Paraphrase Generation and Identification at Paragraph-Level

Arwa Al Saqaabi [1][0009-0009-6431-0642], Craig Stewart [2], Eleni Akrida[3], and Alexandra.I Cristea[4]

[1] Qassim University, Buraydah, 52571, SA
A.AlSaqabi@qu.edu.sa
[1,2,3,4] University of Durham, Durham. DH1 3LE, Durham, UK
craig.d.stewart@durham.ac.uk
eleni.akrida@durham.ac.uk
alexandra.i.cristea@durham.ac.uk

Abstract. The availability and growth of tools and natural language generation (NLG) models that are used to paraphrase text could be helping to improve students' writing and comprehension skills or a threat to intellectual property and educational integrity specifically when the text has been copied from other authors. These tools can be used by plagiarists to paraphrase individual words, phrases, sentences, and paragraphs. To solve this issue, much work has been done on plagiarism detection (PD) and paraphrase identification (PI) utilising downstream tasks and natural language processing (NLP) methods. These works mainly focus on sentence length and sentence-level paraphrasing. In this paper, we investigate paragraph-length and paragraph-level paraphrasing as the most common method of committing plagiarism is copying and paraphrasing paragraphs from other authors. Here, we construct a novel, large-scale paragraph-level paraphrasing dataset by implementing and examining a state-of-the-art Transformer-based model to reorder and paraphrase sentences without affecting a paragraph's meaning. In a first-of-a-kind study, we consider both intra-sentence and inter-sentence similarity before examining the efficiency of state-of-the-art Transformer-based models in detecting paraphrased paragraphs. We offer a technique that serves as both a tool for honing paraphrasing skills and a means of identifying plagiarism. Our outcomes surpass those presented in the existing literature.

Keywords. Natural Language Generation (NLG), Natural Language Processing (NLP), Paragraph-Level Paraphrasing, Transformer-Based Model

1 Introduction

Paraphrase generation is a commonly studied NLG task. On the one hand, such paraphrased text could be used to enhance plagiarism detection, machine translation, and summarisation for NLP downstream tasks. In addition, paraphrasing can be employed to assist comprehension and writing skills development. On the other hand, paraphrase generation could undermine academic integrity if it is misused by students seeking to plagiarise existing work. According to [1] the current state of artificial intelligence (AI)

models makes it possible to create highly coherent and contextually suitable paraphrased material that might be used to generate plagiarised content. In addition, [2] concluded that it is difficult to differentiate artificially paraphrased text from human-written text. Thus, AI models have the potential to be utilized both for improving students' writing skills and simultaneously detecting plagiarism.
Paraphrase generation is the task of generating an output text that preserves the meaning of the input text in other forms of text [3]. Current state-of-the-art research focuses on paraphrasing texts at the sentence-level [[4],[5]] and paragraph-level [6] but utilises sentence-level paraphrasing methods only. These approaches consider the meaning of each sentence independently; they did not determine any semantic relationships between sentences. The novel research presented in this paper paraphrases paragraphs utilising paragraph-level paraphrasing by considering both the intra-sentence and inter-sentence relationships by implementing paraphrase generation Transformer-based models. This is harder and more valuable than sentence-level paraphrasing because it considers the diversity across multiple sentences beyond the lexical and syntactic diversity of a single sentence. This holds practical significance as it is a necessary skill that needs to be cultivated and applied in educational tasks, such as citing the work of others. In addition, according to [7], plagiarists reuse paragraphs not sentences the most frequently.

Paragraph-level paraphrasing includes sentence reordering, sentence splitting, and sentence merging. The initial work in this area was presented in[8], where the authors applied an algorithm to detect paraphrasing (focusing on the paragraph level); however, this work was limited by the fact that very few suitable datasets are available for this type of research. As there are no published paragraph-level paraphrase datasets established using paraphrase generation Transformer-based models, we implement two algorithms based on state-of-the-art Transformer-based models that have become the standard paradigm for most NLG tasks. We perform sentence reordering considering inter-sentence diversity before paraphrasing the paragraphs using state-of-the-art paraphrase generation models. Specifically, we apply the Sentence Order Prediction (SOP) of the ALBERT [9] re-training model and Transformer-based models **(**BERT [10], RoBERTa [11] and Longformer [12] for paraphrasing. The output paragraphs are generated based on the semantic relations among the source sentences. We generate multiple paraphrased versions for each source, making our approach effective for improving students' writing abilities. In this work, this dataset (ALECS) enables us to investigate the Transformer-based models' ability to distinguish between the source and paraphrased text after reordering and paraphrasing its sentences using a variety of levels within the masked language model (MLM). This research aims to investigate the following research question (RQ):

- RQ: How efficiently can state-of-the-art Transformer-based models discriminate between the original and paraphrased text at the paragraph-level with sentence reordering?

To the best of our knowledge, this study provides the first extensive dataset of paragraphs (ALECS) that have been paraphrased at the paragraph-level using Transformer-

based models along with a study of Transformer-based models' performance in detecting paragraph-level paraphrasing.

The remainder of this paper is organised as follows: Section 2 outlines the main related endeavors. Section 3 details our methodology, while Section 4 contains the dataset evaluation. The experimental results and discussion are reported in Section 5. Section 6 summarises the main conclusions and future research directions.

2 Related Work

PI is a main task in NLP that is involved in many downstream tasks such as PD [13] and data augmentation[14]. It is considered a classification task which can be performed at the sub-sentence level, sentence-level, paragraph-level, or document-level. According to[15]:

- Sub-sentence level: the algorithm finds the pertinent sub-expression categories that are contained within a sentence.
- Sentence-level: The appropriate categories of a single sentence are obtained.
- Paragraph-level: A single paragraph's relevant categories are retrieved by the algorithm.
- Document-level: The algorithm uses the entire document to extract the relevant categories.

In this research, we focus on sentence-level and paragraph-level PI.

Most of the existing work has been done at the sub-sentence-level or sentence-level by applying machine learning classification algorithms on hand-crafted features such as syntactic dependency features [14] and lexical features from a Bag of Words[16]. Other works used neural networks that focus on word embedding[17], recurrent neural networks, , or Transformer-based models[10].

These works were implemented on sentence-length datasets such as the Microsoft Research Paraphrase Corpus (MRPC), PAN, and Quora Question Pairs (QQP). A total of 5801 pairs in the MRPC corpus have been manually tagged as paraphrases or non-paraphrases [18]. MRPC was collected from online news collection by using heuristics to identify candidate document pairs and candidate sentences from the documents. The PAN datasets include cases that were obfuscated using elementary automated techniques that did not preserve the text's intended meaning. These heuristics include, for instance, randomly deleting, adding, or changing words or phrases, as well as exchanging words with randomly chosen synonyms, antonyms, hyponyms, or hypernyms [19]. In addition, in QQP[1] question titles from the forum are divided into duplicate-or-not questions. These questions are published on a website where users can post questions and receive answers. The designers of the enormous QQP dataset state that, despite it containing labels made by humans, the labels were not intended to be used for PI tasks.

These datasets have a limitation on their size which makes training neural or Transformer-based models difficult. To solve this limitation, many datasets have been

[1] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

created using a variety of techniques. PARADE [6] created computer science concepts from online user-generated flashcards. They implemented clustering to group each specific term's definitions. They then selected one as the source and the other one as a paraphrased text. A four-label system was used to manually annotate each extracted sample. In addition, [5] created a dataset of sentence-level paraphrasing that was generated by machine translation. They translated the text to another language (Czech) and then translated it back to the original text language (English). The quality of the paraphrased text is affected by the efficiency of the translation model used. Despite the differences in style and content quality of the above-mentioned datasets, they all consist of sentence-level paraphrasing text which is not suited for paraphrase identification at the paragraph-level paraphrasing. They mainly apply different algorithms to paraphrase each sentence independently as a result all the works have been done on these datasets focusing on sentence-level. This type of paraphrasing is less common among plagiarists as they tend to paraphrase a paragraph by using sentence reordering, splitting and/or merging with consideration of the paragraph's meaning [20].

Nowadays, paragraph paraphrasing and classification has become possible, especially with Transformer-based models that accept a long input of tokens. However, few studies have considered a paragraph as an input for PI; [20] artificially constructed a dataset using content from Wikipedia, theses, and arXiv articles by paraphrasing text using Transformer-based models. They also applied state-of-the-art Transformer-based models to distinguish between source and paraphrased text. They achieved commendable results. The main difference between this work and ours is that they directly paraphrased text without reordering sentences as we do to achieve document-level paraphrasing in the future.

Two works, [21][22] considered paragraph-level paraphrasing by applied sentence reordering after paraphrasing the text through back-translation. However, auto-translation for paraphrasing text may still cause errors where a word is translated into a synonym which may not be contextually valid [23]. In [21][22] graph models are implemented to generate the best order of the sentences based on the paraphrased text ignoring the relation of the source's sentences. Thus, the sentences semantic relations will be affected by the quality of the paraphrasing algorithm used. In addition, they have not yet applied the paraphrase identification method to these datasets.

In our work, we aim to avoid this limitation by using SOP on the original text to generate two different sentence orders for each paragraph based on the source text's inter-sentence similarity and intra-sentence similarity; then, we produce paraphrased paragraphs using a state-of-the-art Transformer-based model to achieve paraphrase generation. To the best of our knowledge, this is the first dataset for training PI classification models that consists of paragraph-level paraphrasing utilising Transformer-based models.

3 Methodology

3.1 Dataset Creation

Our dataset, ALECS, contains text relating to social science domains collected from Wikipedia[2]. We eliminate linguistics articles because of the manner that is used for paraphrasing, and law articles as models appropriate for such articles would need to be trained on legalistic language specifically. The major goal of this step is to create a dataset that can be used to enhance the students' writing skills and distinguish between human-written and machine-paraphrased texts in order to identify plagiarism in academic writing. The total number of paragraphs is 391,205 after filtering the collected text into paragraphs of 50–151 words in length (this is the average paragraph length in English[24]) with at least three sentences. This minimum sentence requirement was put in place as our paraphrased text creation methodology (see below) utilises inter-sentence semantics, and therefore, any fewer than three sentences would have rendered the paraphrased text more akin to texts created using sentence-level methods used by other datasets (see Figure 1).

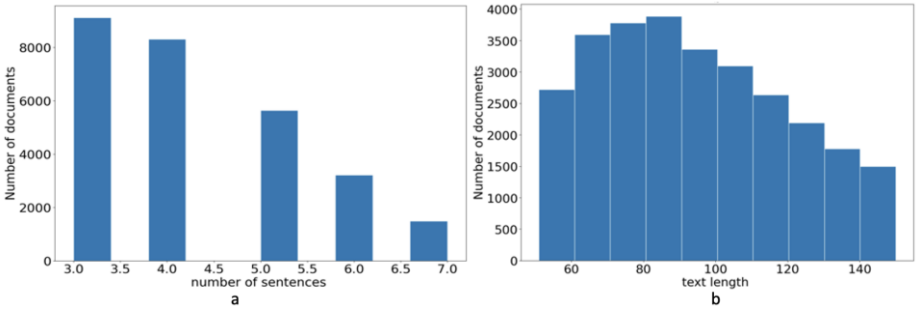

Fig. 1. a) Number of samples containing a specific number of sentences. b) Number of samples in relation to the number of words in the documents.

Inter-sentence paragraph coherence score (sentences reordering). Text coherence has been the subject of a lot of research; coherence is described in different terms such as *entity* [25] and *word co-occurrence[26]*. In this work, we implement the SOP of the ALBERT Transformer-based model as it considers inter-sentence coherence and generates a coherence score that represents the validity of the order between two sentences [9]. This model outperforms other Transformer-based models in terms of paragraph coherence [27].

Firstly, we convert each paragraph D into a fully connected directed graph G where the set of sentences S serve as nodes:

$$V(G) = S_1, S_2, \ldots, S_n \tag{1}$$

[2] https://en.wikipedia.org/wiki/Social_science

$$P_{SOP}(S_s^i, S_s^j) = \begin{cases} P \geq \varepsilon, & \text{i} \neq \text{j} \\ 0, & i = j \end{cases} \quad (2)$$

Then, we apply two algorithms to reorder the paragraph's sentences depending on the SOP probability. Each algorithm suggests a path that passes over all nodes without repeating and has the highest coherence score based on a coherence measurement approach of the algorithm used. The paragraph's sentences are shuffled based on the suggested path before being evaluated by human evaluators (discussed in Section 4) to determine which algorithm is the best and investigate the correlation between the human-written paragraphs and generated paragraphs.

Inter-sentence shuffling. Assume that G is a fully connected directed graph where the nodes are the document's sentences, and the weight of the edges is measured by the SOP probability. The generated path must pass through each node without repeating.

– Algorithm SALAC1

This algorithm gives priority to the nodes that are linked by the strength coherence score that are represented as edges' weights in the graph. SALAC1 determines the order of sentences depending on many conditions that are shown in the flowchart in Figure 2. Let us assume that we have a paragraph consisting of four sentences and its graph matrix shown in Figure 3; the strength coherence score is 0.7 which is linked S1 to S2 and S2 to S4. This means S1 and S2 must come before S4 although we could insert other sentences between them without breaking down their relation. Moreover, the weakest coherence score in this example is 0.4 and the rest of the coherence scores are distributed between the strongest and weakest scores. We replace the matrix diameter values with 0 to remove the path from a sentence to itself.

The first step as shown in the flowchart (Figure 2) considers only sentences that are linked with the highest coherence scores, which are in the example S1-S2-S4. Then, SALAC1 checks if the path is completed or if there are unincluded sentences. Then it removes all the coherence scores that are considered in the previous step leading to a decrease in the highest coherence score from 0.7 to 0.6 in this example. Now SALAC1 selects all sentences that have the highest coherence score, then it inserts them in the path depending on their relations to the sentences already in the path considering their relations to each other (parent or child). In some cases, the sentence has the same coherence score as all the nodes in the graph, which means it could be at any position on the path. In the example (Figure 3), S3 links to all sentences with a coherence score of 0.5 so we can locate it at the end of the path S1-S2-S4-S3.

Another condition can be seen in the flowchart in Figure 2: when there is a sentence with a strong link to the second sentence in the path while its relation to the first sentence is weak; in this case, this sentence is a parent to the second sentence (i.e., it should be inserted before the second sentence) but a child to the first one (i.e., it must come after it).

– *Algorithm SALAC2.*

SALAC2 goes over all possible paths in the graph and picks a path with the best

coherence score. It calculates the path coherence between a parent node to its child node using Equation 3.

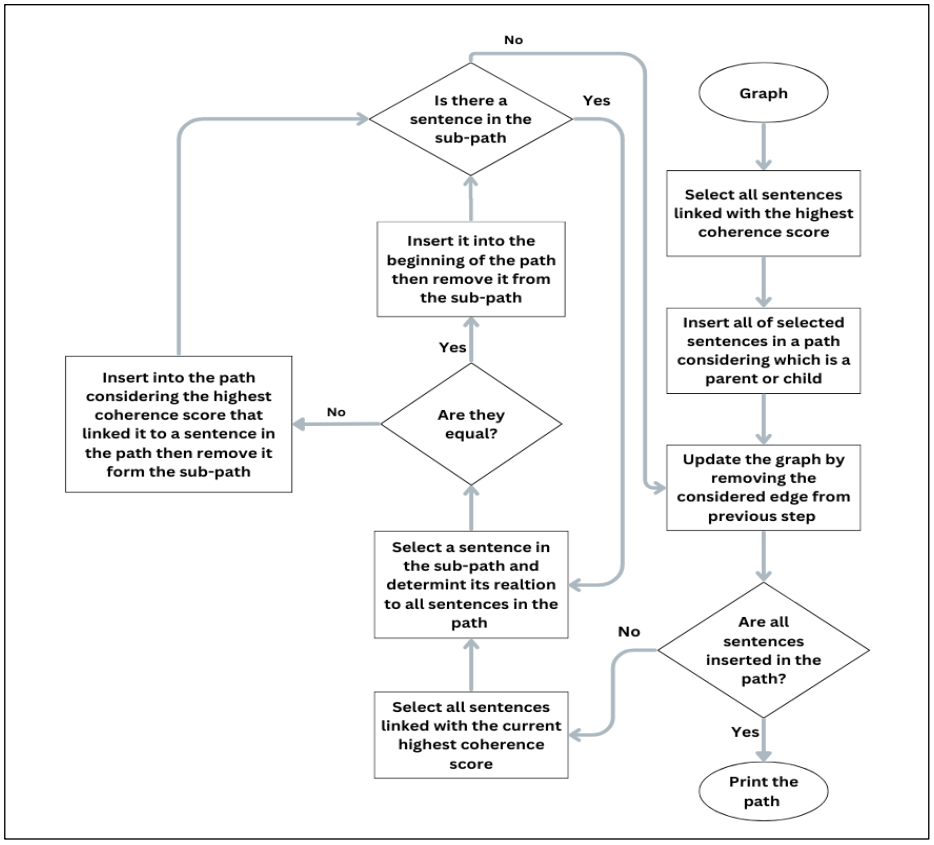

Fig. 2. SALAC1 flowchart algorithm.

	Sentence S1	Sentence S2	Sentence S3	Sentence 4
Sentence S1	0	**0.7**	0.6	0.5
Sentence S2	0.4	0	0.6	**0.7**
Sentence S3	0.6	0.6	0	0.6
Sentence S4	0.4	0.5	0.6	0

Fig. 3. SALAC 1 graph matrix example, scores in bold represent the strength coherence score while underlined scores represent the weakest coherence score.

By implementing these algorithms, we generate for each source paragraph two paragraphs with different sentence orders compared to the source. Then, we calculate the paragraph's coherence score by implementing Equation 3.

$$COH = \sum_{i}^{n-1} \sum_{j=i+1}^{n} P_{SOP}\left(S_s^i, S_s^j\right) \quad (3)$$

Intra-sentence masking (paraphrasing). In an effort to develop a paragraph-level dataset, we implement three state-of-the-art Transformer-based models to paraphrase paragraphs after applying the sentence shuffling algorithms. For paraphrasing, we implement BERT [10], which is mostly used as a baseline in NLG research, RoBERTa [11], which is built on BERT to handle longer documents, and Longformer [12], which is mainly developed for long documents. To account for the diversity of our dataset, we apply a variety of levels of the masked language model (MLM) for all three Transformer-based models. It masks a part of the words from a sequence of input or sentences and requires the designed model to predict the most likely word choices to complete the sentence. To avoid producing false information compared to the source, we exclude named entities and punctuation, such as brackets, digits, currency symbols, and quotation marks from paraphrasing as in [20].

4 Evaluation

4.1 Human Evaluation

To demonstrate the efficacy of the task, we perform a manual evaluation study, which is the most common approach in NLG. In the next paragraphs, we explain the human evaluation method applied to this study.

Firstly, based on the task and goal of this study, we must evaluate the quality of the output text of our algorithms by collecting and analysing numerical data. To achieve this, we implement a quantitative study as an intrinsic approach. In more detail, we measure the differences in the text semantics by comparing the generated paragraph to the source. Both documents (source and output) should convey a similar meaning, that is, we aim to maintain the meaning of the source by reordering the paragraph's sentences.

Secondly, we randomly sample 100 paragraphs from the dataset. In NLG, the median number of samples used for human evaluation is 100 [28]. In addition, three evaluators check each sample, and the decision on whether the source and the output were similar in meaning is taken based on majority voting. In terms of the human assessors, six highly educated fluent speakers are selected as the text used in this study is extracted from Wikipedia articles written for a general readership.

The participants are provided with the source texts and the reconstructed paragraphs for each sample, then they are asked to evaluate each of the generated paragraphs in terms of semantic similarity to the source. According to[28], complex concepts cannot be captured in a single arbitrary rating [19], therefore the participants are asked to select a score on a 5-point Likert scale where each score represents a defined value as follows:

5: Almost identical
4: Very similar, with only minor changes to the meaning
3: Similar, with major changes to the meaning
2: Dissimilar, with significant changes to the meaning

1: Extremely different

The experiment was approved by the University's ethics committee and took about three hours to complete.

4.2 Automatic Evaluation

Inter-annotator agreement (IAA) correlation. Inter-annotator correlation or agreement (IAA) determines the degree of agreement between the evaluations of different raters. It is commonly used when using multiple annotators. According to [29], the acceptable range of IAA is between 0.3–0.5 in NLG research where the higher the IAA, the more valuable.

In this work, we implement the kappa coefficient as an IAA statistical test; this involves two groups of three evaluators, with each group evaluating 50 samples of different generated texts. The results for both groups of evaluators show a low correlation, namely 0.4. In [28], the authors explained that IAA is more likely to be low when measuring a complex language concept such as semantic similarity as in this study. However, this score reaches an outstanding correlation of 0.8 after categorising the rating scores into two categories depending on their differentiations (5,4,3 = A, 2,1 = B).

Efficiency of the Algorithms. We compare the algorithms' efficiency in relation to the human evaluation results. A total of 300 scores were given by the evaluators for each algorithm. The results in Table 1 show that SALAC1 and SALAC2 are comparable. SALAC1 and SALAC2 generate paragraphs with identical meanings to the source by 39% and 40%, respectively. In contrast, the percentage of samples that have different meanings to the source is very low in all algorithms' results, 1% and 3%, respectively. The difference between SALAC1 and SALAC2 can be noticed in similar and dissimilar samples. To make it clear, we categorise the scores depending on their definition (Table 1 grey columns). SALAC1 is higher by 6% in similar samples and lower by 6% for dissimilar samples compared to SALAC2.

Table 1. Distribution of 300 votes to the scores given by humans.

Score	1	2	3	4	5	1, 2	3, 4, 5
SALAC1	1%	9%	27%	24%	39%	10%	90%
SALAC2	3%	13%	21%	23%	40%	16%	84%

Correlation between the paraphrased paragraph's similarity score and the human-written paragraph's similarity score. Measuring the correlation between the paraphrased paragraph's similarity score and the human-written paragraph's similarity score is important as we try to generate a paragraph-level paraphrased text based on the human-written paragraph. To achieve this objective, we apply Equation 3 to measure the coherence score on source paragraphs. We then compare it to the generated paragraph's coherence score for each algorithm.

In Figure 4, SALAC1 and SALAC2 provide high Pearson's correlation values,

namely 0.89 and 0.80, respectively. This high correlation indicates that the implemented algorithms maintain the original text's semantics.

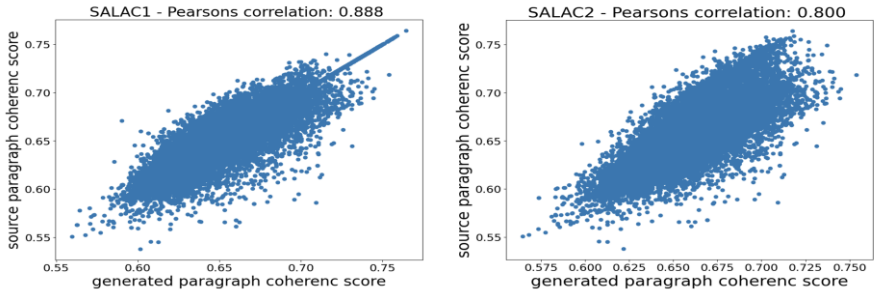

Fig. 4. Correlation of the generated paragraphs to the human-written paragraphs.

Mask applied method. We apply Transformer-based models (BERT, RoBERTa, Longformer) to paraphrase the paragraphs generated by the SALAC algorithms with 0.15, 0.20, and 0.30 MLM. Thus, we have six paraphrased texts for each source paragraph with each Transformer-based model as we apply two algorithms and three MLM levels to consider the variety of abilities to paraphrase a text that usually happens in reality. The highest correlation is obtained by SALAC1 and Longformer as shown in Figure 5. Thus, Longformer's capacity to handle longer input sequences may be useful in producing longer paraphrased texts. For instance, if the input text is a paragraph, Longformer might be better able to capture the overall context of the paragraph and use this context to generate a more accurate and meaningful paraphrased output text.

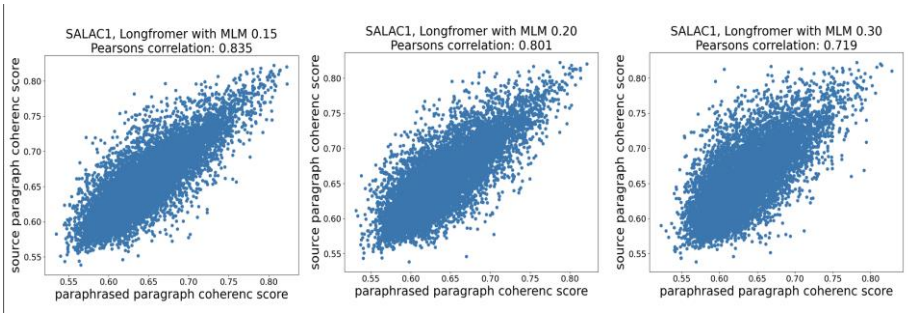

Fig. 5. Correlation of the paraphrased paragraph to the human-written paragraph.

5 Experiment

To address the lack of existing paragraph-level paraphrasing datasets created by paraphrase-generation models, we create the ALECS dataset which is divided into training and testing sets with 938,892 and 234,723 samples, respectively. The main objective is

to study the efficiency of the Transformer-based model in detecting paraphrased paragraphs after reordering their sentences. We apply three state-of-the-art Transformer-based models in their default hyperparameters configurations for paragraph paraphrasing and paraphrase identification: RoBERTa [11], an extension of BERT designed to accommodate lengthier documents, Longformer [12], primarily developed for processing extended documents, and BERT [10], often utilised as a baseline in NLP and NLG studies. However, we consider samples paraphrased using Longformer as they show the highest correlation with the human-written paragraphs (Section 4.2). In addition, we report only the best results obtained by Longformer, because of restricted number of pages.

5.1 Baseline

We use off-the-shelf BERT as a baseline classifier model, which is commonly implemented in most of the existing work. Moreover, Additionally, we consider the work of [20] as a ground truth on paragraph-level classification for the PI task as this study was performed on a paragraph-sized but sentence-level paraphrasing dataset. Furthermore, we compare the classification results of paragraph-level paraphrasing and sentence-level paraphrasing as in[8].

5.2 Results and Discussion

The results in Table 2 show that the paraphrasing level and MLM levels affect the Transformer-based model's efficiency. For Transformer-based models including BERT and Longformer, MLM is a primary, self-supervised, fine-tuning objective. In paraphrase generation models MLM represents the percentage of paraphrased words. Since 0.15 MLM is the standard percentage of paraphrased words when utilising available paraphrase generation tools [32] and because it's more challenging for the Transformer-based model in detecting paraphrased content, we compare our results to those of others at that level. In general, our result outperforms the existing work result using Longformer with 0.15 MLM by 4%. For BERT, although the fact that we reorder the sentences in the paragraphs then paraphrase them, our output is high as in another work result that directly paraphrases text without changing the sentence order. Additionally, we notice that considering the paragraph-level rather than the sentence-level has a positive impact on the Transformer-based model's output.

From the sentence reordering point of view, we can compare our results to [20]. The main difference is that they carried out paragraph paraphrasing without the sentence reorder step. The results prove that Transformer-based models can distinguish between the source text and reordered-paraphrased paragraphs without providing pair information. Longformer provides the best results at the paragraph and sentence levels. We suppose that the global attention prediction (GAP), a feature used by Longformer, enables the model to learn how to focus on the most important sections of a long text.

To expand on what other researchers have found in terms of how text length affects the machine learning algorithm's capacity[8], we can notice the same effect on the

Transformer-based model's results: longer text provides more context and semantics thus improving the efficiency of machine learning and Transformer-based models in PI and PD tasks. Specifically, the F1-score of BERT and Longformer results increase by 18% and 23%, respectively, in detecting paragraph-level paraphrasing with 0.15 MLM (see Table 2 for the differences between the results for sentence vs. paragraph length). These percentages decrease as the percentage of paraphrased paragraphs' words increase.

Table 2. Classification results represented as F1 macro scores.

	The [20] results		**Our results**					
Classifier model	**Bert**	**Long-former**	**Bert**			**Longformer**		
MLM	**0.15**	**0.15**	**0.15**	**0.20**	**0.30**	**0.15**	**0.20**	**0.30**
Paragraph-level length	69	86	83	89	96	90	95	98
Sentence-level length	-	-	65	71	80	67	85	85

6 Conclusion and Limitation

In this work, we investigate the features of Transformer-based models in distinguishing between samples of original paragraphs and their paraphrases at the paragraph-level. Our excellent results with sentence reordering mean that the splitting and merging approach could potentially be used to develop a highly accurate paragraph-level paraphrasing detection approach although this would require a new dataset. To achieve this important objective, we create a large-scale paragraph-level paraphrasing dataset of content from multiple domains mostly related to education. We address the RQ using an experiment that shows high efficiency in detecting even the most difficult sample where the percentage of paraphrased tokens is low (15%) without any information from the source paragraph. Moreover, we report on the impact of text length on the Transformer-based models' efficiency.

In terms of limitation, an examination is conducted on the cutting-edge Transformer-based models, completely omitting ChatGPT due to its inconsistency, which renders it unsuitable for our dataset generation objectives. As for the evaluation methodology, an alternative forum might be explored, but we adhere to the approach advocated by researchers, involving 100 samples and conducting quantitative analysis based on qualitative analysis. Additionally, we implement automatic analysis across the entire dataset.

For future work, based on the findings of our experiment, which show that reordering paragraph sentences does not affect the classifier's capacity to recognise paraphrased paragraphs, we aim to determine how well Large Learning Models (LLMs) can generate and identify paragraph-level paraphrases.

References

[1] R. J. M. Ventayen, 'OpenAI ChatGPT Generated Results: Similarity Index of Artificial Intelligence (AI) Based Model', *Available at SSRN 4332664.*, 2023.

[2] J. Becker, J. P. Wahle, T. Ruas, and B. Gipp, 'Paraphrase Detection: Human vs. Machine Content'. arXiv, Mar. 24, 2023. Accessed: Sep. 15, 2023. [Online]. Available: http://arxiv.org/abs/2303.13989

[3] A. Gupta, A. Agarwal, P. Singh, and P. Rai, 'A Deep Generative Framework for Paraphrase Generation', *AAAI*, vol. 32, no. 1, Apr. 2018, doi: 10.1609/aaai.v32i1.11956.

[4] J. Ganitkevitch, B. V. Durme, and C. Callison-Burch, 'PPDB: The Paraphrase Database', 2013.

[5] J. E. Hu, R. Rudinger, M. Post, and B. Van Durme, 'ParaBank: Monolingual Bitext Generation and Sentential Paraphrasing via Lexically-constrained Neural Machine Translation'. arXiv, Jan. 11, 2019. Accessed: May 31, 2022. [Online]. Available: http://arxiv.org/abs/1901.03644

[6] Y. He, Z. Wang, Y. Zhang, R. Huang, and J. Caverlee, 'PARADE: A New Dataset for Paraphrase Identification Requiring Computer Science Domain Knowledge', *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, Accessed: Sep. 21, 2021. [Online]. Available: http://arxiv.org/abs/2010.03725

[7] T. Foltýnek, N. Meuschke, and B. Gipp, 'Academic Plagiarism Detection: A Systematic Literature Review', *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–42, Jan. 2020, doi: 10.1145/3345317.

[8] A. A. Saqaabi, E. Akrida, A. Cristea, and C. Stewart, 'A Paraphrase Identification Approach in Paragraph Length Texts', in *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, Orlando, FL, USA: IEEE, Nov. 2022, pp. 358–367. doi: 10.1109/ICDMW58026.2022.00055.

[9] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, 'ALBERT: A Lite BERT for Self-supervised Learning of Language Representations'. arXiv, Feb. 08, 2020. Accessed: Mar. 22, 2023. [Online]. Available: http://arxiv.org/abs/1909.11942

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. arXiv, May 24, 2019. Accessed: Mar. 21, 2023. [Online]. Available: http://arxiv.org/abs/1810.04805

[11] Y. Liu *et al.*, 'RoBERTa: A Robustly Optimized BERT Pretraining Approach'. arXiv, Jul. 26, 2019. Accessed: Mar. 21, 2023. [Online]. Available: http://arxiv.org/abs/1907.11692

[12] I. Beltagy, M. E. Peters, and A. Cohan, 'Longformer: The Long-Document Transformer'. arXiv, Dec. 02, 2020. Accessed: Mar. 21, 2023. [Online]. Available: http://arxiv.org/abs/2004.05150

[13] J. Roe and M. Perkins, 'What are Automated Paraphrasing Tools and how do we address them? A review of a growing threat to academic integrity', *Int J Educ Integr*, vol. 18, no. 1, p. 15, Dec. 2022, doi: 10.1007/s40979-022-00109-w.

[14] S. Wan, M. Dras, R. Dale, and C. Paris, 'Using Dependency-Based Features to Take the "Para-farce" out of Paraphrase', *In Proceedings of the Australasian language technology workshop*, p. 8, 2006.

[15] Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, and Brown, 'Text Classification Algorithms: A Survey', *Information*, vol. 10, no. 4, p. 150, Apr. 2019, doi: 10.3390/info10040150.

[16] R. Ferreira, G. D. C. Cavalcanti, F. Freitas, R. D. Lins, S. J. Simske, and M. Riss, 'Combining sentence similarities measures to identify paraphrases', *Computer Speech & Language*, vol. 47, pp. 59–73, Jan. 2018, doi: 10.1016/j.csl.2017.07.002.

[17] R. Yang, J. Zhang, X. Gao, F. Ji, and H. Chen, 'Simple and Effective Text Matching with Richer Alignment Features', *arXiv:1908.00300 [cs]*, Aug. 2019, Accessed: Sep. 08, 2021. [Online]. Available: http://arxiv.org/abs/1908.00300

[18] W. B. Dolan and C. Brockett, 'Automatically Constructing a Corpus of Sentential Paraphrases', *In Third International Workshop on Paraphrasing*, 2005.

[19] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso, 'An Evaluation Framework for Plagiarism Detection', *In Coling 2010: Posters (pp. 997-1005)*, 2010.

[20] J. P. Wahle, T. Ruas, N. Meuschke, and B. Gipp, 'Are Neural Language Models Good Plagiarists? A Benchmark for Neural Paraphrase Detection', in *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, Champaign, IL, USA: IEEE, Sep. 2021, pp. 226–229. doi: 10.1109/JCDL52503.2021.00065.

[21] Z. Lin, Y. Cai, and X. Wan, 'Towards Document-Level Paraphrase Generation with Sentence Rewriting and Reordering'. arXiv, Sep. 15, 2021. Accessed: May 31, 2022. [Online]. Available: http://arxiv.org/abs/2109.07095

[22] D. Qiu, 'Document-level paraphrase generation base on attention enhanced graph LSTM', *Applied Intelligence, 1-13*, p. 13, 2022.

[23] F. M. Prentice and C. E. Kinden, 'Paraphrasing tools, language translation tools and plagiarism: an exploratory study', *Int J Educ Integr*, vol. 14, no. 1, p. 11, Dec. 2018, doi: 10.1007/s40979-018-0036-7.

[24] Larock MH, Tressler JC, and Lewis CE, *Mastering effective English*. Copp Clark Pitman, Mississauga, 1980.

[25] Elsner, M. and Charniak, E, 'Extending the Entity Grid with Entity Specific Features'. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (pp. 125-129), Jun. 2011.

[26] R. Soricut and D. Marcu, 'Discourse generation using utility-trained coherence models', in *Proceedings of the COLING/ACL on Main conference poster sessions -*, Sydney, Australia: Association for Computational Linguistics, 2006, pp. 803–810. doi: 10.3115/1273073.1273176.

[27] A. Shen, M. Mistica, B. Salehi, H. Li, T. Baldwin, and J. Qi, 'Evaluating Document Coherence Modeling', *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 621–640, Jul. 2021, doi: 10.1162/tacl_a_00388.

[28] C. van der Lee, A. Gatt, E. van Miltenburg, and E. Krahmer, 'Human evaluation of automatically generated text: Current trends and best practice guidelines', *Computer Speech & Language*, vol. 67, p. 101151, May 2021, doi: 10.1016/j.csl.2020.101151.

[29] J. Amidei, P. Piwek, and A. Willis, 'Rethinking the Agreement in Human Evaluation Tasks (Position Paper)', *In Proceedings of the 27th International Conference on Computational Linguistics (pp. 3318-3329)*, 2018.