Alvi *et al. Int J Educ Technol High Educ* (2021) 18:42
https://doi.org/10.1186/s41239-021-00277-8

International Journal of Educational Technology in Higher Education

RESEARCH ARTICLE Open Access

Paraphrase type identification for plagiarism detection using contexts and word embeddings

Faisal Alvi[1*], Mark Stevenson[2] and Paul Clough[3]

*Correspondence:
alvif@kfupm.edu.sa
[1] Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
Full list of author information is available at the end of the article

Abstract

Paraphrase types have been proposed by researchers as the paraphrasing mechanisms underlying acts of plagiarism. Synonymous substitution, word reordering and insertion/deletion have been identified as some of the common paraphrasing strategies used by plagiarists. However, similarity reports generated by most plagiarism detection systems provide a similarity score and produce matching sections of text with their possible sources. In this research we propose methods to identify two important paraphrase types – synonymous substitution and word reordering in paraphrased, plagiarised sentence pairs. We propose a three staged approach that uses context matching and pretrained word embeddings for identifying synonymous substitution and word reordering. Our proposed approach indicates that the use of Smith Waterman Algorithm for Plagiarism Detection and ConceptNet Numberbatch pretrained word embeddings produces the best performance in terms of F_1 scores. This research can be used to complement similarity reports generated by currently available plagiarism detection systems by incorporating methods to identify paraphrase types for plagiarism detection.

Keywords: Plagiarism, Plagiarism detection, Paraphrase types, Synonymous substitution, Word reordering, Context matching, Word embeddings

Introduction

Plagiarism has been on the rise with the widespread availability of digital information and the ease with which it can be copied. Recent and past surveys suggest an increase in cases of plagiarism in both academic work and scientific literature (Dias and Bastos 2014; Fatima et al. 2019; Kauffman and Young 2015; Schmidt Hanbidge et al. 2020). In response, commercial plagiarism detection systems (e.g. TurnitIn, iThenticate) have been developed in order to detect and prevent plagiarism. These plagiarism detection systems calculate a similarity score, and provide a report displaying the positions of textual matches between the source and plagiarised documents. However, the final decision on whether a reported similarity score represents a genuine case of plagiarism rests with a human evaluator (McKeever 2006). This is because the decision "*requires a careful consideration of these annotated matches by a person to determine which, if any, constitute plagiarism*" (Mphahlele and McKenna 2019).

In past work researchers have identified several types of plagiarism, such as no obfuscation (copy and paste), translation obfuscation and summarisation plagiarism (Alzahrani et al. 2012; Potthast et al. 2014). *Paraphrase plagiarism* (Carmona et al. 2018) refers to and captures situations in which text is copied from sources and obfuscated using lexical and semantic transformations, such as synonymous substitutions, word reordering and rephrasing. These modifications change the surface form of a text, but preserve its overall meaning, thereby making it difficult for computers and humans to identify and prove that plagiarism has occurred.

In the context of paraphrase plagiarism, several researchers (Barrón-Cedeño et al. 2013; Bhagat and Hovy 2013; Sun and Yang 2015) have identified a number of different *paraphrase types*: specific categories of text rewrite operations a plagiarist might use in order to obfuscate copied text. In a study based on analysing a collection of simulated cases of plagiarism from the PAN PC-10 corpus, Barrón-Cedeño et al. (2013) have reported that same polarity or synonymous substitution, i.e., the substitution of synonymous words and phrases, forms the largest proportion of paraphrase types in plagiarised text. These findings have been corroborated by Bhagat and Hovy (2013) and Sun and Yang (2015), who have also stated that synonym substitution forms a large proportion of rewrite operations in paraphrased texts. Furthermore, these research works have also identified word reordering as a less frequently used, but an important paraphrase type in the context of plagiarism.

Surveys on plagiarism detection tools, such as (Kanjirangat and Gupta 2016; Weber-Wulff 2014) provide useful information on the current state of effectiveness of these tools. A recent survey on testing of plagiarism detection tools by Foltỳnek et al. (2020) stated that out of 15 available plagiarism detection tools, none satisfied their criteria of being labelled as a useful system. In their words, "*the results... indicate insufficient systems. The performance on plagiarism from Wikipedia disguised by a synonym replacement was generally poorer and almost no system was able to satsfyingly identify manual paraphrase plagiarism.*" These tests indicate that despite advances in educational technology for plagiarism detection, synonym replacement and paraphrasing represent a challenge for plagiarism detection systems.

In this work we propose a three staged approach to identify synonymous substitutions and word reordering in paraphrased, plagiarised sentence pairs. The primary motivation for this research is to develop methods that identify paraphrase types used in plagiarism. This information about detected paraphrase types can be useful to a human evaluator in making an informed decision about the occurrence of plagiarism. We present a novel approach that uses context matching and pretrained word embeddings to identify paraphrase types in plagiarised sentence pairs. To the best of our knowledge, identifying synonymous substitutions and word reorderings using approaches reported in this paper has not been carried out in previous work.

Our dataset consists of pairs of paraphrased sentences annotated for paraphrase types from the Corpus of Plagiarised Short Answers (Clough and Stevenson 2011). Our proposed three staged approach begins with preprocessing which includes sentence filtering. These pairs of paraphrased sentences are then processed as inputs to two parallel paths for identifying the two paraphrase types. For identifying reordered word segments (word reordering) we use paraphrase patterns and permutations of words and text

segments. For the detection of synonymous substitutions we use the Smith Waterman Algorithm and ConceptNet Numberbatch pretrained word embeddings. Our experiments report an F_1 score of 0.906 for identifying word reorderings and an F_1 score of 0.802 for identifying synonymous substitutions for the entire dataset.

The rest of this article is organised as follows: Section "Background" provides a review of prior work on plagiarism detection, paraphrase plagiarism detection and monolingual text alignment. Section "Experimental setup" gives the experimental setup by identifying measurement parameters and the dataset used. Section "Proposed approach" provides details of the proposed approach using the three staged framework for detection of word reordering and synonymous substitution. Section "Results and discussion" states the results of our evaluations and comparison by varying alignment method and word embeddings. Finally, Section "Conclusions and future work" concludes the paper by highlighting our contributions and future work.

Background

In this section we provide the background pertinent to the problem of identifying paraphrase types in plagiarism detection. We begin by stating a brief overview of plagiarism and its various forms (Section "Forms of plagiarism") as well briefly discuss the motivations for plagiarism. This is followed by a description of paraphrase plagiarism and its position in various plagiarism taxonomies in research surveys (Section "Paraphrase Plagiarism"). We then present an overview of paraphrase types identified in plagiarised texts by describing various paprahrase typologies (Section "Paraphrase typologies"). We conclude this section by presenting an overview of methods for detecting paraphrase plagiarism (Section "Plagiarism detection") and a brief description of monolingual textual alignment (Section "Monolingual Textual Alignment").

Forms of plagiarism

Text reuse is the reuse of text either in its original or modified form (Clough 2010). Plagiarism is a case of text reuse, but when proper attribution is lacking and can be defined as *"the use of ideas and/or words from sources without giving due acknowledgement"* (Meuschke and Gipp 2013).

Barrón-Cedeño (2012, p. 18) states some of the reasons why students engage in plagiarism which can be classified as teacher oriented, student oriented and educational system oriented. These reasons can be attributed to (a) a lack of commitment by teachers (such as repeating the same assignments), (b) students' attitude to school and learning process (such as investing the least amount of time and effort), and (c) lack of clear rules from the educational institution.

Several types of plagiarism have been identified in the literature from various perspectives which include: no obfuscation (copy and paste), translation obfuscation, random obfuscation and summary obfuscation (Potthast et al. 2015, 2014). Both translation obfuscation and summary obfuscation involve the use of paraphrasing.

Paraphrase plagiarism

Paraphrase plagiarism can be defined as a form of plagiarism wherein rephrasing, substitution and restructuring of words and phrases may be used to obfuscate copied text.

Table 1 Classification indicating paraphrase plagiarism in various research surveys

Research survey & plagiarism classification
(Foltýnek et al. 2019)
Plagiarism → Semantics Preserving Plagiarism → Paraphrasing
(Weber-Wulff 2014)
Plagiarism → Disguised, Structural Plagiarism → Paraphrasing
(Alzahrani et al. 2012)
Plagiarism → Intelligent Plagiarism → Text Manipulation → Paraphrasing
(Maurer et al. 2006)
Forms of Plagiarism → Paraphrase Plagiarism

Researchers have identified paraphrase plagiarism in surveys ranging over two decades (Alzahrani et al. 2012; Foltýnek et al. 2019; Maurer et al. 2006; Weber-Wulff 2014). Table 1 provides a classification of paraphrasing as a form of plagiarism from selected past surveys. From Table 1 it can be observed that paraphrase plagiarism has been classified as a sub-type of various plagiarism types such as semantics-preserving (Foltýnek et al. 2019), disguised and structural (Weber-Wulff 2014), intelligent plagiarism (Alzahrani et al. 2012) and as a form of plagiarism (Maurer et al. 2006).

An example of plagiarism in journal articles using source and paraphrased text segments (Sun and Yang 2015) as a form of substitution is stated as follows (substitutions are marked with superscripts):

1 (Source Text) These findings are relevant[1]... In particular, the interactive virtual-labs[2] are effective... to monitor[3] the learning process and to determine[4] whether learning is taking place as planned.
2 (Paraphrased Text) These findings are important[1]... In particular, the interactive VLs[2] are effective... in monitoring[3] the learning process and determining[4] whether learning is taking place as planned.

Paraphrase typologies

Paraphrase Typologies have been proposed from various perspectives such as discourse analysis and computational linguistics (Vila et al. 2014). From a plagiarism detection perspective, Barrón-Cedeño et al. (2013) have proposed a paraphrase typology comprising twenty paraphrase types classified into four categories. Some of the more important paraphrase types in their work are: same polarity substitution, opposite polarity substitution and modal verb change. Sun and Yang (2015) have also identified paraphrase types as paraphrasing strategies particularly in the context of plagiarism in journal articles. Some of the important paraphrasing strategies in their proposed list are: substitution, insertion/deletion, reordering. Likewise (Bhagat and Hovy 2013) have proposed a comprehensive list of paraphrase types with important types being synonym substitution, antonym substitution and change of modality. We present a partial mapping of these paraphrase types in Table 2 between the works of Barrón-Cedeño et al. (2013) and Bhagat and Hovy (2013). It can be observed that the mappings generally represent

Table 2 Partial mapping of paraphrase types between (Bhagat and Hovy 2013) and (Barrón-Cedeño et al. 2013)

#	Bhagat and Hovy (2013)'s list	Barrón-Cedeño et al. (2013)'s list
1	Synonym Substitution	Same Polarity Substitution
2	Converse Substitution	Converse Substitution
3	Repetition/Ellipsis	Ellipsis
4	Antonym Substitution	Opposite Polarity Substitution
5	Change of Person	Direct/Indirect Style Alternations
6	General/Specific Substitution	Same Polarity Substitution
7	Change of modality	Modal Verb Changes
8	Approx. numerical equivalences	Same Polarity Substitution
9	-	Punctuation and format changes
10	External Knowledge	-

equivalent paraphrase types, for example, antonym substitution being equivalent to opposite polarity substitution.

From a frequency of occurrence perspective, quantitative data from Barrón-Cedeño et al. (2013), Bhagat and Hovy (2013) and Sun and Yang (2015) show that substitution of synonymous words and phrases is a frequently occuring paraphrase type. This can be corroborated from the percentages of synonymous substitutions which are 45% (Barrón-Cedeño et al. 2013), 37% (Bhagat 2009) and 32% (Sun and Yang 2015) of all paraphrase types in their respective datasets. These findings highlight the importance of synonym substitution as a frequently used paraphrase mechanism underlying acts of plagiarism.

Plagiarism detection

Plagiarism detection refers to techniques, tools and methods used for automated detection of plagiarism, since manual detection becomes infeasible with large amounts of information. In this section we provide a brief overview of various approaches proposed for the detection of plagiarism and paraphrase plagiarism. In particular, approaches based on character and word *n*-gram similarity (Bensalem et al. 2019; Sánchez-Vega et al. 2017), vector space models (Sanchez-Perez et al. 2014), natural language processing (Chong 2013; Kanjirangat and Gupta 2018) machine translation similarity metrics (Madnani et al. 2012) and alignment algorithms (Nichols et al. 2019) have been successfully applied towards plagiarism detection. Despite these advances, plagiarism detection when text has been paraphrased remains a challenge due to limited success in measuring semantic overlap (Carmona et al. 2018).

A number of recent research works on paraphrase plagiarism detection have adopted various approaches. Carmona et al. (2018) introduce two new semantically informed distance measures between texts, which are based on the Jaccard similarity measure and Levenshtein edit distance by merging WordNet and Word2Vec based similarity measures. Sanchez-Perez (2018) use WordNet similarity metrics, as well as similarity metrics from Word2Vec and GloVe pretrained word embeddings. Sánchez-Vega et al. (2017) combine six different character level features to compute textual similarity based on the Dice coefficient. Kanjirangat and Gupta (2018) propose a new syntactic-semantic similarity measure based on the WUP WordNet similarity, while Chitra and Rajkumar

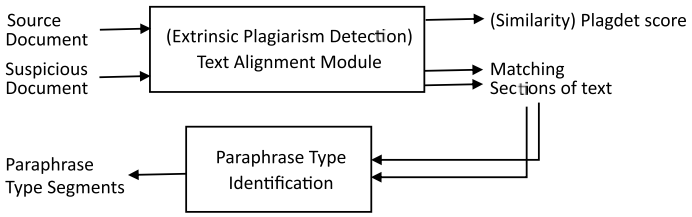


Fig. 1 Model proposing Integration of a Paraphrase Type Identification Method within a Plagiarism Detection System

(2016) have utilised machine learning to create a paraphrase recogniser for plagiarism detection.

These research works have used novel approaches for paraphrase plagiarism detection and therefore represent state of the art. However, paraphrase type identification can be considered as a different problem as compared to paraphrase plagiarism detection. This is because in paraphrase type identification we aim to identify paraphrase types within text pairs that have been marked as plagiarised.

From a research perspective, a proposed method to identify paraphrase types can be integrated within an existing plagiarism detection system. This has been modeled in Fig. 1, where a proposed method for identifying paraphrase types is appended to a textual alignment module for extrinsic plagiarism detection. The output of the plagiarism detection module (matching sections of text) can be sent as input to the paraphrase type identification module. This will subsequently identify paraphrase types within matched sections of text, thereby achieving successful integration of paraphrase type identification within a plagiarism detection system.

Monolingual textual alignment

Textual alignment is the task of linking similar textual entities between two textual units. Bitext Alignment (Tiedemann 2011) is a particular application of textual alignment in machine translation, where words or phrases in sentences of different languages are linked. Monolingual textual alignment can be defined as, "*the task of discovering and aligning similar semantic units in a pair of sentences expressed in a natural language*" (Sultan et al. 2014). Monolingual textual alignment is of particular interest to us as it is similar to the problem of identifying paraphrase types in plagiarised text.

Figure 2 illustrates monolingual textual alignment between two sentences (Wang et al. 2013) as a word similarity matrix with shaded squares representing alignments. Here, '*read into*' ↔ '*interpeted*' can be considered as word to phrase alignment in addition to other word and phrase alignments.

Alignment tools, such as Meteor, GIZA++ and Berkeley Aligner, are readily available for monolingual textual alignment. These tools have been used for text reuse detection, such as the Berkeley Word Aligner having been used for aligning sentences from a parallel corpus on a token level (Moritz et al. 2018).

Sultan et al. (2014) have proposed a successful pipeline architecture for aligning words between source and target sentences as follows: (i) aligning identical word sequences, (ii) aligning named entities, then (iii) aligning content words, and finally (iv) aligning stopwords. This sentence aligner was one of the best performing aligners at Semeval-2015.

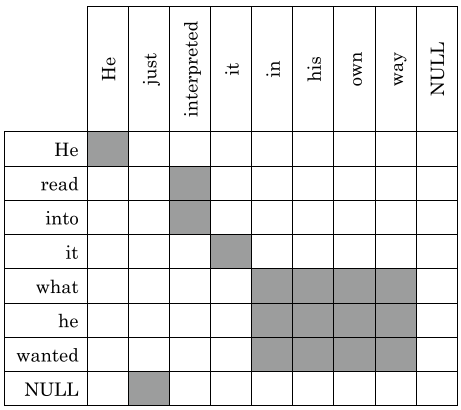


Fig. 2 Alignment Example for Textual Entailment (Wang et al. 2013)

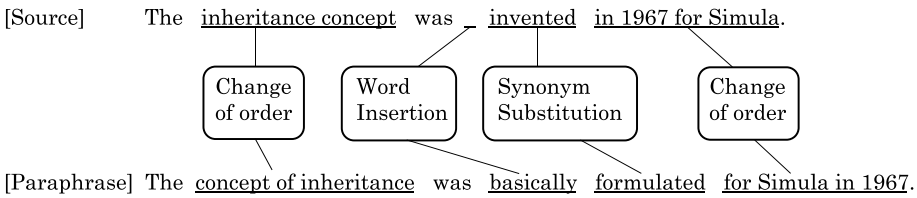


Fig. 3 Examples of Paraphrase Types in a pair of source and paraphrased sentences

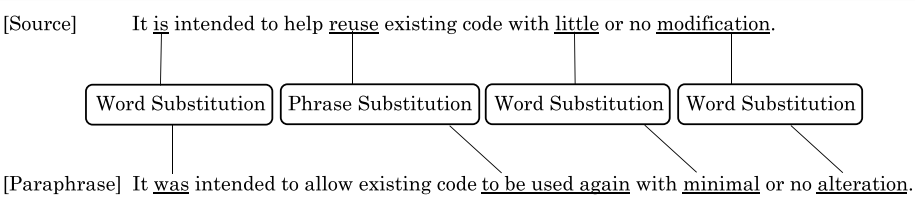


Fig. 4 Examples of Synonymous Substitutions in a pair of source and paraphrased sentence

Experimental setup

This section provides details about the experimental setup. We begin with a formal description of the problem, followed by details of the dataset used and the measurement parameters.

Problem description

In this research our objective is the detection of paraphrase types in text pairs that have been marked as plagiarised. We focus on detecting two fundamental paraphrase types: (a) change of order (or word reordering), and (b) synonymous substitution. Our input data consists of pairs of source and paraphrased sentences available as the Subcorpus of

Paraphrased Sentences (Alvi et al. 2012) extracted from the Corpus of Plagiarised Short Answers (Clough and Stevenson 2011).

In Figure 3 we provide examples of various paraphrase types in source and paraphrased sentences including change of order, synonym substitution and insertion on a pair of source and paraphrased sentences. It can be observed that there are two changes of order (word reorderings), one word insertion and one synonym substitution in the sentence pair. Figure 4 gives further examples of synonym substitutions which include three word substitutions and one phrase substitution. Both examples are on sentence pairs from Task A of the Corpus of Plagiarised Short Answers.

The objective of this research is to detect matching segments of text as paraphrase types from a pair of source and paraphrased sentences. We also aim to identify whether an extracted text segment (as a paraphrase type) is a synonymous substitution or a word reordering.

Dataset used

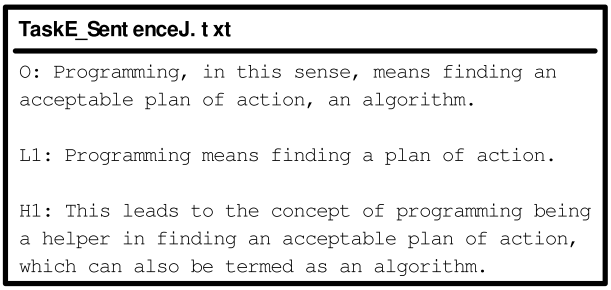
We use the Corpus of Plagiarised Short Answers (Clough and Stevenson 2011) as the data source for the detection of paraphrase types. The Corpus of Plagiarised Short Answers is a collection of simulated cases of plagiarism divided into five tasks and four levels of revision. The five tasks correspond to five questions posed to university students from Wikipedia, while the four levels of revision are (a) near copy, (b) light revision, (c) heavy revision, and (d) no plagiarism.

Alvi et al. (2012) have extracted a subcorpus of paraphrased pairs of sentences from the Corpus of Plagiarised Short Answers. This subcorpus consists of 101 files, each with a given source sentence from Wikipedia and the corresponding light and heavily revised paraphrased sentences across the five tasks. These sentence pairs represent actual instances of paraphrasing by university students, thereby simulating plagiarism. Figure 5 shows a sample file with a given source and two paraphrased sentences from Task E.

Selection of sentence pairs

We extract our collection of sentence pairs from the Subcorpus of Paraphrased Sentences. We follow the filtering criteria for the Microsoft Research Paraphrase Corpus (Dolan et al. 2004) outlined in detail in (Dolan and Brockett 2005). Criteria 1 and 2 are exactly adopted from the Microsoft Research Paraphrase Corpus, while criteria 3 and 4 are slight modifications, stated as follows:

1 *Sentence Length Criteria*: We include sentences having 5–40 words only. Sentences having less than 5 words or more than 40 words are excluded. The rationale here is to exclude sentences that are too short or long.
2 *Overlap Criteria*: We apply upper and lower limits of word overlap between sentences. To ensure minimal word overlap, sentences that share at least 3 words in common are included. For having some word diversity, we include sentence pairs whose edit distance is at least 3.


TaskE_Sent enceJ. t xt

O: Programming, in this sense, means finding an acceptable plan of action, an algorithm.

L1: Programming means finding a plan of action.

H1: This leads to the concept of programming being a helper in finding an acceptable plan of action, which can also be termed as an algorithm.

Fig. 5 A Sample File in the Subcorpus of Paraphrased Sentences (Alvi et al. 2012)

3 *Similarity Metric Criteria*: Sentence pairs are included whose sentence cosine similarity in terms of words is at least 0.25.
4 *Length Ratio*: Finally we include sentences such that the shorter sentence is at least 60% of the longer one in terms of the number of words.

These filtering criteria result in 211 pairs of source and paraphrased sentences which serve as our collection of sentence pairs for the annotation as well as the detection step. In the annotation phase, we annotate the sentence pairs for the presence of each of the two paraphrase types i.e., change of order (word reordering) and synonymous substitution.

Word reordering (change of order)

Word reordering or change of order has been identified as a paraphrase type in several research works in the context of plagiarism (Barrón-Cedeño et al. 2013; Sun and Yang 2015). However, the definition of change of order is quite general in these works and may span multiple paraphrase types such as addition of content words. We refer to (Sousa-Silva 2014) for a more specific definition of word reordering in the context of plagiarism as follows: *"Word reordering is used to describe the linguistic operations whereby the original words are reused, but in a different order"*. Likewise, Bhagat and Hovy (2013) refer to reordering of words in the context of paraphrasing as follows: "*The words in the new sentence were allowed to be reordered (permuted) if needed and only function words (and no content words) were allowed to be added to the new sentence."*

We define word reordering to be a permutation of the words in a phrase or a sentence with the addition of stopwords only such as prepositions, conjunctions or determiners. In the context of statistical machine translation, Bisazza and Federico (2016) present two examples from English that highlight our definition of word reordering as follows:

1 "I saw the cat" ↔ "The cat I saw"
2 "the tail of the cat" ↔ "the cat's tail"

In these examples, the same set of content words is used along with addition or removal of stopwords. Using the above definition of word reordering, we have annotated our collection of source and paraphrased sentence pairs from the Corpus of Plagiarised Short Answers. The statistics of our annotation are given in Table 3. We observe that the

Table 3 Task wise paraphrase type statistics for the dataset

Paraphrase types/tasks	Synonymous substitution	Word reordering	Total for each task
Task A	53	11	64
Task B	38	4	42
Task C	81	5	86
Task D	58	2	60
Task E	74	6	80
Total for Each Type	304 (91.57%)	28 (8.43%)	332 (100.00%)

number of word reorderings is much less than the number of substitutions which agrees with the observation in (Sousa-Silva 2014) i.e., *"this linguistic strategy (word reordering) is not as common as word substitution"*.

Synonymous substitution

We use the term synonymous substitution to refer to both word and phrasal substitutions in plagiarism. Our definition of synonymous substitution generally agrees with the definition of same polarity substitutions defined by Barrón-Cedeño et al. (2013). In general, a synonymous substitution can be considered as the replacement of a word or phrase with another one having exact or approximate meaning such that the overall sense of the sentence remains the same. Examples of change of order and synonymous substitutions appear in both Figs. 3 and 4. Table 3 provides the statistics of annotation of synonymous substitutions and word reordering.

From Table 3 it can be observed that the number of synonymous substitutions is much more as compared to word reorderings. From a quantitative perspective, word reorderings (28 annotations) are 8.43% of the entire set of annotations and 9.21% of substitutions. This is in general agreement with other datasets such as the number of paraphrasing strategies found by Sun and Yang (2015), where word reorderings are approximately 2.49% of the entire dataset and 6.64% of substitutions.

Measurement parameters

We use the information retrieval measures of precision, recall and F_1 score to measure the effectiveness of our proposed approach. The application of precision and recall for identifying paraphrase types is similar to that used in the PAN plagiarism detection evaluation labs (Potthast et al. 2015, 2014). In this method, each instance of a paraphrase type annotation is considered as a four-tuple $s = (s_{start}, s_{size}, t_{start}, t_{size})$, where

- s_{start} = starting index of the paraphrase type annotation in the source sentence,
- s_{size} = length of the paraphrase type annotation in the source sentence,
- t_{start} = starting index of the paraphrase type annotation in the paraphrased sentence,
- t_{size} = length of the paraphrase type annotation in the paraphrased sentence.

Similarly, we identify a detection as a four-tuple $r = (s'_{start}, s'_{size}, t'_{start}, t'_{size})$ detected using an algorithm or approach with similar definitions as described above.

A *match* between an annotation $s = (s_{start}, s_{size}, t_{start}, t_{size})$ and a detection $r = (s'_{start}, s'_{size}, t'_{start}, t'_{size})$ is the number of overlapping positions of characters represented as $s \cap r$. For the entire dataset, we calculate precision by dividing the size of each match by the size of corresponding detection; for recall, match size is divided by the size of the corresponding annotation. These individual quantities are then summed and further normalised by dividing by the number of instances for the respective measures (i.e., the number of detections for precision $|R|$, and the number of annotations $|S|$ for recall). This gives us mean averaged precision and recall. The F_1 score is the harmonic mean of both precision and recall and is given by the following equations:

$$precision = \frac{1}{|R|} \sum_{r \epsilon R} \frac{|\bigcup_{r \epsilon R}(s \cap r)|}{|r|}, \tag{1}$$

$$recall = \frac{1}{|S|} \sum_{s \epsilon S} \frac{|\bigcup_{s \epsilon S}(s \cap r)|}{|s|} \tag{2}$$

$$F_1\ score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3}$$

In the preceding equations R is the set of detections, while S is the set of annotations. The above computed F_1 score is *macro*-averaged where each paraphrase type annotation is given an equal weight irrespective of its size in terms of the number of characters. The rationale for choosing character matches instead of word matches is that character matches account for partial overlaps as compared to word matches which match whole words only.

Proposed approach

In this section we present our proposed approach for the detection of word reorderings and synonymous substitutions in paraphrased, plagiarised sentence pairs. Starting with the paraphrased sentence pairs from the Corpus of Plagiarised Short Answers, our proposed approach consists of the following three steps:

1. *Preprocessing*: In this stage, we apply punctuation removal and case folding. We then filter the sentence pairs according to the criteria set out in Section "Selection of sentence pairs". These pairs of sentences are then sent as input for detection of paraphrase types.
2. *Identification of Word Reorderings*: In this stage, we detect identical textual segments from the sentence pairs to used as inputs for the identification of word reorderings. We use permutations of identical textual segments and paraphrase patterns for the detection word reorderings.
3. *Identification of Synonymous Substitutions*: Finally, we identify synonymous substitutions within the sentence pairs using contexts, word alignment and word embeddings. We use ConceptNet Numberbatch pretrained word embeddings (Speer and Lowry-Duda 2017; Speer et al. 2017) and the Smith Waterman Algorithm for Plagiarism Detection (Glinos 2014) to detect these substitutions.

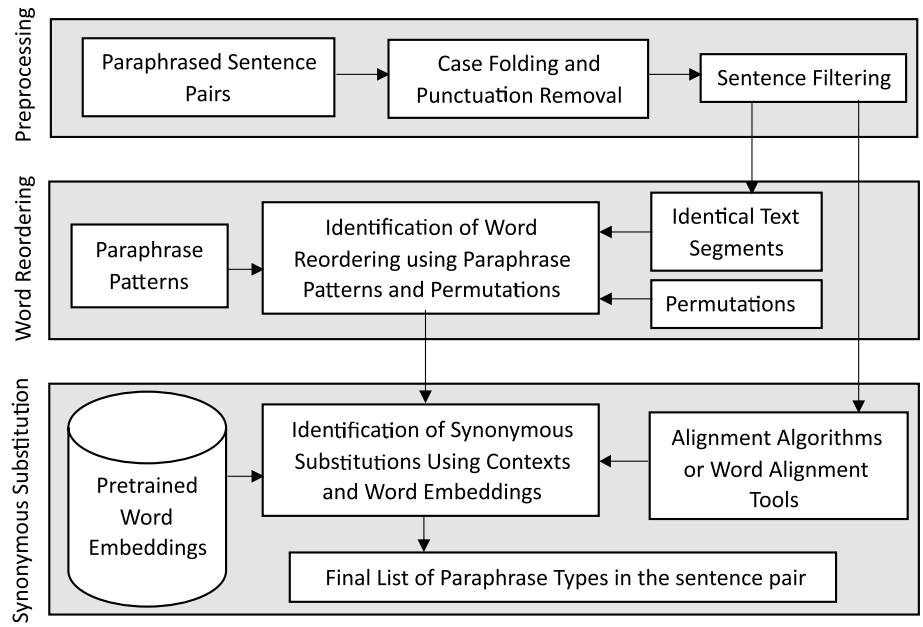


Fig. 6 Block Diagram of the Proposed Three Staged Approach

The output of these three stages consists of detected text segments as paraphrase types, identified as word reorderings or synonymous substitutions. Figure 6 shows the overall block diagram for our proposed approach. In the following subsections, we describe these stages in more detail.

Preprocessing

The preprocessing step is carried out to achieve the following tasks: (a) punctuation removal and case folding, and (b) filtering of sentences.

1 *Punctuation Removal and Case Folding*: We begin by removing all punctuation signs except for the apostrophe(') from the sentence pairs. The apostrophe is not removed as it represents the possessive form of several words such as Google's and Bayes' in the corpus. Furthermore, we also carry out case folding, i.e. all uppercase letters are converted to lowercase.

2 *Sentence Filtering*: In the second step we filter out sentences according to the criteria presented in Section "Selection of sentence pairs". In particular, for step 3 of the criteria, cosine similarity is calculated on sentence pairs by considering each sentence as a vector of words. More formally, let $\vec{S}$ and $\vec{T}$ be vector representations of two sentences. Therefore,

$$sim(\vec{S}, \vec{T}) = \frac{\vec{S} \cdot \vec{T}}{|\vec{S}| \cdot |\vec{T}|} \tag{4}$$

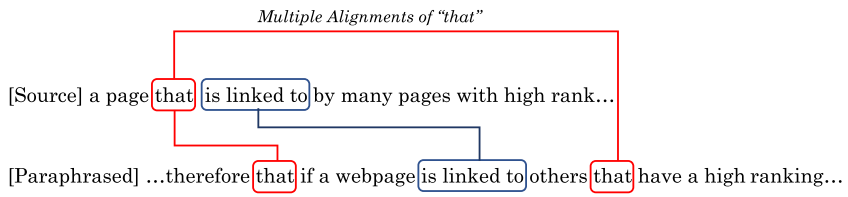


Fig. 7 Alignment of identical text fragment of size 3 and possible multiple alignments for 'that'

This value of $sim(\vec{S}, \vec{T})$ is used for sentence filtering.

Detection of word reorderings

The next stage in our approach is the detection of word reordering (or change of order) paraphrase type. Word reordering can be considered as a rearrangement of words along with addition or removal of function words as discussed in Section "Word Reordering". In our proposed approach, we detect word reorderings as the second stage before synonymous substitutions. This is because a word reordering is subject to a rearrangement of words from the source sentence only. In contrast, a synonymous substitution may involve replacement of words using words from the source sentence as well as from external sources. In this sense, a word reordering is more specific as compared to a substitution, hence we prioritise the detection of word reorderings.

We use permutations of identical textual segments and paraphrase patterns to detect word reorderings. This proceeds in the following steps:

Detection of identical text segments

We use the Greedy String Tiling Algorithm (Wise 1995) to find identical textual segments between the two sentences. We relax the definition of identical to near identical by considering words that end in an 's' or an apostrophe-s ('s or s') to be identical (e.g. "Google" and "Google's" are considered near identical). The Greedy String Tiling Algorithm (Wise 1995) identifies all matching string tiles between two strings starting with the longest matching substring and subsequently reducing the sizes of the matching substring. We find all matching string tiles of n-grams, (where $n \geq 2$) and align them between the source and the paraphrased sentences. These large identical fragments represent exactly reproduced text from the source sentence. However, we do not align identical unigrams as there is a high probability of misalignment due to multiple occurrences. Figure 7 shows the result of aligning identical textual segments between the source and the paraphrased sentences. We can observe that the word ("that") might result in misalignment due to multiple occurrences in the paraphrased sentence.

Permutations of identical text segments

In this step, we search for permutation patterns of (near) identical textual segments found in the previous stage. Any permutation of words or text segments between the source and paraphrased sentences is considered as a word reordering. For example, the sequence 'apples carrots bananas' (ACB) is a 3-permutation of 'bananas carrots

Table 4 Permutation patterns for 3-permutations of identical textual segments

Source	Paraphrased	Description
A B C	A B C	A B C is an identical form of A B C
A B C	A C B	B C ↔ C B is a 2-permutation with A as the left context
A B C	B A C	A B ↔ B A is a 2-permutation with C as the right context
A B C	B C A	A (BC) ↔ (BC) A is a 2-permutation of A and BC
A B C	C A B	(AB) C ↔ C (AB) is a 2-permutation of AB and C
A B C	C B A	A B C ↔ C B A is the only 3-permutation in the list

apples' (BCA) and hence can be considered as a word reordering. However, since permutations might repeat similar elements as left or right contexts, the search space for permutation patterns can be significantly reduced leading to efficient search. This is illustrated in the Table 4, where out of $3! = 6$ permutations for 3 elements, almost all 3-permutations can be considered as 2-permutations except for the last one.

Due to this reduction in the number of permutation patterns, we search for the patterns AB ↔ BA (2-permutation) and ABC ↔ CBA (3-permutation) in the sentence pairs.

Paraphrase patterns

Paraphrase patterns can be considered as sets of semantically equivalent paraphrases, with placeholders for words. Zhao et al. (2008) define paraphrase patterns as *"sets of semantically equivalent patterns, in which a pattern generally contains two parts i.e. the pattern word and the slots. For example, in the pattern 'X solves Y', 'solves' is the pattern word while 'X' and 'Y' are slots"*. The Paraphrase Database (PPDB) (Ganitkevich et al. 2013) is a collection of over 140 million paraphrase patterns in the English language.

For the detection of word reorderings, we consider paraphrase patterns in which the slots *X* and *Y* interchange their positions within the pattern. For example, the following pattern: "*X* announced by *Y* ↔ *Y* announced a *X*" appearing in (Bhagat 2009, p. 200) can be considered as a word reordering.

Some of the paraphrase patterns used for the detection of word reorderings in our sentence pairs are shown with examples as follows:

- *X* of *Y* ↔ *Y X* (University of Stanford ↔ Stanford University)
- *X* and *Y* ↔ *Y* and *X* (apples and oranges ↔ oranges and apples)
- *X* and *Y* ↔ *X*, *Y* (cars and trucks ↔ cars, trucks)
- *X* is *Y* ↔ *Y* has *X* (PageRank Algorithm is a trademark of Google ↔ A trademark of Google has PageRank Algorithm)

The above patterns are also reversible i.e., given the pattern '*X* of *Y* ↔ *Y X*', the pattern '*X Y* ↔ *Y* of *X*' is also a paraphrase pattern corresponding to a word reordering. It can also be observed that the pattern '*X* and *Y* ↔ *Y* and *X*' can also be considered as a 3-permutation of the form ABC ↔ CBA. Although we use just a few patterns since the number of annotations is small, the design of the approach is flexible to accommodate a large number of patterns based on dataset.

Using our approach a wide variety of word reordering text segments can be successfully detected, such as the examples shown in Figure 3. Even entire sentences written by a plagiarist using word reordering can be detected using this approach, as shown in the following example from Task B of the corpus:

- (Source) It is intended to help reuse existing code with little or no modification.
- (Paraphrased) With little or no modification it is intended to help reuse existing code.

This completes the description of our proposed approach for the detection of word reorderings.

Detection of synonymous substitutions

In this subsection, we present the details of the third stage of our proposed approach, i.e. the detection of synonymous substitutions in paraphrased, plagiarised sentence pairs. Synonymous substitutions can be considered as the substitution of a word or phrase in a sentence such that the overall sense of the sentence remains the same (Section "Synonymous substitution"). From a detection perspective, we classify synonymous substitutions into two different types: contextual substitutions and non-contextual substitutions. These are described as follows:

1 *Contextual Substitutions:* Contextual substitutions can be considered as synonymous substitutions where the left and right contexts of a given word or phrasal substitution are identical in both the source and the paraphrased sentences. For example, given the following fragments (Fig. 4),

   '...with little or...' ↔ '...with minimal or..', the pair 'little ↔ minimal' can be considered as a contextual synonymous substitution. This can be observed as both the left context ('with') and the right context ('or') of the words 'little' and 'minimal' match.
2 *Non-contextual Substitutions:* Non-contextual substitutions can be considered as synonymous substitutions such that their left, right or both contexts may not match. For example (Fig. 4), the text fragments
   'help reuse existing code' ↔ 'existing code to be used again' the phrase pair 'reuse ↔ to be used again' can be considered as a non-contextual synonymous substitution. We consider this pair as non-contextual synonymous substitution since the corresponding left and right contexts do not match.

For the detection of synonymous substitutions, we begin with alignment of sentences. We use the Smith Waterman Algorithm for Plagiarism Detection (Glinos 2014) for the alignment of words in sentences. For sentences having m and n words, the Smith Waterman Algorithm begins by constructing an alignment matrix M of size $(m+1) \times (n+1)$. We construct the scoring scheme for the Smith Waterman Algorithm such that the cost of a match is higher than the cost of a mismatch or gap penalty. In particular, for the scoring equation below, we use the following parameters: `sim(a, b) = 10 (match), -1 (mismatch)` and for the gap penalty, `gap = -1`, stated in the following equation:

Source Fragment/ Paraphrased Fragment	Long	documents	are	poorly	represented	because	they	have	poor	similarity
Longer	Syn.									
documents		exact								
are			exact							
represented				order	order					
poorly				order	order					
because						exact				
the							Syn.			
documents							Syn.			
have								exact		
poor									exact	
similarity										exact

Fig. 8 Detection of Contextual Synonymous Substitutions (Colored squares represent exact match, word reordering and synonymous substitutions)

$$M[i,j] = max \begin{cases} M[i-1,j-1] + \texttt{sim(a, b)}, \\ M[i,j-1] + \texttt{gap}, \\ M[i-1,j] + \texttt{gap}, \\ 0, \quad otherwise. \end{cases} \tag{5}$$

In order to match words or phrases at the beginning or end of the sentence, matching contexts are added at the beginning and end of the sentence. These correspond to sentinel rows and columns which indicate a match.

For the detection of synonymous substitutions, we divide our approaches based on contextual and non-contextual substitutions as follows:

Contextual substitutions

For contextual substitutions, our proposed approach is based on the distributional hypothesis, which states that "*words are similar if their contexts are similar*" (Freitag et al. 2005). Our proposed approach proceeds as follows:

1 We begin with the Smith Waterman alignment matrix with rows corresponding to words from the source sentence and columns corresponding to words from the paraphrased sentence. Furthermore, we also mark matrix elements as 'order' if these have already been identified as a word reordering.
2 Given a word alignment in the matrix form, we consider a given word or a phrase pair as a contextual synonymous substitution, if their left contexts are identical and their right contexts are identical. This can be observed from the Fig. 8 where 'the documents ↔ they' have been marked as synonymous substitutions. In terms of the alignment matrix representation, this is seen by a match in the top-left and bottom-right cells of the synonymous substitution.

Source Fragment/ Paraphrased Fragment	that	it	will	no	longer	be	required
we	0.4243	0.3579	0.3893	0.2068	0.1249	0.2719	0.162
won't	0.4656	0.3971	0.7411	0.4143	0.3113	0.5804	0.2721
need	0.1729	0.0907	0.5027	0.2519	0.2294	0.3283	**0.6346**
a	0.3673	0.3494	0.1216	0.2052	0.1888	0.2346	0.1098
particular	0.3239	0.1282	0.0319	0.1266	0.0747	0.0412	0.188
solution	0.0404	0.0894	0.0603	0.049	0.067	0.1253	-0.0138
anymore	0.4086	0.4587	0.2038	0.5074	**0.5103**	0.286	0.1625

Fig. 9 Detection of Non-contextual synonymous substitutions using ConceptNet Numberbatch pretrained word embeddings

3 Furthermore, sentinel rows and columns in the alignment matrix ensure that matching substitutions in the beginning and end of the sentence are also detected. This can be observed for 'Longer ↔ Long' in Fig. 8 as a contextual synonymous substitution.

Non-contextual substitutions

Non-contextual substitutions can be considered as synonymous substitutions which do not share identical contexts. We use the following methods for the detection of non-contextual synonymous substitutions described here:

1 We use the cosine similarity score of two word vectors (using pretrained word embeddings) for considering a word pair as a non-contextual synonymous substitution. A threshold value for this similarity score is chosen, which is 0.50 in case of ConceptNet Numberbatch pretrained word embeddings. However, we only consider content (non stopwords) only as word cosine similarity scores of stopwords can be quite high. Consideration of stopwords as non-contextual synonymous substitutions may result in a large number of false positives due to the frequency of occurrence of stopwords.
2 Apart from these, a number of non-contextual synonymous substitutions can be considered as derived by punctuation changes to a word, resulting in a word to phrase substitution. For example, the word pair 'subproblem' ↔ 'sub problem' and 'webpage' ↔ 'web page' can be considered as punctuation based non-contextual synonymous substitutions.

Figure 9 gives an example of the detection of non-contextual substitutions using word embedding similarity scores from ConceptNet Numberbatch pretrained word embeddings. It can be seen that the word pairs 'need' ↔ 'required' and 'anymore' ↔ 'longer' have high (≥ 0.500 threshold) similarity values and hence can be considered as non-contextual substitutions. We can also observe high similarity values of stopword pairs such as 'won't' ↔ 'will' and 'won't' ↔ 'be'. Due to high similarity values of stopword pairs, we detect non-contextual synonymous substitutions between content (non stopwords) only.

Table 5 Precision, recall and F_1 scores for word reordering detection

Tasks	Precision	Recall	F_1 score
Task A	0.96111	0.95817	0.95964
Task B	0.57143	1.00000	0.72727
Task C	1.00000	1.00000	1.00000
Task D	0.66667	0.83159	0.74005
Task E	0.99313	0.91146	0.95054
Overall	0.86339	0.95256	0.90579

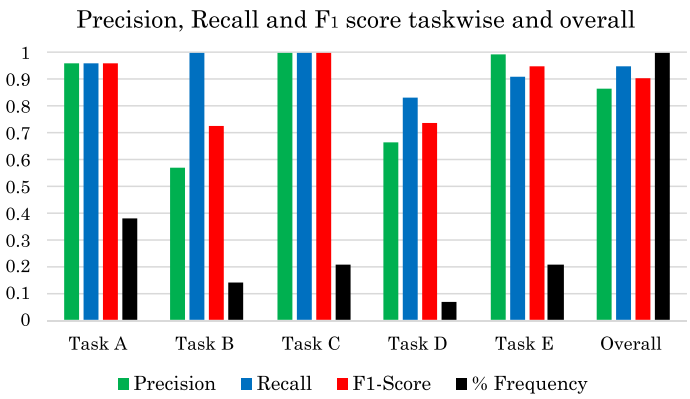


Fig. 10 Precision, Recall and F_1 scores taskwise and overall for detecting word reorderings

This completes the description of our approach for the detection of non-contextual synonymous substitutions.

Results and discussion

In this section we state the results of our proposed approaches for the detection of word reorderings and synonymous substitutions.

Word reordering

Table 5 gives the results of detection of word reorderings in terms of precision, recall and F_1 score. It can be observed that overall F_1 score is 0.905, while task wise the F_1 scores vary from 0.727 to 1.000.

Figure 10 gives the results alongwith the percentage frequency of word reorderings for each task in a bar chart. It can be observed that the precision, recall and F_1 score are generally high for tasks with a higher percentage of instances such as tasks A, C and E. However, for tasks B and D the results are somewhat lower.

While our approach successfully detects a large number of word reorderings correctly, from Table 5 and Fig. 10 we see the precision to be somewhat lower for tasks B and D. This is due to the generation of false positives which can occur due to misalignment of single terms. Let us consider one such false positive case stated below from a heavily revised example of Task B as follows:

1 (Source) Votes cast by pages that are themselves important weigh more heavily and help to make other pages important.
2 (Paraphrased) Expanding on this theory we can then say that the links from important pages are themselves more important.

In the above example, although "pages important ↔ important pages" is a permutation, it does not correspond to a change of order. This is because the last occurrence of important in both sentences should align, leading to "pages important ↔ important pages" being a false positive. In other words, "pages that are themselves important" in the source sentence corresponds to "important pages" in the paraphrased sentence.

From an educational perspective, we observe that our approach of using permutations, paraphrase patterns and string tiling has resulted in high F_1 scores overall and across all of the five tasks. The reasons for this result can be attributed to the nature of word reordering paraphrase type. Our proposed approach detects word reordering where students may (a) reorder words within a phrase, (b) reorder words and insert function words thereby mimicking a paraphrase pattern, or (c) reorder entire phrases within a sentence. Several examples of these types of word reordering can be found within the Corpus of Plagiarised Short Answers, thus providing a plausible explanation for the result.

Comparison with other approaches

To the best of our knowledge this work is a first attempt at defining and proposing an approach for the detection of word reordering paraphrase type. There is a wide variety of definitions available for word reordering in the literature (Barrón-Cedeño et al. 2013; Sousa-Silva 2014; Sun and Yang 2015), hence a direct comparison with other approaches is not possible. Furthermore, most research articles deal with the detection of plagiarism, as opposed to the detection of individual instances of paraphrase types.

Kumar (2014) have proposed a graph based approach for detecting plagiarism specifically in the context of artificial word reordering. Their technique uses a graph representation of word patterns. Their reported detection scores for the Corpus of Plagiarised Short Answers are (Precision = 0.698, Recall = 0.672 and F_1 = 0.674). These scores of represent the challenge of detecting plagiarism in the presence of artificial word reordering.

Synonymous substitutions

For the detection of synonymous substitutions in paraphrased sentence pairs, we present a comparison by varying the alignment method and the pretrained word embeddings. Tables 6, 7 and 8 present the results of using various alignment methods and pretrained word embeddings for the overall dataset and for each task.

From the perspective of alignment methods we have used the Smith Waterman Algorithm for Plagiarism Detection (Glinos 2014). We have also used the Meteor monolingual aligner (Denkowski and Lavie 2014) as an alignment tool and the Semeval-2015 monolingual aligner by Sultan et al. (2014) as another tool for performance comparison. These alignment methods are easily implementable as well as usable and can be readily utilized for paraphrase type identification.

Table 6 Measures (M) of Precision (Pr), Recall (Rc) and F_1 scores for Overall Dataset and Task A

Embedding / Alignment	M	Overall			Task A		
		ConceptNet Numberbatch	FastText	GloVe	ConceptNet Numberbatch	FastText	GloVe
SW Alg.	Pr	**0.76158**	0.68496	0.71786	**0.66757**	0.61279	0.63582
	Rc	**0.84658**	0.82085	0.78619	**0.80403**	0.77855	0.76629
	F_1	**0.80184**	0.74678	0.75047	**0.72947**	0.68580	0.69498
Meteor	Pr	0.74246	0.65843	0.70000	0.64333	0.58004	0.61300
	Rc	0.79774	0.77281	0.74832	0.74151	0.70151	0.70661
	F_1	0.76911	0.71105	0.72335	0.68894	0.63502	0.65648
Sultan	Pr	0.74671	0.66075	0.70408	0.66085	0.58692	0.61435
	Rc	0.75142	0.73845	0.70532	0.67048	0.65437	0.63557
	F_1	0.74906	0.69744	0.70470	0.66563	0.61881	0.62478

Bold values refer to the highest values of F_1 scores of each task

Table 7 Measures (M) of Precision (Pr), Recall (Rc) and F_1 scores for Task B and Task C

Embedding / Alignment	M	Task B			Task C		
		ConceptNet Numberbatch	FastText	GloVe	ConceptNet Numberbatch	FastText	GloVe
SW Alg.	Pr	0.56943	0.46703	0.51105	**0.88001**	0.82606	0.82806
	Rc	0.86546	0.85395	0.72438	**0.89978**	0.87509	0.88744
	F_1	0.68691	0.60382	0.59929	**0.88979**	0.84987	0.85672
Meteor	Pr	**0.63040**	0.49371	0.57982	0.86499	0.80150	0.81956
	Rc	**0.79725**	0.78989	0.67254	0.88104	0.85635	0.86540
	F_1	**0.70408**	0.60763	0.62275	0.87294	0.82802	0.84186
Sultan	Pr	0.66949	0.52127	0.62355	0.81954	0.77723	0.79699
	Rc	0.71955	0.73759	0.60434	0.85668	0.83199	0.84667
	F_1	0.69362	0.61084	0.61380	0.83770	0.80368	0.82108

Bold values refer to the highest values of F_1 scores of each task

Table 8 Measures (M) of Precision (Pr), Recall (Rc) and F_1 scores for Task D and Task E

Embedding / Alignment	M	Task D			Task E		
		ConceptNet Numberbatch	FastText	GloVe	ConceptNet Numberbatch	FastText	GloVe
SW Algo.	Pr	**0.88186**	0.78737	0.81406	**0.76623**	0.70969	0.73386
	Rc	**0.89947**	0.83510	0.83050	**0.76770**	0.76360	0.68662
	F_1	**0.89058**	0.81053	0.82220	**0.76696**	0.73566	0.70945
Meteor	Pr	0.79964	0.72269	0.74560	0.71632	0.64403	0.66710
	Rc	0.83633	0.80790	0.79422	0.71683	0.69614	0.65296
	F_1	0.81757	0.76293	0.76914	0.71657	0.66907	0.65995
Sultan	Pr	0.79733	0.71772	0.74089	0.72821	0.64249	0.67028
	Rc	0.78477	0.76078	0.74266	0.68411	0.67920	0.62313
	F_1	0.79120	0.73862	0.74177	0.70563	0.66034	0.64585

Bold values refer to the highest values of F_1 scores of each task

From the viewpoint of using pretrained word embeddings, we have used ConceptNet Numberbatch 19.04[1] (Speer et al. 2017), FastText[2] (Mikolov et al. 2018) and GloVe[3] (Pennington et al. 2014) pretrained word embeddings. We have used a word similarity threshold of 0.500 as the cutoff score for considering a pair of words as similar. These pretrained word embeddings have proven useful for a variety of NLP tasks such as sentiment analysis and question answering.

Tables 6, 7 and 8 outline the precision, recall and F_1 scores for the overall dataset and for each of the tasks. It can be observed that the choice of the Smith Waterman Algorithm (for Plagiarism Detection) and ConceptNet Numberbatch pretrained word embeddings outperforms all other combinations for almost all of the tasks (except Task B) as well as for the overall dataset. If we consider the performance of these methods on the overall dataset we observe that the Smith Waterman Algorithm with ConceptNet Numberbatch produces an F_1 score of 0.80184, followed by F_1 scores of 0.76911 and 0.74906 using Meteor and Sultan's word aligners. By varying the pretrained word embeddings we observe a gradual reduction in F_1 scores using FastText and then GloVe, as compared to ConceptNet pretrained word embeddings. Furthermore, we observe a high recall (0.84658) using the Smith Waterman Algorithm and ConceptNet Numberbatch.

From a taskwise analysis perspective, we observe highest F_1 scores of 0.72947, 0.70408, 0.88979, 0.89058 and 0.76696 for each of the tasks A, B, C, D and E. In particular F_1 scores for Task B are low for all of the alignment methods and pretrained word embeddings. This is due to the low precision being reported for this task. This is entirely expected as this Task has the highest percentage of sentence pairs in the category of high revision ($23/28 = 82.142\%$) as compared to other tasks. Another point worth observing is that the Meteor monolingual aligner outperforms the Smith Waterman Algorithm and the aligner by Sultan et al. (2014) in terms of F_1 scores for this task due to a higher precision but lower recall.

From an educational perspective, we observe that our approach of using the Smith Waterman Algorithm with ConceptNet Numberbatch pretrained word embeddings produces the best detection score in terms of precision, recall and F_1 scores. The reasons can be attributed to generally well-aligned sentences within the Corpus of Plagiarised Short Answers with students replacing both words and phrases with synonymous substitutions for simulating plagiarism. Furthermore, the application of word cosine similarity using word embeddings for identifying word similarity is an effective approach at finding pairs of similar words in simulated plagiarised text. This is true whether the substitution is a change of form of the same word ('Longer' ↔ 'Long') as shown in Fig. 8 or a synonymous substitution ('need' ↔ 'required') as shown in Fig. 9.

In summary, we observe that the combination of pretrained word embeddings and alignment methods produces a high detection of paraphrase types for plagiarised, paraphrased sentence pairs. This coupled with the ease of implementation and use with which these methods can be applied gives rise to an opportunity for enriching plagiarism

[1] http://github.com/commonsense/conceptnet-numberbatch (numberbatch-en-19.08.txt.gz)

[2] http://fasttext.cc (wiki-news-300d-1M.vec)

[3] http://nlp.stanford.edu/projects/glove (glove.6B.zip)

detection methods. Such an addition may result in additional information (paraphrase types) being detected, which may prove useful for a human evaluator in making an informed decision about the actual occurrence of plagiarism.

Conclusions and future work

In this work we proposed methods to identify paraphrase types in paraphrased, plagiarised sentence pairs. Several contributions have been presented in this paper outlined here. The proposed idea of this paper, i.e. methods to detect paraphrase types for plagiarism detection complements several research papers that propose paraphrase types and their frequency in plagiarised text. We also proposed methods to identify word reorderings using permutations and paraphrase patterns which has not been presented in earlier work. For the detection of synonymous substitutions, our proposed method of using the Smith Waterman Algorithm and ConceptNet Numberbatch pretrained word embeddings outperformed other combinations of alignment methods and word embeddings.

This research can be used to enhance existing plagiarism detection methods and systems by incorporating methods to detect paraphrase types for plagiarism detection. Such an addition would provide valuable information to a human evaluator in making an informed decision about the actual occurrence of plagiarism.

For future work, methods to detect other paraphrase types can be proposed. In particular methods to detect the insertion/deletion paraphrase type can provide an interesting addition to the proposed collection of methods. Furthermore, an integrated framework for the detection of a multitude of paraphrase types can be designed which will serve to integrate various approaches for the detection of paraphrase types.

It is also worthwhile to have a broader view of the implications of this research from a wider education perspective. This can be initiated by considering the wider concept of academic integrity which encompasses among other aspects, plagiarism detection. Bretag (2018) identifies Academic Integrity as "*an interdisciplinary concept that provides the foundation for every aspect and all levels of education*". Academic integrity is based on the values of honesty, trust, fairness, respect, responsibility, and courage as outlined by the International Center for Academic Integrity (International Center for Academic Integrity 2021). The current research and its focus on plagiarism detection provides support for building on the values of honesty, fairness and trust in the pursuit of academic integrity.

The implications of this research on the wider academic community such as teachers and researchers are manifold. From the perspective of teachers, it provides additional support for the detection of plagiarism by highlighting paraphrase types, thereby assisting in the detection of plagiarism. This aspect can also be used for the promotion of originality by educating students on methods of paraphrasing. From the viewpoint of researchers, data on paraphrase types and their frequencies from the current research as well as from past research works (Barrón-Cedeño et al. 2013; Sun and Yang 2015) provides valuable insights into paraphrase types used in plagiarism.

Although English is the language mostly used worldwide, the findings of this research can be extended to languages other than English. Kopotev et al. (2021) provides an excellent overview of plagiarism and its detection in the Russian Language. Cross Language Plagiarism Detection (CLPD) (Foltýnek et al. 2019) is a widely researched area of

plagiarism detection where the objective is to detect plagiarism from a wide range of multilingual resources (Potthast et al. 2011). Our proposed research methods based on textual alignment and word embeddings can naturally be extended to other languages, since alignment methods have a strong foundation in a multilingual context (Tiedemann 2011). Furthermore word embeddings for other languages (Wang et al. 2020) can be utilized for the detection of paraphrase types for multiple languages.

It is also important to emphasize on the limitations of current research. The current research with its emphasis on paraphrase type identification in the context of plagiarism detection might have limitations in cases of contract cheating or ghostwriting (Meuschke and Gipp 2013). This is because in contract cheating, a plagiarist utilises the services of an external entity for generating academic content. In cases where the external entity writes completely new content, paraphrase type identification will have negligible affect in assisting in the detection of plagiarism.

In summary, we can conclude that the proposed methods of paraphrase type identification in this research can have a wide variety of applications in the academic context. This includes not only assistance in plagiarism detection but also emphasis on enforcing good academic practice.

Acknowledgements
Not applicable.

Authors' contributions
This research is an extension of the dissertation research by the corresponding author F-A. M-S and P-C have supervised the research. All authors read and approved the final manuscript.

Authors' information
The corresponding author F-A is a lecturer at the Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. He completed his PhD in August 2020 supervised by M-S and co-supervised by P-C. M-S is a Senior Lecturer at the Department of Computer Science, University of Sheffield, United Kingdom. P-C is a Professor at the Information School, University of Sheffield, United Kingdom.

Funding
The authors did not receive any funding for this research.

Availability of data and materials
The datasets used and analysed in this research are available from the corresponding author upon reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. [2]Department of Computer Science, University of Sheffield, Sheffield S10 2TN, United Kingdom. [3]Information School, University of Sheffield, Sheffield S10 2TN, United Kingdom.

Received: 5 March 2021 Accepted: 23 May 2021
Published online: 04 August 2021

References

Alvi, F., El-Alfy, E. S. M,. Al-Khatib, W. G., & Abdel-Aal, R. E. (2012). Analysis and Extraction of Sentence-Level Paraphrase Sub-Corpus in CS Education. In *Proceedings of the 2012 ACM Conference of Special Interest Group on IT Education (SIGITE), Association of Computing Machinery*, pp 49–54.

Alzahrani, S. M., Salim, N., & Abraham, A. (2012). Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 42*(2), 133–149.

Barrón-Cedeño, A. (2012). On the Mono- and Cross-Language Detection of Text Re-use and Plagiarism. PhD thesis, Universitat Polytecnica De Valencia.

Barrón-Cedeño, A., Vila, M., Martí, M. A., & Rosso, P. (2013). Plagiarism meets paraphrasing: insights for the next generation in automatic plagiarism detection. *Computational Linguistics, 39*(4), 917–947.

Bensalem, I., Rosso, P., & Chikhi, S. (2019). On the use of character n-grams as the only intrinsic evidence of plagiarism. *Language Resources and Evaluation, 53*(3), 363–396.
Bhagat, R. (2009). Learning paraphrases from text. PhD thesis, University of Southern California.
Bhagat, R., & Hovy, E. H. (2013). What is a paraphrase? *Computational Linguistics, 39*(3), 463–472.
Bisazza, A., & Federico, M. (2016). A survey of word reordering in statistical machine translation: computational models and language phenomena. *Computational Linguistics, 42*(2), 163–205.
Bretag, T. (2018). Academic integrity. In *Oxford Research Encyclopedia of Business and Management*, Oxford University Press.
Carmona, M. Á. Á., Franco-Salvador, M., Villatoro-Tello, E., Montes-y-Gómez, M., Rosso, P., & Pineda, L. V. (2018). Semantically-informed distance and similarity measures for paraphrase plagiarism identification. *Journal of Intelligent and Fuzzy Systems, 34*(5), 2983–2990.
Chitra, A., & Rajkumar, A. (2016). Plagiarism detection using machine learning-based paraphrase recognizer. *Journal of Intelligent Systems, 25*(3), 351–359.
Chong, M. (2013). A Study on Plagiarism Detection and Plagiarism Direction Identification using Natural Language Processing Techniques. PhD thesis, University of Wolverhampton.
Clough, P. (2010). Measuring text reuse in the news industry. In: L. Bently , J. Davis & J. C. Ginsburg (Eds.), (pp. 247–259). Cambridge University Press: Copyright and Piracy.
Clough, P., & Stevenson, M. (2011). Developing a corpus of plagiarised short answers. *Language Resources and Evaluation, 45*(1), 5–24.
Denkowski, M., & Lavie, A. (2014). Meteor Universal: language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, pp 376–380.
Dias, P. C., & Bastos, A. S. C. (2014). Plagiarism phenomenon in European Countries: results from GENIUS project. *Procedia-Social and Behavioral Sciences, 116,* 2526–2531.
Dolan, B., Quirk, C., & Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics, Association for Computational Linguistics*.
Dolan, W. B., & Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005), Asia Federation of Natural Language Processing*.
Fatima, A., Abbas, A., Ming, W., Hosseini, S., & Zhu, D. (2019). Internal and external factors of plagiarism: evidence from Chinese public sector universities. *Accountability in Research, 26*(1), 1–16. https://doi.org/10.1080/08989621.2018.1552834.
Foltýnek, T., Meuschke, N., & Gipp, B. (2019). Academic plagiarism detection: a systematic literature review. *ACM Computing Surveys, 52*(6), 1–42. https://doi.org/10.1145/3345317.
Foltỳnek, T., Dlabolová, D., Anohina-Naumeca, A., Razı, S., Kravjar, J., Kamzola, L., et al. (2020). Testing of support tools for plagiarism detection. *International Journal of Educational Technology in Higher Education, 17*(46).
Freitag, D., Blume, M., Byrnes, J., Chow, E., Kapadiam, S., Rohwer, R., & Wang, Z. (2005). New Experiments in Distributional Representations of Synonymy. In *Proceedings of the Ninth Conference on Computational Natural Language Learning, Association for Computational Linguistics, Stroudsburg, PA, USA, CONLL '05*, pp 25–32.
Ganitkevich, J., Durme, B. V., & Callison-Burch, C. (2013). PPDB: The paraphrase database. In *Proceedings of the Human Language Technology Conference (HLT) 2013, North American Chapter of the Association for Computational Linguistics*, (pp 758–764).
Glinos, D. G. (2014). Discovering Similar Passages within Large Text Documents. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction - 5th International Conference of the CLEF Initiative, CLEF 2014, Sheffield*, UK, pp 98–109.
International Center for Academic Integrity (2021) The Fundamental Values of Academic Integrity, 3rd Edition. https://www.academicintegrity.org/the-fundamental-values-of-academic-integrity/, Accessed May 2021.
Kanjirangat, V., & Gupta, D. (2016). Study on extrinsic text plagiarism detection techniques and tools. *Journal of Engineering Science & Technology Review, 9*(5), 9–23.
Kanjirangat, V., & Gupta, D. (2018). Unmasking text plagiarism using syntactic-semantic based natural language processing techniques: comparisons, analysis and challenges. *Information Processing & Management, 54*(3), 408–432.
Kauffman, Y., & Young, M. F. (2015). Digital plagiarism: an experimental study of the effect of instructional goals and copy-and-Paste affordance. *Computers & Education, 83,* 44–56.
Kopotev, M., Rostovtsev, A., & Sokolov, M. (2021). Shifting the norm: the case of academic plagiarism detection. *The Palgrave Handbook of Digital Russia Studies* (pp. 483–500). Cham: Palgrave Macmillan.
Kumar, N. (2014). A graph based automatic plagiarism detection technique to handle artificial word reordering and paraphrasing. In *International Conference on Intelligent Text Processing and Computational Linguistics, Springer International Publishing*, (pp 481–494).
Madnani, N., Tetreault, J., & Chodorow, M. (2012). Re-examining machine translation metrics for paraphrase identification. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, (pp 182–190).
Maurer, H. A., Kappe, F., & Zaka, B. (2006). Plagiarism-a survey. *Journal of Universal Computer Science, 12*(8), 1050–1084.
McKeever, L. (2006). Online plagiarism detection services - saviour or scourge? *Assessment & Evaluation in Higher Education, 31*(2), 155–165.
Meuschke, N., & Gipp, B. (2013). State-of-the-art in detecting academic plagiarism. *International Journal for Educational Integrity, 9*(1), 50–71.
Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2018). Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, (pp 52–55).
Moritz, M., Hellrich, J., Büchel, S. (2018). A method for human-interpretable paraphrasticality prediction. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, (pp 113–118).
Mphahlele, A., & McKenna, S. (2019). The use of turnitin in the higher education sector: decoding the myth. *Assessment & Evaluation in Higher Education, 44*(7), 1079–1089.

Nichols, L., Dewey, K., Emre, M., Chen, S., & Hardekopf, B. (2019). Syntax-based improvements to plagiarism detectors and their evaluations. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education, Association of Computing Machinery*.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), vol 14, pp 1532–1543.

Potthast, M., Barrón-Cedeno, A., Stein, B., & Rosso, P. (2011). Cross-language plagiarism detection. *Language Resources and Evaluation, 45*(1), 45–62.

Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., & Stein, B. (2014), Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction, Springer International Publishing*, (pp 268–299)

Potthast, M., Goering, S., Rosso, P., & Stein, B. (2015). Towards data submissions for shared tasks: first experiences for the task of text alignment. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11*, 2015.

Sanchez-Perez, M. (2018). Plagiarism detection through paraphrase recognition. PhD thesis, Instituto Politécnico Nacional, Mexico.

Sanchez-Perez, M., Sidorov, G., & Gelbukh, A. (2014). A winning approach to text alignment for text reuse detection at PAN 2014 – Notebook for PAN at CLEF 2014. Working Notes for CLEF 2014 Conference, Sheffield, UK pp 1004–1011.

Sánchez-Vega, F., Villatoro-Tello, E., Montes-y Gómez, M., Rosso, P., Stamatatos, E., & Villaseñor-Pineda, L. (2017). Paraphrase plagiarism identification with character-level features. Pattern Analysis and Applications pp 669–681.

Schmidt Hanbidge, A., Tin, T., & Tsang, H. (2020). Academic integrity matters: successful learning with mobile technology. In *International Conference on Interactive Collaborative Learning, Springer International Publishing*, (pp 966–977).

Sousa-Silva, R. (2014). Investigating academic plagiarism: a forensic linguistics approach to plagiarism detection. *International Journal for Educational Integrity, 10*(1), 31–41.

Speer, R., & Lowry-Duda, J. (2017). ConceptNet at SemEval-2017 Task 2: extending word embeddings with multilingual relational knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics*.

Speer, R., Chin, J., & Havasi, C. (2017), ConceptNet 5.5: an open multilingual graph of general knowledge. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4–9. (2017). *San Francisco* (pp. 4444–4451). USA: California.

Sultan, M. A., Bethard, S., & Sumner, T. (2014). Back to basics for monolingual alignment: exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics, 2,* 219–230.

Sun, Y. C., & Yang, F. Y. (2015). Uncovering published authors' text-borrowing practices: paraphrasing strategies, sources, and self-plagiarism. Journal of English for Academic Purposes. pp. 224–236.

Tiedemann, J. (2011). Bitext alignment. *Synthesis Lectures on Human Language Technologies, 4*(2), 1–165.

Vila, M., Martí, M. A., Rodríguez, H., et al. (2014). Is this a paraphrase? what kind? paraphrase boundaries and typology. *Open Journal of Modern Linguistics, 4*(01), 205–218.

Wang, X., Chen, Y.Y., Zhao, H., Lu, B.L. (2013). Labeled alignment for recognizing textual entailment. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP) 2013, Asian Federation of Natural Language Processing*, (pp 605–613).

Wang, Y., Hou, Y., Che, W., & Liu, T. (2020). From static to dynamic word representations: a survey. International Journal of Machine Learning and Cybernetics pp 1–20.

Weber-Wulff, D. (2014). Plagiarism and academic misconduct. *False Feathers: A Perspective on Academic Plagiarism* (pp. 3–27). Berlin Heidelberg: Springer.

Wise, M. J. (1995). Neweyes: a system for comparing biological sequences using the running Karp-Rabin greedy string-tiling algorithm. In*Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology, Cambridge, United Kingdom, July 16-19, 1995*, (pp 393–401).

Zhao, S., Wang, H., Liu, T., Li, S. (2008). Pivot approach for extracting paraphrase patterns from bilingual corpora. In *Proceedings of the Human Language Technology Conference (HLT) 2008, Association for Computational Linguistics*, (pp 780–788).

Publisher's Note