shakeel et al. (Shakeel *et al.* 2020) detected enhanced paraphrasing in texts by developing a multi-cascaded neural model with data augmentation. They made use of efficiently generated paraphrased and non-paraphrased texts for data augmentation by using graph theory. They employed CNN-LSTM based supervised feature learners over these text pairs. These were provided to a discriminator network for classification with and without soft attention. Their results were at par with the state-of-the-art approaches.

To conclude, the existing DNN-based approaches have been thoroughly explored for several languages but not for Urdu. The present research work proposes two DNN-based approaches, (i) WENGO and (ii) D-TRAPPD based on (Agarwal *et*

al. 2018), for the detection of monolingual paraphrase text reuse and plagiarism in Urdu texts. To the best of our knowledge, the proposed DNN-based approaches have previously neither been developed for nor applied to Urdu for the task of detecting monolingual paraphrase text reuse and plagiarism.

3 Corpus Generation Process

This study presents the first semi-automatically generated Urdu paraphrased corpus at the sentence-level modeled on the original Microsoft Research Paraphrase Corpus (MRPC) approach (Dolan *et al.* 2005). The proposed corpus (i.e., SUSPC) is created by following the MRPC's approach (Dolan *et al.* 2005), with a few modifications to adapt it to Urdu, including the exclusion of a few rules, changes to the filter thresholds, and a few tweaks to the annotation guidelines[9]. The following sections describe the stages of construction and the components of the gold-standard SUSPC, including domain selection, data source, manual evaluation process, corpus statistics, and standardization of the corpus.

3.1 Extracting Sentence Pairs

3.1.1 Domain Selection

In order to develop SUSPC, we targeted the journalism industry. The choice of the journalism domain in SUSPC is motivated by the fact that it is comparatively easier to gather original and reproduced news stories from newspapers, since the majority of the newspapers are freely available in electronic form over the web. Moreover, it is straightforward to get real cases of paraphrasing, text reuse, and plagiarism, which is almost impossible in academia due to confidentiality issues. Further, it is a common practice in the newspaper industry to take the original text from news that is released by news agencies and to paraphrase it using different rewriting techniques (e.g., removing redundant words, changing word order, summarizing the text, inserting synonyms etc.) (Bell 1991; Fries 1997; Jing *et al.* 1999). In addition, the majority of the previously available Urdu text reuse and plagiarism corpora (Sharjeel *et al.* 2016; Sharjeel *et al.* 2016; Sameen *et al.* 2017) are based on newspapers, which is another reason why we chose newspapers for the construction of SUSPC.

3.1.2 Source Data

To develop SUSPC, we used the COUNTER corpus (see Section 2.1) as a source. The motivation behind the selection of this source is the dire need for news clusters that consist of topically and temporally coherent news stories. The proposed SUSPC is modeled on the footsteps of MRPC, which is extracted from 32,408 news clusters

[9] After multiple attempts with different combinations of values and rules, we have taken only those rules/filters that gave the best results on Urdu sentences.

Table 2: Distribution of the source sentence pairs w.r.t classes in the COUNTER corpus

Document Class	Sentence Pairs
Wholly Derived	8,352 (14.30%)
Partially Derived	28,223 (48.32%)
Non-Derived	21,831 (37.38%)
Total	58,406

that are coherent in topic and focus. These news clusters were collected from the internet over two years by the MRPC team (Dolan *et al.* 2004; Dolan *et al.* 2005). Such news clusters were not readily available for Urdu news stories. Therefore, we used the COUNTER corpus in which each document pair is considered a cluster of sentences from related news.

COUNTER is a benchmark text reuse corpus that is publicly available, widely used, and frequently cited, containing text from newspapers. COUNTER contains 1200 documents, mainly categorized into source and derived documents. The average length of a source document is nine sentences, whereas the average length of a derived document is eight sentences. The derived documents are further annotated as: 1) Wholly Derived (WD, 135 documents), i.e., most of the text in the document is a word-to-word copy of the text provided by the news agency, which is the only source of the news; 2) Partially derived (PD, 288 documents), i.e., most of the text is paraphrased from multiple news agencies with the addition of a few facts and figures by the journalist; and 3) Non-Derived (ND, 177 documents), i.e., most of the text is new either because a news agency's text was not used or the journalist heavily paraphrased the source news and/or incorporated new findings.

3.1.3 Candidate Pairs Search Space Reduction

In the construction of SUSPC, each source/derived document pair was considered as one class. In each class, both source and derived documents were broken into sentences. Each sentence from the source document was paired with the corresponding sentence of the derived document (i.e., sentence-level cross product of the two documents was obtained). This resulted in 58,406 sentence pairs, of which 8,352 (14.30%) belonged to WD, 28,223 (48.32%) to PD, and 21,831 (37.38%) to ND class. Table 2 shows the distribution of the initial sentence pairs‘ pool with respect to their classes in the source COUNTER corpus.

To compute the string level similarity between the two sentences, the word-based Levenshtein Edit Distance (LED)(Levenshtein 1966) was used. Levenshtein Edit Distance is a textual similarity metric that examines the two words and returns a numerical value indicating their distance based on characters. Similarly, word-based

LED (a variant of the original LED) compares the two sentences and provides a numerical value that shows how far apart they are from one another. In word-based LED, we can think of a sentence as a string (word) drawn from the English alphabet, where each character is a word (assuming that spaces mark the start and end of a character).

To ensure at least minimum divergence among the sentences, and to narrow down the initial candidate pairs space (58,406 pairs) for subsequent human evaluation, three heuristic[10] rules were applied (Dolan *et al.* 2005). The three rules are based on the common lexical properties and sentence positions in the document, are as follows:

- Rule 1 — The word-based LED of the two sentences, must be in the range $1 < \text{LED} < 20$, and the character-based length ratio between the two sentences must be greater than 66%. In addition, the first 3 sentences of the source and derived documents of each pair are also to be included in the candidate pair space, regardless of the sentences' LED or length ratio.
- Rule 2 — In a sentence pair, the length n (number of words) of each sentence must be in the range $5 \leq n \leq 40$. In other words, very short sentences (of length less than 5 words) and very long sentences (of length greater than 40 words) must be excluded.
- Rule 3 — The two sentences must share at least 3 words in common.

Rule 1 is based on string similarity computation and a heuristic for journalism. For the string similarity measurement, the source and derived sentences' word-based LED and character-based sentence length ratios were used as features. LED was calculated using the minimum edit distance (insertion, deletion, substitutions). Both sentences were split into words aka tokens, and dynamic programming was used to select a path with the minimum edit distance at each step to convert a sentence into another (Levenshtein 1966). To ensure that there was no identical sentence pair in the resultant corpus, we used $LED >= 1$. Further, to rule out sentence pairs in which one was too long and the other one too short, the character-based length ratio of both the source and the derived sentences was calculated. Only the sentence pairs with less than 50% length difference were selected.

Another common practice in journalism, namely summarizing the whole article in two or three opening sentences was exploited (Dolan *et al.* 2005) during the creation of SUSPC. Journalists use the "inverted pyramid" structure to write news pieces. The most important information is placed at the top of the inverted pyramid, and the least important information is placed at the bottom [11],[12]. Moreover, journalists also try to give the summarized information of the article in the first couple of sentences. Therefore, if both the source and derived documents are paraphrased,

[10] Heuristic is any approach for problem-solving that utilizes a viable strategy that's not ensured to be optimal but is sufficient for reaching a quick, short-term solution or conclusion (Simon *et al.* 1958).
[11] https://www.theguardian.com/books/2008/sep/25/writing.journalism.news
[12] https://writingcenter.gmu.edu/guides/news-writing-fundamentals

then the first couple of sentences from both articles would also most likely be paraphrased (Dolan *et al.* 2004).

As per Rule 2, sentences that are very short (of length less than 5 words[13]) and very long (of length greater than 40 words[14]) are excluded on the basis of sentence length. Lastly, Rule 3 depends on the number of shared words between the two sentences.

Figure 1 summarizes the process of semi-automatic extraction of the paraphrased sentence pairs. The process started by breaking up each document into sentences and then taking the sentence-level cross product of each document pair. This resulted in 58,406 sentence pairs. Then, Rule 1 was applied to filter sentence pairs based on string similarity and a heuristic rule, which resulted in 13,690 sentence pairs. These pairs were given as input to Rule 2 to exclude very short and very long sentences, which further reduced the number of sentence pairs to 10,679. Finally, Rule 3 was used to ensure that there were at least three common words in both the source and the derived sentences, which further reduced the search space to 3,147 sentence pairs. Among these 3,147 pairs, 774 (24.6%) sentence pairs belonged to WD, 1,478 (46.96%) sentence pairs belonged to PD, and the remaining 895 (28.44%) sentence pairs belonged to ND class as per the COUNTER corpus (see Section 3.1.2).

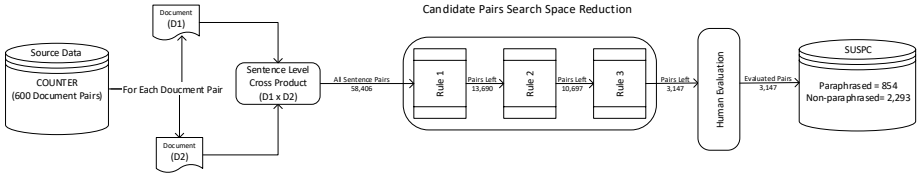


Fig. 1: Corpus Generation Process

3.2 Human Evaluation

The resultant 3,147 semi-automatically extracted and likely to be paraphrased sentence pairs were then required to be examined by human beings.

3.2.1 Evaluation Guidelines

The human evaluation, which followed the semi-automatic extraction of likely paraphrased sentence pairs, would classify the sentences as either paraphrased (P) or non-paraphrased (NP). The evaluation guidelines prepared for the task are described below:

[13] We have to further reduce our selection on the basis of 5 shared words
[14] Maximum sentence length is selected based on experiments performed on the corpora extracted at various sentence lengths.

- Paraphrased (P): If both of the sentences share exactly the same news but with different wording or with minor changes in their text structure, or if they have an addition of related information or are rephrased (see Section 3.1.1) while keeping the semantics of the original text, they are marked as paraphrased (P).
- Non-Paraphrased (NP): If both sentences share the same general topic of the news, but are written in the journalist's own words and use her own findings, or if the two sentences have too few words in common, the pair is tagged as non-paraphrased (NP).

3.2.2 Evaluation Process and Inter-Annotator Agreement

The proposed SUSPC was manually evaluated by three judges (A, B, and C) over the course of one month. All three judges were Urdu speakers with a good understanding of the paraphrase detection task. They were graduate-level students of Computer Sciences at the Information Technology University [15] and had prior experience in text tagging process. The judges were asked to tag a sentence pair into one of the two classes, i.e., paraphrased or non-paraphrased. The complete evaluation was carried out in three phases: (i) training phase, (ii) annotations, and (iii) conflict resolution.

At the start, two judges (A and B) were given 30 randomly selected sentence pairs to tag. They were provided with the evaluation guidelines (see Section 3.2.1) and were trained. The training included lectures on rewriting operations and paraphrasing practices used by journalists, newspapers reading sessions, etc. After this process, a comprehensive meeting with both judges was organized to discuss the problems faced during the tagging process and to resolve the conflicting pairs. The results of these 30 sentence pairs were saved and both judges were asked to evaluate the remaining 3,117 sentence pairs independently.

For all 3,147 sentence pairs, both of the judges agreed with 2,718 sentence pairs. In order to measure the degree of clarity and the judges' comprehension of the annotation guidelines and the uniformity between annotators' judgment, an inter-annotator agreement was computed. This was found to be 86.37%. Moreover, to further measure the reliability of the annotators to classify the sentence pairs, we used the Cohens Kappa Coefficient (k) (Cohen 1960), which is a more robust measure than the accuracy and simple harmonic mean(F_1), because the latter two measures do not consider the hypothetical probability of chance agreements. The value of k for SUSPC is 65.95%, which shows that the reliability of the agreement between the two annotators is substantial.

The rest of the 429 conflicts were resolved by a third judge (C) with a similar skill set. The most prominent conflict found was the amount of text used from the original sentence to generate a paraphrased sentence. We found that 854 (27.14 %) out of 3,147 filtered sentence pairs were paraphrased. For test time evaluations,

[15] https://itu.edu.pk/

we have labelled the paraphrased class as true negative as described in details in section 5.3.

In order to check the likelihood of plagiarised sentence pairs being rejected by the semi-automatic sentence pairs extraction approach, we selected 300 sentence pairs from the rejected pool of 55,559 sentence pairs at random. Out of the 300 rejected sentence pairs, only 8 were found to be plagiarized. This implies that only 2.66% of paraphrased sentence pairs were missed by the semi-automatic sentence pairs extraction approach. Table 3 shows the annotation statistics for the selected and rejected pools of sentence pairs.

Table 3: SUSPC Human Evaluation Statistics

	Selected Pool	Rejected Pool
Total Sentence Pairs	3,147	55,559
Selected Sentence Pairs for Annotation	3,147	300
Sentence Pairs for Initial Annotation	30	0
Remaining Sentence Pairs for Annotation	3,117	300
Agreed	2,718	294
Conflicted	429	06
Inter-annotator Agreement	86.37 %	98 %
Kappa Coefficient (k)	65.95 %	56.14 %
Paraphrased Sentence Pairs	854	08
Non-Paraphrased Sentence Pairs	2,293	292

3.3 Corpus Statistics

SUSPC contains 3,147 sentence pairs, 131,513 words (tokens), and 8,033 unique words (types). More than half of the pairs belong to the non-paraphrased class (2,293 sentence pairs, 96,057 tokens, and 7,135 types) while the rest are paraphrased (854 sentence pairs, 35,456 tokens, and 4,229 types). Table 4 shows the statistics of the proposed corpus.

3.4 Distribution of sentence pairs

Table 5 presents a distribution of the resultant annotated sentence pairs in SUSPC plotted against the sentence pairs‘ initial classes in the source corpus, i.e., COUNTER (Section 3.1.2). It can be seen that the SUSPC corpus contains 3,147 sentence pairs, of which 854 are paraphrased while the rest (2,293) are non-paraphrased. The paraphrased class comprises 387 WD, 370 PD, and 97 ND sentence pairs, whereas the non-paraphrased class includes 387 WD, 1108 PD, and 798 ND sentence pairs.

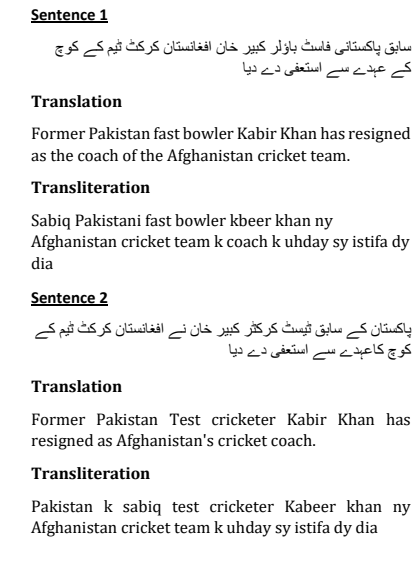
Table 4: SUSPC Statistics

Total Sentence Pairs	3,147	
Paraphrased Sentence Pairs	854 (27.14 %)	
Non-Paraphrased Sentence Pairs	2,293 (72.86 %)	
	Source	**Derived**
Total Tokens	66,494	65,019
Total Types	4,778	6,472
Min Tokens Per Example	6	5
Max Tokens Per Example	40	40

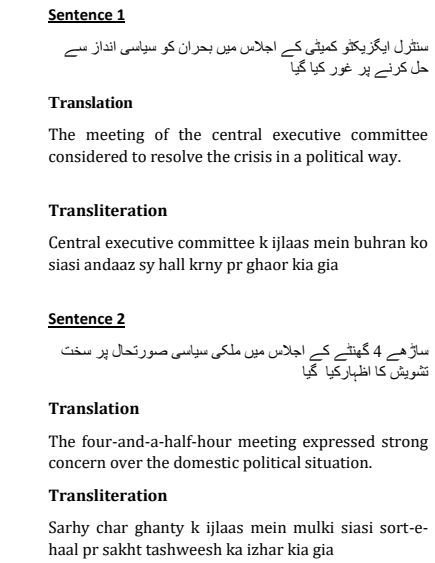
It can also be observed from Table 5 that the selected sentence pairs‘ pool (i.e., 3,147 sentence pairs) is 5.39% of the initial sentence pairs‘ pool (i.e., 58,406 sentence pairs). This leads to the development of a hypothesis that the size of selected sentence pairs‘ pool (i.e., SUSPC) has a relation of direct proportionality with the initially developed clusters and their size. Moreover, it can also be observed that the pattern of the classification of selected pairs to their respective classes has a relative proportion with the pattern of initial sentence pairs‘ distribution (see Table 2). As can be seen, the PD sentence pairs' class was the dominant class in the initial distribution (i.e., 48.32%), which is also the trend that can be seen in the resultant sentence pairs' pool (i.e., 46.96% came from PD). Similarly, the contribution of sentence pairs of the ND class 37.38% in the initial and filtered sentence pairs distributions was and 28.44%, respectively. Finally, the WD category followed the trend with 14.30% in the initial sentence pairs pool and 24.60% in the resultant sentence pairs pool.

Table 5: Distribution of the semi-automatically generated sentence pairs in SUSPC w.r.t. COUNTER corpus

		COUNTER Classes			**Total**
		Wholly Derived	**Partially Derived**	**Non Derived**	
SUSPC Classes	**Paraphrased**	387	397	70	854
	Non-paraphrased	387	1108	798	2293
Total		774	1478	895	3147



(a) Example sentence pair annotated as *paraphrased*



(b) Example sentence pair annotated as *non-paraphrased*

Fig. 2: Example sentence pairs from SUSPC (a) *paraphrased*, (b) *non-paraphrased*

3.5 Examples from the Corpus

Figure 2a shows a semi-automatically extracted sentence pair, manually tagged as paraphrased. The length of both sentences is quite similar, hence satisfying Rule 1. The word level edit distance between the two sentences is 8 — meaning that the news editor made 8 word edits in the original news text — thus also satisfying Rule 2. Finally, both sentences share 13 words in common, thus satisfying Rule 3 as well. Consequently, the semi-automatic approach picked this sentence pair to be included in the initial data. Later on, the pair was tagged as paraphrased by the human judges because the two sentences shared the same news event, with slightly different wording and minor changes in the sentence structure, thus meeting the criteria for being marked as paraphrased as per the annotation guidelines (see Section 3.2.1).

Figure 2b shows a sentence pair that is tagged as non-paraphrased. This pair also satisfies the three heuristic rules to be included in the initial data of selected pairs. The sentences' length ratio is greater than 66%, there are 11 word level edits to convert the first sentence into the second, and they share 5 words in common. The judges marked this sentence pair as non-paraphrased because the information in both texts is not the same.

3.6 Corpus Limitations

Although the proposed SUSPC corpus is the first-ever semi-automatically sentence-level paraphrased corpus in Urdu, it has some significant limitations:

- The size of the proposed SUSPC corpus (3,147 sentence pairs) is only a little larger than the size of the manually created USTRC corpus (2,680 sentence pairs, see Section 2.1) corpus. The main reason could be the limited size of clusters in the source news data, as the average numbers of sentences in each cluster are nine and eight for the source and derived documents, respectively. Increasing the size of the source and derived documents could enhance the size of the SUSPC corpus. In other words, if the number of documents in the source corpus increased, the number of sentence pairs in the resultant corpus would also increase.
- Another limitation of the SUSPC corpus is the imbalance between the assigned classes: where the number of paraphrased sentence pairs is only 854 (27%), the number of non-paraphrased sentence pairs is 2,293 (72.26%). The lower count of the paraphrased class is entirely plausible because ND documents are independently written and have a lot of new text, as per the annotation guidelines of the COUNTER corpus (Sharjeel *et al.* 2017). Thus, the ND documents are bigger than the WD and PD documents. In addition, the strict filters in place for the reduction of the candidate pairs' search space (see Section 3.1.3) could also have significantly reduced the number of paraphrased sentence pairs. For example, the filter $LED \geq 1$ ensured that neither verbatim nor almost identical cases were included in SUSPC.
- The proposed corpus SUSPC contains only text examples from journalism. In the future, text examples from other fields, such as academics, Urdu literature, history, etc., can be added to increase the size of the SUSPC corpus.
- Similarly, the vocabulary of SUSPC is limited to only five domains national, foreign, business, sports, and showbiz), which can be expanded to further domains like health, education, current affairs, politics, etc.

4 Paraphrase Detection Approaches

This section presents the two proposed DNN-based approaches to detect sentence-level paraphrasing in Urdu texts: (i) Word Embeddings *n-gram* Overlap (WENGO), and (ii) Deep Text Reuse and Paraphrase Plagiarism Detection (D-TRAPPD).

4.1 WENGO Approach

WENGO is inspired by the popular but simple Word *n-gram* overlap approach (Alzahrani *et al.* 2011), which is used to detect paraphrasing between two texts. In lieu of words, we used their respective pre-trained word embedding vectors (using FastText pre-trained word embeddings model, see Section 5.2). These vectors represent a word in a 300-dimensional vector space with a capability to capture semantic and syntactic properties of the text (Mikolov *et al.* 2013). These embedding

vectors are the learned representations of words from a text where semantically similar words have similar representations (Goldberg Graeme 2017). For instance, the words "car" and "bus" are semantically similar, and their embedding vectors will have almost the same representation. These embedding vectors have shown impressive performance in a variety of NLP's challenging problems like Sentiment Analysis (Yu *et al.* 2017), Machine Translation (Klein *et al.* 2017), Information Retrieval (Vulic *et al.* 2015; Ganguly *et al.* 2015), and Semantic Textual Similarity (Kenter Maarten 2015) detection.

In order to detect the paraphrasing between sentence pairs, we used uni-gram, bi-gram, tri-gram, and four-gram word embeddings overlap for each sentence in a sentence pair. These embedding vectors were added together to make an average vector (see equation 1) for all words in a particular *n-gram*. The average vectors were concatenated together to make an average embedding matrix for each sentence in a sentence pair. Suppose, we have a sentence s that contains d words $\{w_1,w_2,w_3,...,w_d\}$, whose respective embedding vectors are $\{v_1,v_2,v_3,...,v_d\}$, there will be a total of $(d-n+1)$ *n-gram* tuples, where n is the length of an *n-gram* tuple. For example, let's consider a sentence s that contains 5 words (d=5). If we make tri-gram (n=3) tuples of s then there will be 3 tri-gram tuples. For each word in each tuple, the word embedding vector was extracted from a pre-trained word embedding model. The average vector of all words in each tuple was generated by taking the average of all 3 embedding vectors (see equation 1). Finally, an average embedding matrix (M) was generated for each sentence s in a given sentence pair, and the *cosine similarity score* was computed for the two average embedding matrices for each text pair using equation 2.

$$EV_t = \frac{1}{n}\sum_{i=1}^{n} v_i \tag{1}$$

$$Cos(M_{s1}, M_{s2}) = \frac{\vec{M_{s1}} \cdot \vec{M_{s2}}}{|\vec{M_{s1}}||\vec{M_{s2}}|} \tag{2}$$

In equation 1, EV_t is the *average embedding vector* of an *n-gram* tuple t, and n is the length of the tuple i.e. uni-gram, bi-gram, tri-gram , or four-gram. In equation 2, M_{s1} and M_{s2} are the average embedding matrices of sentences s_1 and s_2, respectively, constructed from the average embedding vectors (see equation 1) of each *n-gram* tuple. These matrices are converted into a single flattened vector to compute their cosine similarity score.

The WENGO approach is computationally inexpensive. However, it is based on a bag-of-words model which causes it to lose the order of the n-grams, hence deteriorating the underlying text semantics. Secondly, it gives high weights to unrelated words when taking the average of all embedding vectors in an n-gram tuple making it difficult to differentiate among the word embedding vectors. Therefore, in this work, we use DNN-based approaches (i.e. CNNs, LSTMs). In most cases, these DNN-based approaches achieve state-of- the-art results compared to the WENGO approach. Particularly in text classification problems, these DNN-based approaches

perform better than the traditional linear classifiers especially when working with pre-trained word embedding representations (Zhang et al. 2015; Goldberg 2016).

4.2 D-TRAPPD Approach

Besides WENGO, we proposed another DNN-based approach D-TRAPPD for the task of Urdu paraphrase detection in monolingual settings. It is based on the work of Agarwal et al. (Agarwal *et al.* 2018) and consists of two major modules: (i) CNN and (ii) LSTM.

The CNN module is responsible for the extraction of meaningful and salient structures from the text, which is represented using word embedding vectors (Goldberg Graeme 2017). It is also noteworthy that certain word sequences are good indicators of the underlying semantics or topic of the text irrespective of their position (Goldberg 2016). In CNN, the convolutional layers in combination with the pooling layers are able to extract strong local features of words/phrases regardless of their position in the input text (Goldberg 2016). Although CNN is an important feature extraction neural network architecture, it shows better performance when integrated with a large neural network(Goldberg Graeme 2017). Therefore, we used it in combination with LSTMs, which are capable of learning long term dependencies and are specifically designed to learn the temporal ordering of long input sequences (Hochreiter *et al.* 1997), which is exactly what we require for the problem under discussion.

Figure 3 shows the high-level architecture of our proposed D-TRAPPD approach for paraphrase detection in Urdu short text pairs. Firstly, Urdu word embedding vectors were extracted from a pre-trained word embedding model (FastText (Grave *et al.* 2018); see Section 5.2) for both input sentences (s_1 and s_2) in a sentence pair to get a distributional vector representation matrix for each sentence. Secondly, these embedding matrices were provided as inputs to a Siamese CNN module (two replicas of the same CNN working on two different input vectors to produce comparable output vectors) (Chicco 2021) to meaningful and salient structures from the text. The word-wise convolutions were performed on the input embedding matrices using kernels of two different sizes, i.e., 2 (called bi-gram) and 3 (called tri-gram), 128 filters, and stride size 1 for all spatial dimensions. The convolutional layers were activated by ReLU (Rectified Linear Unit (Nair Geoffrey 2010)) followed by a *dropout* layer (Srivastava *et al.* 2014) to prevent the network from over-fitting. To summarize the resulting feature map, a max-pooling layer was added to the module. These condensed feature maps of both sizes (bi-gram and tri-gram) were concatenated to prepare the input for the next LSTM module with 64-dimensional output space and L_2 kernel regularizer.

The element-wise difference of the LSTM's output vectors (for both sentences s_1 and s_2) was taken using a *lambda* layer. The subsequent difference vector was the separating representative vector of the sentence pair that was utilized as a feature vector for learning the similarity between the two texts. It was used to classify the given sentence pair at the output layer, using two fully connected layers followed by their respective dropouts (Kingma *et al.* 2014) to regularize the proposed neu-

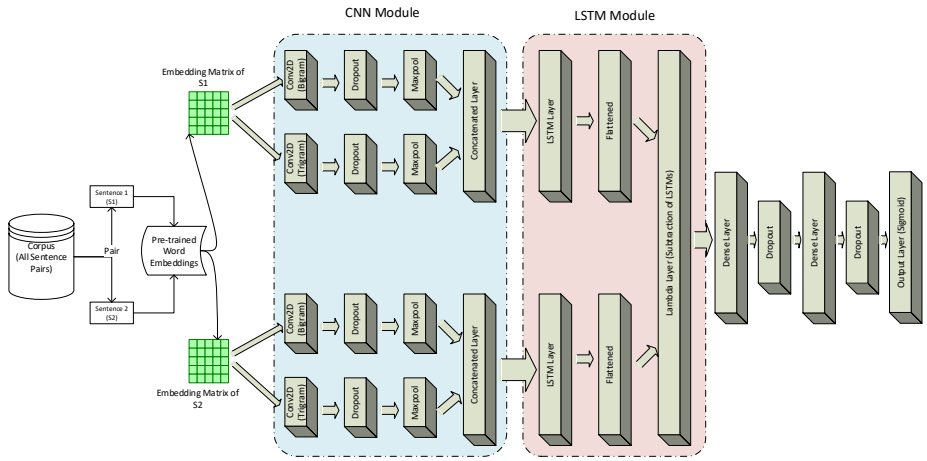


Fig. 3: Proposed Deep Neural Network Architecture for Paraphrased Text Reused Detection.

ral network. At the output layer, Sigmoid (Han *et al.* 1995) activation was used to perform binary classification (i.e. into paraphrased or non-paraphrased), and Softmax activation function (Goodfellow *et al.* 2016) was utilized for ternary classification (i.e. into verbatim, paraphrased, or non-paraphrased) tasks. Finally, the two separate models were trained for both binary and multi-classification.

5 Experimental Setup

This section describes the experimental setup used along with the evaluation tasks and corpora, the text pre-processing and performance measures used for paraphrase detection and other evaluation tasks in monolingual settings.

5.1 Evaluation Tasks

We evaluated the proposed DNN-based approaches for two tasks: (i) paraphrase detection, and (ii) text reuse and plagiarism detection. The evaluation of multiple tasks allows us to report a fair generalization of the proposed approaches for the detection of textual similarity detection in Urdu texts.

5.1.1 Paraphrase Detection

The paraphrase detection task was aimed at finding whether two sentences were paraphrased, based on their semantic similarity. We have selected UPPC (see Section 2.1) and SUSPC corpora (see Section 1) because both have been developed for the task of paraphrase detection. The task has been studied as a binary classification because text pairs in both corpora are either paraphrased or non-paraphrased.

It is also worthwhile to note here that, for UPPC, we are the pioneers in evaluating the corpus for automatic paraphrased plagiarism detection tasks, particularly using the DNN-based approaches. Therefore, the results can serve as a baseline for future experiments on both corpora for Urdu paraphrase detection.

5.1.2 Text Reuse and Plagiarism Detection

Text reuse is the act of borrowing and using text from a previously published text (i.e. source text). It could occur at the sentence, passage, or document level. The text could be reused verbatim (word-for-word) or paraphrased by changing the word order, exchanging words with appropriate synonyms, compressing or expanding the text, etc. The counterpart of text reuse is plagiarism, which is the unacknowledged reuse of text.

For the text reuse and plagiarism detection task, we selected the USTRC (see Section 2.1) corpus. USTRC has been developed for the detection of text reuse and plagiarism, and considers three types of text reuse and plagiarism cases (i.e. verbatim, paraphrased, and independently-written/non-paraphrased). We conducted our study for both binary classification and multi-classification tasks. For binary classification, we used USTRC by merging *verbatim* and *paraphrased* classes to make a single class called *paraphrased* (i.e. verbatim + paraphrased = paraphrased), while using the *non-paraphrased* class as it was. For multi-classification, we considered all of the three classes (i.e. verbatim, paraphrased, and independently-written/non-paraphrased). As a baseline for text reuse and plagiarism detection, we used the work of Sameen et.al (Sameen *et al.* 2017) on USTRC.

5.2 Text Pre-Processing and Word Embeddings Extraction

The essays (in UPPC) and long passages (in USTRC) were converted into a single sentence by removing all sentence separators, and each sentence was pre-processed by removing all of its numbers, punctuation, more than one white space, line breaks, and all other characters other than Urdu letters (Sharjeel *et al.* 2017; Maaz *et al.* 2020). Furthermore, each sentence was tokenized on a single white space. For each token, its respective word embedding vector was extracted from a pre-trained word embedding model for Urdu (i.e. FastText (Grave *et al.* 2018)).

Although several word embedding models (e.g. Word2Vec (Mikolov *et al.* 2013; Qasmi *et al.* 2020), Glove (Pennington *et al.* 2014), FastText (Grave *et al.* 2018), ELMO (Peters *et al.* 2018), BERT (Devlin *et al.* 2018), RoBERTA (Liu *et al.* 2019)) were available, the largest pre-trained word embedding model available for Urdu (at the time of experimentation) was FastText. FastText are pre-trained distributed word vectors (extracted by using FastText API[16]), trained on Wikipedia and Common Crawl[17] using continuous bag-of-word (BOW) with position-weights, character 5-gram, in 300 dimensions.

[16] https://fasttext.cc/docs/en/crawl-vectors.html
[17] https://commoncrawl.org/

5.3 Evaluation Methodology and Measures

The main objectives of the experiments performed for this study were two-fold. Firstly, we explored whether it is possible to differentiate between the different levels of Urdu paraphrasing using the proposed DNN-based approaches. Secondly, we evaluated whether the proposed DNN-based approaches perform better than the traditional surface-level similarity measurement based approaches for Urdu text reuse and plagiarism detection task in monolingual settings.

To achieve these objectives, we applied various conventional machine learning classifiers to report a comparison between the extant state-of-the-art approaches and our proposed approaches. Thus, we studied the problem as a classification task and used 10 different classifiers: (i) Nearest Neighbors (NN), (ii) Logistic Regression (LR), (iii)Linear Support Vector Machines (LSVM), (iv) SVM with Radial Basis Function (RBF-SVM), (v) Decision Tree (DT), (vi) Random Forest (RF), (vii) Multi-Layer Perceptron (MLP), (Viii) AdaBoost (AB), (ix)Naive Bayes (NB), and (x) Quadratic Discriminant Analysis (QDA).

We used the standard evaluation measures used in previous studies (Sharjeel *et al.* 2017; Sameen *et al.* 2017) for Urdu text reuse and plagiarism detection tasks. These measures are *precision* (see Equation 3), *recall* (see Equation 4), and F_1 *scores* (see Equation 5). In the context of classification, we have defined True Positives (TP) as the relevant text pairs correctly classified as non-paraphrased class and True Negatives (TN) as the text pairs correctly classified as paraphrased.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

The value of precision and recall ranges between 0 and 1, demonstrating the lowest performance and the best performance, respectively. Normally, there is a trade-off between precision and recall, i.e., a high value of precision with high value shows that the system correctly identified all the relevant text pairs, but the corresponding recall will be low. Similarly, a high recall will result in low precision. To balance the effect of precision and recall trade-off, a harmonic mean is computed by combining the precision and recall values, known as *F-measure* (Baeza *et al.* 1999) or F_1*-measure* [18] (see Equation 5). The value of F_1 also varies between 0 (worst) and 1 (perfect). F_1 is generally used for corpora with imbalanced classes.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

[18] Both recall and precision have equal weights.

5.4 Model Hyper-parameters Setting

The proposed approaches were implemented using Keras[19], a renowned deep learning API written in Python. We tried various combinations of the hyper-parameters and retained those with better performance. Since experiments have been performed on three different corpora for two similar tasks (see Section 5.1), most of the hyper-parameter settings were common, such as (i) *dropout rate*(Srivastava *et al.* 2014) (0.5), (ii) *optimizer* (adam (Kingma *et al.* 2014)), (iii) *kernel regularizer* (L_2), (iv) *learning rate* (1.0), and (v) *epochs* (10,25,50,125,250,500,1000,2000). Similarly, various sequence lengths were tried for each of the three corpora and the lengths with the best results were selected. For sentence-level corpora (USTRC, SUSPC), the appropriate sequence length is 30 words, while for document-level corpora (UPPC), a sequence length of 250 words performed better. For sequences shorter than the maximum length, zero-padding was used.

Different *batch sizes* were used for the USTRC and SUSPC corpora (64, 128, 256, and 512) on the one hand, and the UPPC corpus (8,16,32,64, and 128) on the other. Smaller batch sizes were used for UPPC since it has a smaller number of paraphrased plagiarism cases. For the binary classification task, "binary cross-entropy" was considered as *loss*, whereas for the multi-classification task, "categorical cross-entropy" was considered as *loss*.

To prevent our trained model from over-fitting and to perform an effective unbiased evaluation, we used stratified 10-fold cross-validation along with *validation split* (0.1). At first, input data was randomly divided into ten equal subsets where nine subsets were used for training and the remaining one subset was used for testing. Further, 10% of the training dataset was used to make a validation dataset for an unbiased evaluation of the model fitted on the training dataset and to tune the model hyper-parameters. Finally, the trained model was tested on the test dataset to assess how well our model is generalized and how well it performs in the production environment.

6 Results and Discussion

This section discusses the results of the two evaluation tasks described in Section 5.1.

6.1 Paraphrase Detection Task Results

6.1.1 Results and Discussion on UPPC

Table 6 shows the results (only the best ones of each approach) of the paraphrase detection task for UPPC. The "Approach" column lists all the approaches applied for the paraphrase detection task, while "Classifier" logs the Machine Learning (ML) algorithms that produced the best F_1 score for binary classification. It is

[19] https://keras.io/

important to note that the following tables list only those classifiers that show the highest F_1 scores together with their respective precision and recall values.

Table 6: Result Comparison of the proposed approaches on UPPC

	Binary Classification			
Approach	Precision	Recall	F_1	Classifier
WENGO uni-gram	78.25	76.3	76.45	NB
WENGO bi-gram	83.88	81.28	81.55	MLP
WENGO tri-gram	83.43	81.38	81.64	QDA
WENGO four-gram	83.43	81.38	81.64	MLP
D-TRAPPD	**80.54**	**89.56**	**84.74**	**DNN**

Overall, the proposed D-TRAPPD approach performed better than all other approaches ($F_1 = 84.74$), whereas among the WENGO-based approaches, WENGO tri-gram and four-gram (using QDA and MLP classifiers, respectively) produced the highest F_1 scores. It can be noted that the performance of WENGO-based approaches increased with an increasing value of n. One possible reason could be that the lengths of the source and the rephrased texts (i.e. between 200 to 300 words) increased the number of *n-grams* containing semantically similar words in the source and paraphrased sentences. Another reason could be that the simulated text generation process of UPPC as students were permitted to look into the provided material — increased the chances of several phrases being simply copied and pasted from the helping material, resulting in more common or semantically similar words in the *n-gram* tuples.

6.1.2 Results and Discussion on SUSPC

Table 7 demonstrates the results of the experiments performed on SUSPC (see Section 3). These are only the best results from the application of each approach. Overall, our proposed D-TRAPPD approach outperformed ($F_1 = 96.80$) all other approaches applied for the binary classification task on SUSPC.

Among the rest of the approaches, the WENGO four-gram approach produced the highest scores ($F_1 = 57.28$). A clear pattern can be observed among F_1 scores of the WENGO-based approaches, i.e., the F_1 score keeps on increasing as the value of n increases. The heuristic filters (see Section 3.1), particularly Rule 3 ("both sentences must have at least 3 words in common"), could be the possible reason for the observed pattern in the output. Since words are represented by their respective embedding vectors, semantically equivalent vectors have similar representation vectors.

Table 7: Result Comparison of the proposed approaches on SUSPC

	Binary Classification			
approach	Precision	Recall	F_1	Classifier
WENGO uni-gram	56.16	54.79	54.80	NN
WENGO bi-gram	70.16	56.93	56.17	NB
WENGO tri-gram	58.24	56.59	56.88	NN
WENGO four-gram	73.44	57.77	57.28	QDA
D-TRAPPD	**96.91**	**96.68**	**96.80**	**DNN**

6.2 Text Reuse and Plagiarism Detection Task Results

6.2.1 Results and Discussion on USTRC

Table 8 shows the results for both binary and multi-classification for text reuse and plagiarism detection task on USTRC. It is important to note that Table 8 only lists those classifiers that show the highest F_1 scores together with their respective *precision* and *recall* scores.

The highest F_1 scores for both binary and ternary classification tasks are produced by the proposed D-TRAPPD approach. Among these, the results for the binary classification task are slightly lower ($F_1 = 87.85$) than the ternary classification ($F_1 = 88.90$). A clear reason is the difference between the precision and recall scores, which are greater in the case of binary classification than that of the multi-classification task. In binary classification, the recall score (83.64) is quite low as compared to the recall score (87.98) of multi-classification. This shows that the data in *verbatim* and *paraphrased* classes do not belong to the same distribution, as can be seen in the ternary classification models. In other words, it shows substantial difference between the means of all three classes, particularly for *verbatim* and *paraphrased* classes. Moreover, in binary classification we are forcing the model to classify sentence pairs into two classes. This confuses the model with respect to the first class (Verbatim+Paraphrased), and leads to some of its instances being turned to the other class, resulting in a high number of false negatives and a dropp in the model's recall score. This pattern can also be observed in the results of the WENGO bi-gram, tri-gram, and four-gram approaches. Only in the case of the WENGO uni-gram approach, the binary classification is seen to be easier than the multi-classification task.

For the WENGO-based approaches, the highest scores ($F_1 = 61.08$) for binary classification are produced by the WENGO uni-gram approach, whereas for multi-classification, the WENGO four-gram ($F_1 = 61.19$) approach performed better than all other WENGO-based approaches. Since we cannot observe a pattern in performance variation of WENGO-based approaches when the length of n is increased, we can conclude from the competitive results that the variation in length of n is not a

Table 8: Result Comparison of the proposed approaches on USTRC

	Binary Classification				Multi Classification			
approach	Precision	Recall	F_1	Classifier	Precision	Recall	F_1	Classifier
WENGO uni-gram	66.69	60.66	61.08	NB	67.63	56.58	59.38	AdaBoost
WENGO bi-gram	59.97	61.43	58.12	QDA	63.04	57.44	58.83	NB
WENGO tri-gram	59.37	60.44	59.32	QDA	64.93	58.77	60.13	NB
WENGO four-gram	60.11	60.38	60.23	NB	65.95	60.02	61.19	QDA
D-TRAPPD	**92.52**	**83.64**	**87.85**	**DNN**	**89.84**	**87.98**	**88.90**	**DNN**

good discriminator to detect text reuse of the source text. Overall, WENGO-based approaches by themselves are not suitable for text reuse and plagiarism detection for the corpus in Urdu of short text pairs derived from real news(i.e. USTRC). However, when these word embeddings are used with some neural network architectures, such as CNN or LSTM (like in the proposed D-TRAPPD approach), they produce better results than the state-of-the-art approaches.

6.3 Best Results Comparison for Both Tasks

Table 9 shows the best results for both the tasks of paraphrase detection (see Section 5.1.1) and text reuse and plagiarism detection (see Section 5.1.2) tasks. It can be seen that the proposed D-TRAPPD approach (see Section 4.2) outperformed all the other approaches applied on the two tasks for both binary classification and multi-classification (where applicable). The apparent reason for this significant difference in results is the application of additional layers (over the pre-trained embedding models) of the complex and computationally expensive deep neural architectures (CNNs and LSTMs). Looking at the two proposed approaches, we find that the WENGO-based approaches only considered the pre-trained embedding vectors in an n-gram overlapping fashion. In comparison, CNNs, on which the D-TRAPPD approach is based, considered not only the pre-trained embedding vectors, but also combined the convolutional layers with the 765 pooling layers to extract the robust local features of the sentence pairs. It did this regardless of the positioning of the words in the input text (see Section 4.2). Moreover, CNNs, in combination with LSTM, also captured the long-term dependencies, specifically the temporal ordering of long input sequences, which was not possible under the WENGO-based approaches.

These results (Table 9) suggest that for the detection of text reuse and paraphrased plagiarism in Urdu, it is much better to use the D-TRAPPD approaches than the pre-trained embedding models that only consider embedding vectors. Moreover, it can also be clearly observed from the best results of all the corpora that the proposed approach produced lower scores on UPPC (simulated cases of text reuse and plagiarism) than it did on SUSPC (semi-automatically generated

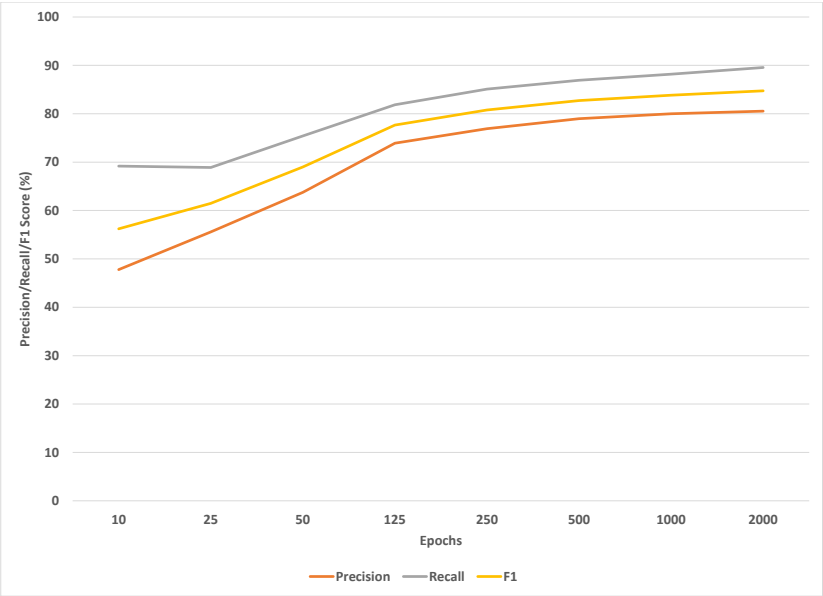
cases of paraphrasing). This implies that simulated cases of paraphrased plagiarism are more difficult to detect than both real cases and semi-automatically generated cases of text reuse and paraphrasing.

It can also be observed that the scores of the proposed D-TRAPPD approach on both USTRC and UPPC are comparable, whereas the scores of D-TRAPPD on SUSPC are quite high. The possible reason of similarity in the former two could be the slight commonalities between the paraphrased text pairs‘ generation process and the annotation guidelines of the two corpora. These commonalities can be explained by the fact that USTRC contains texts from the journalism domain and it is a common practice among journalists to obfuscate the text of new reports using different rewriting operations like synonym replacement, changing word order etc.(Bell 1991; Fries 1997; Jing *et al.* 1999). Similarly, during the simulated texts generation process in UPPC, students were allowed to look at the provided material, which increased the likelihood of their simply copy-pasting phrases from the helping material, resulting in common or semantically similar words. On the other hand, the higher scores observed in the SUSPC output may be due to the rules that were followed in the text pairs' generation process that ensured that there were at least three words in common among the members of each pair. Since words are represented by their respective embedding vectors, semantically equivalent vectors have similar representation vectors.

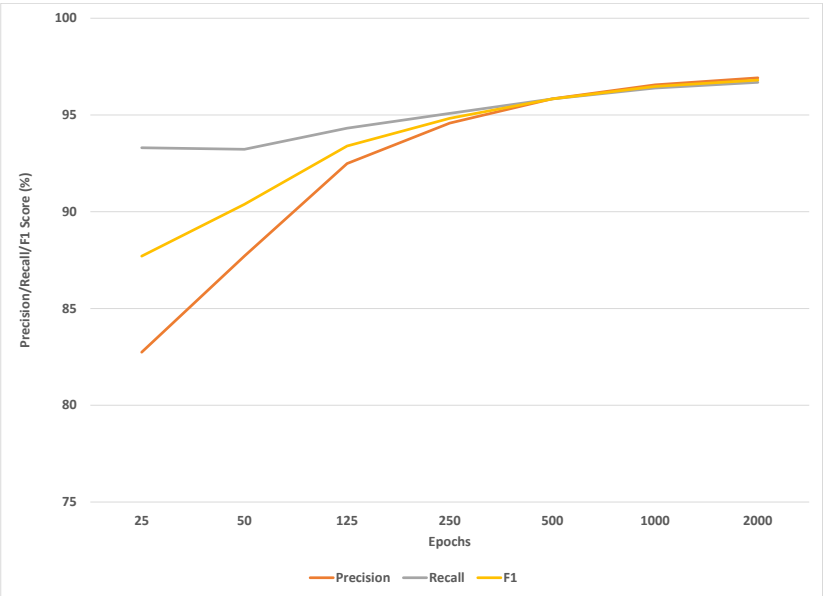
Figure 4 and Figure 5 show the effects of the different number of epochs (see Section 5.4) used in D-TRAPPD for both tasks (paraphrase detection and text reuse and plagiarism detection) on all three corpora (UPPC, SUSPC, and USTRC) for both binary and multi-classification (where applicable). Overall, for both evaluation tasks (see Section 5.1), the scores of all of the three evaluation measures (i.e. precision, recall, and F_1; see Section 5.3) increased as the number of epochs increased. For the paraphrase detection task on UPPC (see Figure 4a), a significant increase ($F_1 = 56.21$ to $F_1 = 84.74$) can be observed in the scores with the increments in epochs, whereas on SUSPC (see Figure 4b), this increase ($F_1 = 84.91$ to $F_1 = 96.80$) is not too striking. This leads to the conclusion that the textual or semantic variation in simulated paraphrased cases is higher than in the semi-automatically generated cases. For the text reuse and plagiarism detection task, a noteworthy improvement in F_1 scores ($F_1 = 17.74$ to $F_1 = 87.85$) for binary classification (see Figure 5a) on USTRC can be observed. Finally,for multi-classification (see Figure 5b), this increase ($F_1 = 34.66$ to $F_1 = 88.90$) is obvious, yet not as remarkable as that of the binary classification task. This shows that, with increasing epochs, the proposed DNN-based models can easily differentiate between the three classes of text reuse as compared to the two classes of USTRC.

6.4 Comparing Results with the Baseline Approaches

For the paraphrase detection task (see Section 5.1.1), there are no baseline approaches and reported results in the past. This implies that we are the first ones to apply any type of paraphrase detection approach(es) on the UPPC. The reported results from both UPPC and SUSPC can serve as a baseline for the task of para-

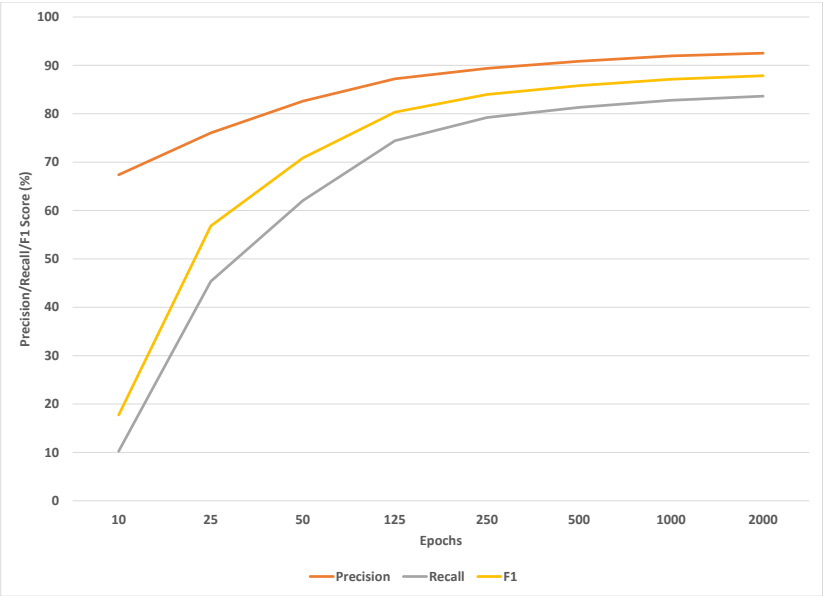


(a) Effect of epochs on binary classification for UPPC

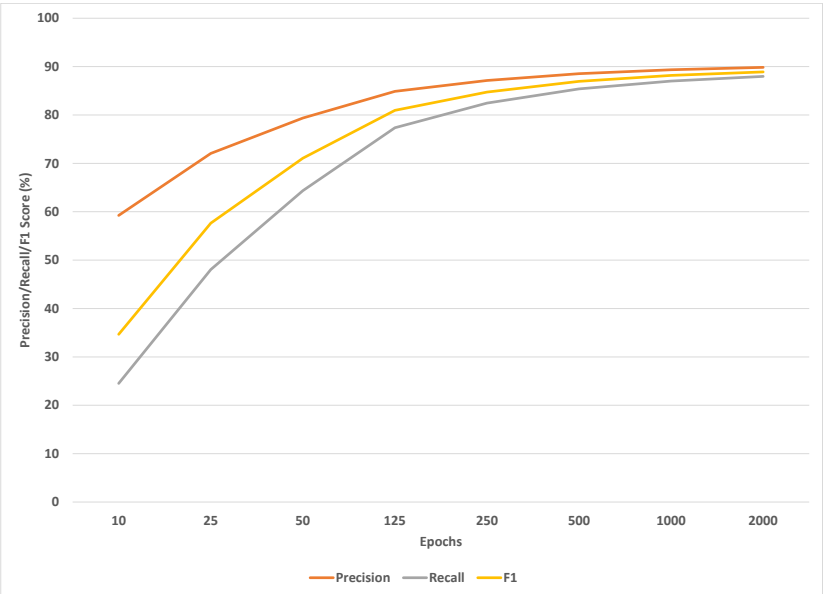


(b) Effect of epochs on binary classification for SUSPC

Fig. 4: Effect of epochs on *precision, recall, and* F_1 for Paraphrase Detection Task (section 5.1.1) on: (a) UPPC and (b) SUSPC.



(a) Effect of epochs on binary classification for USTRC



(b) Effect of epochs on multi-classification for USTRC

Fig. 5: Effect of epochs on *precision, recall, and* F_1 for Text Reuse and Plagiarism Detection Task (section 5.1.2) on USTRC for: (c) binary classification, and (d) multi classification.

Table 9: Comparison of the Best Results for: (i) Paraphrase Detection Task, and (ii) Text Reuse and Plagiarism Detection Task

		Binary Classification				Multi Classification			
Task	Corpus	Precision	Recall	F_1	Classifier	Precision	Recall	F_1	Classifier
Text Reuse and Plagiarism Detection	USTRC	92.52	83.64	87.85	DNN	89.84	87.98	88.90	DNN
Paraphrase Detection	UPPC	80.54	89.56	84.74	DNN				
Paraphrase Detection	SUSPC	96.91	96.68	96.80	DNN				

phrase detection in Urdu. In contrast, for the task of text reuse and plagiarism detection (see Section 5.1.2), we considered the works of Sameen et.al (Sameen *et al.* 2017), including USTRC, as the the baseline and state-of-the-art approach for text reuse and plagiarism detection for Urdu in monolingual settings.

Table 10 presents the comparison of the proposed D-TRAPPD approach for text reuse and plagiarism detection on USTRC with the state-of-the-art approach (Sameen *et al.* 2017) for both binary and multi-classification tasks. A "blank field" shows that the author(s) did not report this evaluation measure. The "Structural (Baseline)" approach refers to the character n-gram overlap (CNG) approach to measure the similarity between two texts. Sameen et.al (Sameen *et al.* 2017) achieved the best results for binary classification ($F_1 = 77.50$ using character 5-gram, character 6-gram, and J48 classifier) and multi-classification ($F_1 = 70.40$ using character 3-gram, and J48 classifier) tasks. Our proposed D-TRAPPD approach outperformed the state-of-the-art approach in both binary classification ($F_1 = 87.85$) and multi-classification (($F_1 = 88.90$)) tasks. In addition, our results also included the precision and recall measures to provide a better insight into the results.

This comparison clearly demonstrates that the proposed D-TRAPPD approach performed better than the structural approaches on the same corpus (i.e. USTRC), particularly for the multi-classification task, which is harder to detect. Moreover, the baseline approaches achieved higher results for binary classification than the multi-classification task, whereas the proposed D-TRAPPD approach obtained better results for multi-classification than the binary classification task. This implies that the proposed semantic based D-TRAPPD approach has a deeper understanding of the distribution of classes as compared to the surface-level structural approaches.

Table 10: Comparison with the state-of-the-art approaches on Urdu Short Text Reuse Corpus

		Binary Classification				Multi Classification			
Task	Approach	Precision	Recall	F_1	Classifier	Precision	Recall	F_1	Classifier
Text Reuse and Plagiarism Detection	Structural (Baseline)			77.50	J48			70.4	J48
Text Reuse and Plagiarism Detection	D-TRAPPD	92.52	83.64	87.85	DNN	89.84	87.98	88.90	DNN

7 Conclusion

This research work focuses on answering the following research questions: (i) how to create a semi- or fully automatically generated corpus for paraphrase detection in Urdu; (ii) whether it is possible to differentiate between different levels of Urdu paraphrasing using the mainstream DNN-based approaches; and (iii) whether the DNN-based approaches perform better than the traditional approaches that measure surface-level similarity between two sentences for Urdu paraphrase detection and text reuse and plagiarism detection.

The first question has been answered by presenting the first-ever semi-automatically generated sentence-level corpus to develop and evaluate Urdu paraphrased detection systems (see Section 3). This corpus was developed in the footsteps of MRPC by using standard procedures, annotation guidelines, and XML encoding format. This corpus has also been made publicly available to foster research and development in Urdu paraphrased detection and text reuse and plagiarism detection. However, the proposed corpus has some limitations including its limited size in terms of domain coverage, vocabulary, number of sentence pairs, etc. In the future, we will explore the recently published paraphrased text generation approaches to increase the size of the proposed SUSPC corpus.

The second question has been answered by presenting the two mainstream DNN-based approaches, i.e., WENGO (see Section 4.1) and D-TRAPPD (see Section 4.2). WENGO extracts embeddings from pre-trained monolingual word embedding models (i.e., FastText) and computes the cosine similarity score between the input text pairs. Conventional machine learning classifiers (e.g., SVM, NB, etc.) were used to differentiate between paraphrased and non-paraphrased text pairs. Moreover, D-TRAPPD, a computationally expensive approach, is presented and fine-tuned to detect paraphrasing in Urdu text pairs. Mainstream CNNs and LSTM architectures were used to capture the input text pairs' salient features and long-term dependencies.

Finally, the third question has been answered by presenting two types of comparisons. Firstly, a comparison was conducted between all the newly proposed (i.e., WENGO and D-TRAPPD) approaches. It was found that D-TRAPPD has outperformed the WENGO-based approaches for both evaluation tasks (see Section 5.5). Secondly, D-TRAPPD was compared with the existing surface-level approaches for Urdu text reuse and plagiarism detection. Experimental results showed that D-TRAPPD performed better than WENGO and the existing surface-level similarity assessment approaches for paraphrase detection and text reuse and plagiarism detection in Urdu texts. All the evaluations performed on SUSPC were external, meaning that we did not perform any Urdu-specific modifications (e.g., stemming, lemmatization, etc.) to evaluate the newly generated SUSPC corpus.

In the future, we aim to further explore the D-TRAPPD approach by focusing on Urdu language-specific modifications such as orthograph, syntax, semantics, etc. Further, we will explore more recent approaches (as mentioned in section 2), including deep learning (GRU, Bi-LSTM, Multi-perspective LSTMs, etc.), and transformers-based approaches for automatic paraphrase generation and detection

and other related tasks. We will explore language agnostic (e.g., LaBERT) and multilingual models (e.g., mBERT, XLM-RoBERTa) to detect paraphrasing in Urdu text reuse and plagiarism cases.

8 Acknowledgements

This work is funded by the Higher Education Commission (HEC) of Pakistan under the National Research Program for Universities (NRPU) grant, and the Information Technology University, Lahore, Pakistan.

References

Wang, Weibo, Aminul Islam, Abidalrahman Moh'd, Axel J. Soto, and Evangelos E. Milios (2021) Nonuniform language in technical writing: Detection and correction, *Natural Language Engineering 27, no. 3*, pp. 293–314.

Noraset, Thanapon and Lowphansirikul, Lalita and Tuarob, Suppawong (2021) Wabiqa: A wikipedia-based thai question-answering system, *Information Processing & Management*, pp. 102–431, Elsevier.

Zandie, Rohola, and Mohammad H. Mahoor (2022) Topical language generation using transformers, *Natural Language Engineering 1, no. 1*, pp. 1–23, DOI: https://doi.org/10.1017/S1351324922000031.

Paris, Cécile L and Swartout, William R and Mann, William C (2013) Natural language generation in artificial intelligence and computational linguistics, *Information Processing & Management*, vol. 119, Springer Science & Business Media.

Forsythe, Chris and Bernard, Michael L and Goldsmith, Timothy E (2006) Cognitive systems: Human cognitive models in systems design, *Psychology Press*.

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

Ji, Donghong and Tao, Peng and Fei, Hao and Ren, Yafeng (2020) An end-to-end joint model for evidence information extraction from court record document, *Information Processing & Management*, vol. 57, pp. 202–305, Elsevier.

Farhan, Wael and Talafha, Bashar and Abuammar, Analle and Jaikat, Ruba and Al-Ayyoub, Mahmoud and Tarakji, Ahmad Bisher and Toma, Anas (2020) Unsupervised dialectal neural machine translation, *Information Processing & Management*, vol. 57, pp. 102–181, Elsevier.

Ehsan, Nava and Shakery, Azadeh (2016) Candidate document retrieval for cross-lingual plagiarism detection using two-level proximity information, *Information Processing & Management*, vol. 52, pp. 1004–1017, Elsevier.

Jing, Nan and Liu, Qi and Sugumaran, Vijayan (2021) A blockchain-based code copyright management system, *Information Processing & Management*, vol. 58, pp. 102–518, Elsevier.

Clough, Paul and Gaizauskas, Robert and Piao, Scott SL and Wilks, Yorick (2002) Meter: Measuring text reuse, *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 152–159.

Potthast, Martin and Stein, Benno and Barrón-Cedeño, Alberto and Rosso, Paolo (2010) An evaluation framework for plagiarism detection, *Proceedings of the 23rd international conference on computational linguistics: Posters*, pp. 997–1005, Association for Computational Linguistics.

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

Sharjeel, Muhammad and Rayson, Paul and Nawab, Rao Muhammad Adeel (2016) UPPC-Urdu paraphrase plagiarism corpus, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 1832–1836.

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

Dolan, Bill and Quirk, Chris and Brockett, Chris (2004) Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources, *Proceedings of the 20th international conference on Computational Linguistics*, pp. 350 , Association for Computational Linguistics.

Dolan, William B and Brockett, Chris (2005) Wabiqa: Automatically constructing a corpus of sentential paraphrases, *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

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

Potthast, Martin and Hagen, Matthias and Gollub, Tim and Tippmann, Martin and Kiesel, Johannes and Rosso, Paolo and Stamatatos, Efstathios and Stein, Benno (2013) Overview of the 5th international competition on plagiarism detection, *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pp. 301–331, CELCT.

Franco-Salvador, Marc and Rosso, Paolo and Montes-y-Gómez, Manuel (2016) A systematic study of knowledge graph analysis for cross-language plagiarism detection, *Information Processing & Management*, vol. 52 pp. 550–570, Elsevier.

Alvi, Faisal and El-Alfy, El-Sayed M and Al-Khatib, Wasfi G and Abdel-Aal, Radwan E (2012) Analysis and extraction of sentence-level paraphrase sub-corpus in CS education, *Proceedings of the 13th annual conference on Information technology education*, pp. 49–54.

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

John Wieting,Mohit Bansal, Kevin Gimpel and Karen, Livescu (2015) Towards Universal Paraphrastic Sentence Embeddings, *4th International Conference on Learning Representations*.

Sanjeev Arora, Yingyu Liang, Tengyu Ma (2017) A Simple but Tough-to-Beat Baseline for Sentence Embeddings, *5th International Conference on Learning Representations*.

Xu, Wei and Callison-Burch, Chris and Dolan, Bill (2015) Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit), *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pp. 1–11.

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

Daud, Ali and Khan, Wahab and Che, Dunren (2017) Urdu language processing: a survey, *Artificial Intelligence Review*, vol. 47, pp. 279–311, Springer.

Haneef, Israr and Adeel Nawab, Rao Muhammad and Munir, Ehsan Ullah and Bajwa, Imran Sarwar (2019) Design and Development of a Large Cross-Lingual Plagiarism Corpus for Urdu-English Language Pair, *Scientific Programming*, pp. 102–431, Hindawi.

Muneer, Iqra and Sharjeel, Muhammad and Iqbal, Muntaha and Nawab, Rao Muhammad Adeel and Rayson, Paul (2019) CLEU-A Cross-language english-urdu corpus and benchmark for text reuse experiments, *Journal of the Association for Information Science and Technology*, vol. 70, pp. 729–741, Wiley Online Library.

Sanchez-Perez, Miguel A and Sidorov, Grigori and Gelbukh, Alexander F (2014) A Winning Approach to Text Alignment for Text Reuse Detection at PAN 2014., *CLEF (Working Notes)*, pp. 1004–1011.

Stein, Benno and Rosso, Paolo and Stamatatos, Efstathios and Koppel, Moshe and Agirre, Eneko (2009) 3rd PAN workshop on uncovering plagiarism, authorship and social software misuse, *25th annual conference of the spanish society for natural language processing (SEPLN)*, pp. 1–77.

Potthast, Martin and Eiselt, Andreas and Barrón Cedeño, Luis Alberto and Stein, Benno and Rosso, Paolo (2011) Overview of the 3rd international competition on plagiarism detection, *CEUR workshop proceedings*, vol. 1177.

Gaizauskas, Robert and Foster, Jonathan and Wilks, Yorick and Arundel, John and Clough, Paul and Piao, Scott (2001) The METER corpus: a corpus for analysing journalistic text reuse, *Proceedings of the corpus linguistics 2001 conference*, pp. 214–223.

Clough, Paul and others (2003) Old and new challenges in automatic plagiarism detection, *National Plagiarism Advisory Service.*

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

El Desouki, Mohamed I and Gomaa, Wael H (2019) Exploring the Recent Trends of Paraphrase Detection, *International Journal of Computer Applications*, vol. 975, pp. 000–8887.

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

Vrbanec, Tedo and Meštrović, Ana (2020) Corpus-Based Paraphrase Detection Experiments and Review, *Information*, vol. 11, pp. 000–241, Multidisciplinary Digital Publishing Institute.

Kim, Yoon (2014) Convolutional neural networks for sentence classification, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751.

Kim, Yoon and Jernite, Yacine and Sontag, David and Rush, Alexander M. (2015) Character-Aware Neural Language Models, *Proceedings of the AAAI conference on artificial intelligence.*

Wang, Zhiguo and Mi, Haitao and Ittycheriah, Abraham (2016) Sentence similarity learning by lexical decomposition and composition, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1340–1349.

Jeena Kleenankandy and K. Nazeer (2020) An enhanced Tree-LSTM architecture for sentence semantic modeling using typed dependencies, *Information Processing & Management*, vol. 57, Elsevier.

Yin, Wenpeng and Schütze, Hinrich and Xiang, Bing and Zhou, Bowen (2016) Abcnn: Attention-based convolutional neural network for modeling sentence pairs, *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 259–272, MIT Press.

Zhiguo Wang and Wael Hamza and Radu Florian (2017) Bilateral Multi-Perspective Matching for Natural Language Sentences, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 4144–4150.

Hochreiter, Sepp and Schmidhuber, Jürgen (1997) Long short-term memory, *Neural computation*, vol. 9 pp. 1735–1780, MIT Press.

Mueller, Jonas and Thyagarajan, Aditya (2016) Siamese recurrent architectures for learning sentence similarity, *Thirteenth AAAI conference on artificial intelligence.*

Marelli, Marco and Bentivogli, Luisa and Baroni, Marco and Bernardi, Raffaella and Menini, Stefano and Zamparelli, Roberto (2014) SemEval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment, *SemEval-2014.*

Shakeel, Muhammad Haroon and Karim, Asim and Khan, Imdadullah (2020) A multi-cascaded model with data augmentation for enhanced paraphrase detection in short texts, *Information Processing & Management*, vol. 57, pp. 102–204, Elsevier.

Bell, Allan (1991) The language of news media, Blackwell Oxford.

Fries, Udo (1997) Summaries in newspapers: A textlinguistic investigation, *The structure of texts*, pp. 47–63.

Jing, Hongyan and McKeown, Kathleen R (1999) The decomposition of human-written summary sentences, *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 129–136.

Levenshtein, Vladimir I (1966) Binary codes capable of correcting deletions, insertions, and reversals, *Soviet physics doklady*, vol. 10, pp. 707–710.

Cohen, Jacob (1960) A coefficient of agreement for nominal scales, *Educational and psychological measurement*, vol. 20, pp. 37–46, Sage Publications.

Vulić, Ivan and Moens, Marie-Francine (2015) Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings, *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 363–372.

Goldberg, Yoav and Hirst, Graeme (2017) Neural Network Methods in Natural Language Processing, *9781627052986 (zitiert auf Seite 69)*.

Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, Greg S and Dean, Jeff (2013) Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems*, pp. 3111–3119.

Yu, Liang-Chih and Wang, Jin and Lai, K Robert and Zhang, Xuejie (2017) Refining word embeddings for sentiment analysis, *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 534–539.

Klein, Guillaume and Kim, Yoon and Deng, Yuntian and Senellart, Jean and Rush, Alexander M (2017) Opennmt: Open-source toolkit for neural machine translation, *Proceedings of ACL 2017, System Demonstrations*, pp. 67–72.

Kenter, Tom and De Rijke, Maarten (2015) Short text similarity with word embeddings, *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp. 1411–1420.

Ganguly, Debasis and Roy, Dwaipayan and Mitra, Mandar and Jones, Gareth JF (2015) Word embedding based generalized language model for information retrieval, *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 795–798.

Nair, Vinod and Hinton, Geoffrey E (2010) Rectified linear units improve restricted boltzmann machines, *ICML*.

Zhang, Xiang and Zhao, Junbo and LeCun, Yann (2015) Character-level convolutional networks for text classification, *Advances in neural information processing systems*, pp. 649–657.

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

Grave, Edouard and Bojanowski, Piotr and Gupta, Prakhar and Joulin, Armand and Mikolov, Tomas (2018) Learning Word Vectors for 157 Languages, *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Goodfellow, Ian and Bengio, Yoshua and Courville, Aaron and Bengio, Yoshua (2016) Deep Learning, *Information Processing & Management*, vol. 2, MIT press Cambridge.

Chicco, Davide (2021) Siamese neural networks: An overview, *Artificial Neural Networks*, pp. 73–94, Springer.

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

Han, Jun and Moraga, Claudio (1995) The influence of the sigmoid function parameters on the speed of backpropagation learning, *International Workshop on Artificial Neural Networks*, pp. 195–201, Springer.

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

Pennington, Jeffrey and Socher, Richard and Manning, Christopher D (2014) Glove: Global vectors for word representation, *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

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

Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina (2018) Bert: Pre-training of deep bidirectional transformers for language understanding, *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*, pp. 4171–4186.

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

Radford, Alec and Narasimhan, Karthik and Salimans, Tim and Sutskever, Ily (2018) Improving language understanding by generative pre-training, *OpenAI*.

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

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

Wang, Wei and Yan, Ming and Wu, Chen (2018) Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1705–1714.

Kingma, Diederik P and Ba, Jimmy (2015) Adam: A method for stochastic optimization, *3rd International Conference on Learning Representations*, 2015.

Simon, Herbert A and Newell, Allen (1958) Heuristic problem solving: The next advance in operations research, *Operations research*, pp. 1–10.

Baeza-Yates, Ricardo and Ribeiro-Neto, Berthier and others (1999) Modern information retrieval, *vol. 463* ACM press New York.

Nighojkar, Animesh and Licato, John (2021) Improving paraphrase detection with the adversarial paraphrasing task, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 7106–7116.

Kadotani, Sora and Kajiwara, Tomoyuki and Arase, Yuki and Onizuka, Makoto (2021) Edit Distance Based Curriculum Learning for Paraphrase Generation, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pp. 229–234.

Meng, Yuxian and Ao, Xiang and He, Qing and Sun, Xiaofei and Han, Qinghong and Wu, Fei and Fan, Chun and Li, Jiwei (2021) ConRPG: Paraphrase Generation using

Contexts as Regularizer, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Springer, pp. 2551–2562.

Corbeil, Jean-Philippe and Ghavidel, Hadi Abdi (2021) Assessing the Eligibility of Back-translated Samples Based on Semantic Similarity for the Paraphrase Identification Task, *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pp. 301–308.

Amjad, Maaz and Sidorov, Grigori and Zhila, Alisa (2020) Data augmentation using machine translation for fake news detection in the Urdu language, *Proceedings of the 12th language resources and evaluation conference*, pp. 2537–2542.

Foltỳnek, Tomáš and Meuschke, Norman and Gipp, Bela (2019) Academic plagiarism detection: a systematic literature review, *ACM Computing Surveys*, pp. 1–42.

Maurer, Hermann A and Kappe, Frank and Zaka, Bilal (2006) Plagiarism-A survey, *J. UCS*, vol.12, pp. 1050–1084.

Butakov, Sergey and Scherbinin, Vladislav (2009) The toolbox for local and global plagiarism detection, *Computers & Education*, vol.52, Elsevier, pp. 781–788.

Barrón-Cedeno, Alberto and Rosso, Paolo and Agirre, Eneko and Labaka, Gorka (2010) Plagiarism detection across distant language pairs, *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 37–45.

Cedeno, Alberto Barrón and others (2013) On the mono-and cross-language detection of text re-use and plagiarism, *Procesamiento del Lenguaje Natural*, pp. 103–105.

Nawab, Rao Muhammad Adeel (2012) PhD Thesis: Mono-lingual paraphrased text reuse and plagiarism detection, *University of Sheffield*.

Burrows, Steven and Potthast, Martin and Stein, Benno (2013) Paraphrase acquisition via crowd sourcing and machine learning, *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol: 4.

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

Barrón-Cedeno, Alberto (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*.

Amjad, Maaz and Sidorov, Grigori and Zhila, Alisa (2020) Data augmentation using machine translation for fake news detection in the Urdu language, *Proceedings of the 12th language resources and evaluation conference*, pp. 2537–2542.

Amjad, Maaz and Sidorov, Grigori and Zhila, Alisa and Gómez-Adorno, Helena and Voronkov, Ilia and Gelbukh, Alexander (2020) Bend the truth: Benchmark dataset for fake news detection in Urdu language and its evaluation, *Journal of Intelligent & Fuzzy Systems*, pp. 2457–2469.

Shafi, Jawad and Iqbal, Hafiz Rizwan and Nawab, Rao Muhammad Adeel and Rayson, Paul (2022) UNLT: Urdu Natural Language Toolkit, *Natural Language Engineering*, pp. 01–36, DOI:10.1017/S1351324921000425.

Marelli, Marco and Menini, Stefano and Baroni, Marco and Bentivogli, Luisa and Bernardi, Raffaella and Zamparelli, Roberto (2014) A SICK cure for the evaluation of compositional distributional semantic models, *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pp. 216–223.

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