Paraphrase Generation and Identification at paragraph-level

AL SAQAABI, ARWA,KHALID,S

Paraphrase Generation and Identification at paragraph-level

Arwa Khalid Al Saqaabi

A thesis presented for the degree of

Doctor of Philosophy in Computer Science

Supervised by: Dr. Craig D. Stewart, Dr. Eleni Akrida, and Prof. Alexandra I. Cristea

Artificial Intelligence and Human Systems Research Group

Department of Computer Science

Durham University in the United Kingdom

2025

DEDICATION

To the sake of Allah, all praise is due to him.

To the memory of my beloved father, Khalid Saleh Al Saqaabi, whose encouragement to study abroad inspired me to pursue my PhD in Computer Science.

To my dear mother Fatimah Abdullah AlShdoky, whose love and prayers have been a source of strength throughout this journey.

To my husband Abdullah Saleh Alkhudhayr and our children Ziyad and Eilaf, whose unwavering support and patience have meant everything to me.

To my brothers and sister, thank you for your constant encouragement and belief in me.

I extend my heartfelt gratitude to my supervisors, whose guidance and mentorship have been invaluable in completing this work.

Paraphrase Generation and Identification at Paragraph-Level

Arwa Khalid Al Saqaabi

Submitted for the Degree of Doctor of Philosophy

2025

ABSTRACT

The widespread availability of the Internet and the ease of accessing written content have significantly contributed to the rising incidence of plagiarism across various domains, including education. This behaviour directly undermines academic integrity, as evidenced by reports highlighting increased plagiarism in student work. Notably, students tend to plagiarize entire paragraphs more often than individual sentences, further complicating efforts to detect and prevent academic dishonesty. Additionally, advancements in natural language processing (NLP) have further facilitated plagiarism, particularly by using online paraphrasing tools and deep-learning language models designed to generate paraphrased text. These developments underscore the critical need to develop and refine effective paraphrase identification (PI) methodologies.

This thesis addresses one of the most challenging aspects of plagiarism detection (PD): identifying instances of plagiarism at the paragraph-level, with a particular emphasis on paraphrased paragraphs rather than individual sentences. By focusing on this level of granularity, the approach considers both intra-sentence and inter-sentence relationships, offering a more comprehensive solution to the detection of sophisticated forms of plagiarism. To achieve this aim, the research examines the influence of text length on the performance of NLP machine learning (ML) and deep learning (DL) models. Furthermore, it introduces ALECS-SS (ALECS – Social Sciences), a large-scale dataset of paragraph-length paraphrases, and develops three novel SALAC algorithms designed to preserve semantic integrity while restructuring paragraph content. These algorithms suggest a novel approach that modifies the structure of paragraphs while maintaining their semantics. The methodology involves converting text into a graph where each node corresponds to a sentence's semantic vector, and each edge is weighted by a numerical value representing the sentence order probability. Subsequently, a masking approach is applied to the reconstructed paragraphs modifying the

lexical elements while preserving the original semantic content. This step introduces variability to the dataset while maintaining its core meaning, effectively simulating paraphrased text. Human and automatic evaluations assess the reliability and quality of paraphrases, and additional studies examine the adaptability of SALAC across multiple academic domains. Moreover, state-of-the-art large language models (LLMs) are analysed for their ability to differentiate between human-written and machine-paraphrased text. This investigation involves the use of multiple PI datasets in addition to the newly established paragraph-level paraphrases dataset (ALECS-SS).

The findings demonstrate that text length significantly affects model performance, with limitations arising from dataset segmentation. Additionally, the results show that the SALAC algorithms effectively maintain semantic integrity and coherence across different domains, highlighting their potential for domain-independent paraphrasing. The thesis also analysed the state-of-the-art LLMs' performance in detecting auto-paraphrased content and distinguishing them from human-written content at both the sentence and paragraph levels, showing that the models could reliably identify reworded content from individual sentences up to entire paragraphs. Collectively, these findings contribute to educational applications and plagiarism detection by improving how paraphrased content is generated and recognized, and they advance NLP-driven paraphrasing techniques by providing strategies that ensure that meaning and coherence are preserved in reworded material.

DECLARATION

The work and experiments in this thesis are based on research carried out within the Artificial Intelligence and Human Systems Group at the Department of Computer Science at Durham University, UK. No part of this thesis has been submitted elsewhere for any other qualification or degree, and it is all the author's work.

List of Publications

- Al Saqaabi, A., Akrida, E., Cristea, Alexandra. I., & Stewart, C. (2022). A Paraphrase Identification Approach in Paragraph Length Texts. *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, 358–367. https://doi.org/10.1109/ICDMW58026.2022.00055 **(Chapter 5)**
- Al Saqaabi, A., Stewart, C., Akrida, E., & Cristea, A. I. (2024, June). Paraphrase Generation and Identification at Paragraph-Level. *In International Conference on Intelligent Tutoring Systems (pp. 278-291)*. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-63031-6_24 **(Chapter 6)**
- Al Saqaabi, A., Stewart, C., Akrida, E., & Cristea, A. I. (2025). A Deep-Learning Approach for Paraphrase Generation and Identification at paragraph-level. *Neural Processing Letters,* 57(3), 59. https://doi.org/10.1007/s11063-025-11771-9 (**Chapters 5&7)**
- Al Saqaabi, A., Stewart, C., Akrida, E., & Cristea, A. I. (2025). Multi-Domain Evaluation of Auto-paraphrase Generation at Paragraph-Level: Insights for Education and Plagiarism Detection. *In International Conference on Intelligent Tutoring Systems* (pp. 229-243). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-98284-2_18 (**Chapter 7)**

ACKNOWLEDGEMENTS

First and foremost, I express my profound gratitude to Allah for granting me the strength, and guidance to complete this work.

I extend my sincere appreciation to my supervisors, Dr. Craig D. Stewart, Dr. Eleni Akrida, and Prof. Alexandra I. Cristea, for their invaluable guidance, insightful feedback, and steadfast support throughout my PhD journey. Their mentorship has been pivotal to both the success of this research and my academic development.

I am deeply indebted to my family. To my mother, Fatimah, whose unwavering love, prayers, and encouragement have been a constant source of strength; to my late father, Khalid, whose vision and encouragement inspired me to pursue this academic endeavour; and to my husband, Abdullah, whose support, patience, and understanding have been immeasurable—thank you for being my pillars of strength.

To my children, Ziyad and Eilaf, your love has been a source of motivation throughout this challenging journey. I also extend my heartfelt gratitude to my sister, Maumonah, and brothers, Musab, Abdullah, Mohammed, Suhaib, Abdulrahman and Tamim, for their encouragement and continuous support.

I would also like to express my heartfelt thanks to my close friends, Dr. Aisha Alsehaim, Dr. Laila Al Rajhi, Dr. Ghada Alosaimi, and Dr. Muna Almushyti, as well as to my colleagues and all the participants who contributed to this research.

I would like to express my sincere gratitude to Qassim University for awarding me a full scholarship, which made it possible to fund my doctoral studies. I am also thankful to the Saudi Cultural Bureau in Britain (SACB) for their continued support throughout this journey. Additionally, I extend my appreciation to the Department of Computer Science at Durham University for welcoming me as a student and granting me access to outstanding facilities that greatly supported my research.

Finally, I wish to acknowledge everyone who has supported me, directly or indirectly, throughout this academic journey. Your contributions, no matter how small, have been instrumental in this achievement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ACC Accuracy

AI Artificial Intelligence

ALBERT A Lite BERT For Self-Supervised Learning of Language Representations

ALECS-SS Paragraph-Level Machine Paraphrased Dataset

ARI Automated Readability Index

BERT Bidirectional Encoder Representations from Transformers

BOW Bag-Of-Words

CBOW Continuous Bag-Of-Words

CLI Coleman-Liau Index

CNN Convolutional Neural Network

DistilBERT A Distilled Version of BERT

DL Deep Learning

FKG Kincaid Grade

FN False Negative

FP False Positive

FRE Flesch Reading Ease

GFI Gunning Fog Index

GNN Graph Neural Network

GPT Generative Pre-Training Transformer Model

GRUs Gated Recurrent Units

IAA Inter-Annotator Agreement

LLM Large Language Model

Longformer Long-Document Transformer

LR Logistic Regression

LSTM Long-Short-Term Memory

ML Machine Learning

MLM Masked Language Modelling

MLP Multi-Layer Perceptron

MPM Message Passing Mechanism

MSRP Microsoft Research Paraphrase Corpus

NLG Natural Language Generation

NLP Natural Language Processing

NLTK Natural Language Toolkit

NSP Next Sentence Prediction

P Precision

P4P Paraphrase for Plagiarism Dataset

PD Plagiarism Detection

PI Paraphrase Identification

POS Part-Of-Speech

PSA Plagiarized Short Answers Dataset

QQP Quora Question Pairs

R Recall

R-GCNs Relational Graph Convolutional Networks

RoBERTa Robustly Optimized BERT Pretraining Approach

RRN Recurrent Neural Network

SALAC Sentence Reorder Algorithm

SBERT Modified Version of the Pre-Trained BERT

SMOG Simple Measure of Gobbledygook

SOP Sentence Order Prediction

SRL Semantic Role Labelling

SVM Support Vector Machines

T5 Text-To-Text Transfer Transformer

TF-IDF Term Frequency-Inverse Document Frequency

TN True Negative

TP True Positive

Webis-CPC-11 Webis Crowd Paraphrase Corpus

CHAPTER 1: INTRODUCTION

1.1. Introduction

Plagiarism detection (PD) and paraphrase identification (PI) are crucial tasks in natural language processing (NLP), which provide computational tools and methodologies to analyse textual data and recognise language patterns. PD and PI aim to maintain academic integrity and verify the content's originality in the digital evaluation age. This evaluation provides easy access to information, content, and online paraphrasing tools, which create an opportunity to simplify or explain the text in different ways. They rewrite the text without affecting its meaning. This method of text paraphrasing could be useful in academic aspects; however, it has negative impacts on education, as represented by plagiarism. Students may paraphrase the original text and present it as their work using auto-paraphrasing tools.

Thus, the need to develop effective approaches to detect plagiarism and identify paraphrases has become more important in academia, journalism, publishing, and other fields where innovation, novelty, and originality are respected, especially with the noticed increase of plagiarism in these fields (Elkhatat, 2023; Alsallal et al., 2013; Clough & Stevenson, 2011; Mihalcea et al., 2004; Cohn et al., 2008; Sánchez-Vega et al., 2013; Becker et al., 2023). Plagiarism is the act of using others' work or ideas without acknowledgement by verbatim copying or paraphrasing (Shoyukhi et al., 2023). Verbatim copying is a direct copy of a source, while paraphrasing involves rewriting the source content in one's own words.

PD approaches fall into two primary categories, namely intrinsic and extrinsic (Ehsan et al., 2019). While intrinsic approaches are used to identify inconsistent segments of text, extrinsic methods can detect paraphrased text and exact verbatim copying by matching suspicious segments in a text to the sources (Jirapond Muangprathub et al., 2021). In other words, intrinsic methods rely on the internal analysis of a document without directly comparing it to extrinsic sources. This analysis checks the text's general coherence, vocabulary, sentence structure, sentence length, and punctuation to create a profile of the author's writing style. Thus, variations from this profile could be a sign of possible plagiarism. This method requires a large amount of training data in order to generate accurate author profiles; however, inaccurate

results are expected when the authors change their writing style or when several authors have similar writing styles, making it difficult to accurately identify the true author.

In the case of extrinsic plagiarism detection methods, they retrieve related text from a large collection of already published materials, including scholarly papers, published works, and web content. Then, compare it to the suspicious text considering different levels, from words and phrases to whole documents. It could be relatively simple and straightforward to implement, especially in detecting direct copying or verbatim plagiarism, which led to missed cases of paraphrasing or rephrasing plagiarism. To enhance extrinsic plagiarism detection approaches, the PI must be involved.

Hence, PI is a fundamental aspect of extrinsic plagiarism detection approaches that aim to identify texts reflecting the same meaning. PI involves evaluating the semantic similarity between different fragments of text regardless of how they are worded or structured. Therefore, PI is crucial for PD as well as for numerous NLP tasks such as machine translation (Thompson & Post, 2020), information retrieval (H. Li & Xu, 2014), summarisation (Chali & Egonmwan, 2024), and question answering (Barron-Cedeno & Vila, 2013; Y. Yang et al., 2015; Ben Aouicha et al., 2018). One common approach to PI is the use of alignment-based methods, which compare pairs of text to identify similarities at the word or phrase level (A. Gupta et al., 2018). It includes implementing neural language models, such as encoder-decoder architectures or transformer-based models to learn semantic representations of sentences and assess their similarity (Razaq et al., 2024).

In this thesis, a contribution is made to the field of PI by detecting paraphrased text within paragraph-length and paragraph-level paraphrasis, which involve sentence reordering, splitting, and/ or joining, using state-of-the-art transformer models. Additionally, the inter-sentence and intra-sentence relations are taken into consideration to create a large-scale paragraph-level dataset called ALECS-SS. After that, the large language models (LLMs) are examined in terms of their ability to distinguish between the auto-paraphrased text and the sources.

An overview of this thesis's subject is provided in this chapter. First, Section 1.1 defines the scope of the research. This contributes to the clarification of the research motivation in Section 1.2 and the research problem in Section 1.3. The objectives of this thesis are then covered in Section 1.4. The research questions are presented in Section 1.5. Then, in Section

1.6, a summary of the scientific contributions is provided. Lastly, Section 1.7 presents the thesis structure.

1.2. Research Scope

PI can be seen as a kind of text classification task, which is when a model takes a text segment as input and assigns one or more predefined labels to describe the content. The number of classes can vary depending on the type of classification: multiclass, binary, or multilabel. In multiclass classification, the text is assigned to one class out of many, while in binary classification, it is assigned to one of just two classes, which are paraphrased or non-paraphrased in PI. Multilabel classification allows a text segment to have multiple class labels assigned to it at the same time (Wu, 2022).

While considerable research has been dedicated to the PI task, the majority of existing studies concentrate on developing a paraphrase detection algorithm taking sub-sentence level or sentence-level into consideration, ignoring the paragraph-level. According to Kowsari et al. (2019), there are four levels of scope in the text classification system that can be used: sub-sentence level, sentence-level, paragraph-level, and document-level.

PI at the paragraph-level has received less attention, primarily due to the absence of a suitable dataset, presenting a significant challenge and requiring substantial time investment. This thesis seeks to fill this gap by concentrating on exploring intra-sentence and inter-sentence relationships in paragraphs. Firstly, it investigates the impact of text length on the ML and DL models' results. This is followed by employing SALAC and state-of-the-art DL models known for their capacity to analyse and paraphrase lengthy text, resulting in the creation of a paragraph-level dataset called ALECS-SS. The examples of sentence-level and paragraph-level paraphrases are illustrated in Table 1.1. ALECS-SS offers an opportunity to analyse cutting-edge classification algorithms for distinguishing between paragraphs written by humans and those generated through automated paraphrasing techniques. The primary objective of this research is to contribute to the field of PD. To achieve this, ALECS-SS comprises text from various academic subjects, including economics and sociology, sourced exclusively from paragraph-length extracts of Wikipedia articles. Then, reorder each paragraph's sentences based on their semantic relationships by implementing and examining three innovative algorithms. After that, reconstructed paragraphs are paraphrased using LLMs to create the

ALECS-SS dataset. Following this, both human judgments and automated metrics are employed to assess how well SALAC performs across the selected domain. This dataset is then used to train PI algorithms that could distinguish between human-written and auto-paraphrased paragraphs.

Table 1.1 The examples of sentence-level and paragraph-level paraphrases

Paraphrasis type	**Text**
Sentence-length	***Source:*** Considering the goal of obtaining universal health care as part of Sustainable Development Goals, scholars request policymakers to acknowledge the form of healthcare that many are using.
	Sentence-level paraphrases: Considering the importance of obtaining universal health care as part of Sustainable Development Goals, scholars request policymakers to acknowledge the form of healthcare that many are using.
Paragraph-length	***Source:*** Considering the goal of obtaining universal health care as part of Sustainable Development Goals, scholars request policymakers to acknowledge the form of healthcare that many are using. Scholars state that the government has a responsibility to provide health services that are affordable, adequate, new and acceptable for its citizens. Public healthcare is very necessary, especially when considering the costs incurred with private services. Many citizens rely on subsidized healthcare. The national budget, scholars argue, must allocate money to the public healthcare system to ensure the poor are not left with the stress of meeting private sector payments.
	Sentence-level paraphrases: Considering the importance of obtaining universal health care as part of Sustainable Development Goals, scholars request policymakers to acknowledge the form of medical care that many are using. Scholars state that the government bears a responsibility to provide health services that are affordable, adequate, new and acceptable for its citizens. Public medical care is very necessary, especially when considering expenses incurred with private services. Many citizens rely on subsidized medical care. The national spending plan, scholars argue, must allocate resources to the public healthcare system to ensure the poor are not left with the pressure meeting private sector payments.
	Paragraph-level paraphrases: Many citizens rely on subsidized medical care. Public medical care is very necessary, especially when considering expenses incurred with private services. Scholars state that the government bears a responsibility to provide health services that are affordable, adequate, new and acceptable for its citizens. Considering the importance of obtaining universal health care as part of Sustainable Development Goals, scholars request policymakers to acknowledge the form of medical care that many are using. The national spending plan, scholars argue, must allocate resources to the public medical care system to ensure the poor are not left with the pressure meeting private sector payments.

1.3. Research Motivation

The growing reliance on the internet provides convenient access to textual content and tools for creating paraphrases. Students often exploit these resources to present someone else's ideas

or work as their own. This type of plagiarism puts academic integrity and intellectual property at risk for several reasons. Firstly, it weakens innovation and originality. In other words, it compromises the principles of originality and creativity, which are vital to intellectual and scholarly endeavours. It impedes the production of new knowledge as well as the expansion of research and scholarship. Secondly, undermining academic objectives as education aims to develop abilities in critical thinking, research skills, and the ability to produce new knowledge. Plagiarism avoids these goals by encouraging students to skip the real learning process. It lessens the importance of tests and assignments since they are no longer trustworthy measures of a student's knowledge and abilities. In addition, plagiarism threatens the trust and credibility that support academic institutions. If academic work is not authentic and reliable, it compromises the integrity of the entire educational system. This could have long-term impacts on people's and institutions' reputations. Moreover, impeding equitable assessment, it is impossible to fairly assess someone's knowledge and abilities when they are plagiarised. Educators, assessors, and employers find it difficult to fairly evaluate the skills of students or researchers who turn in plagiarised work. Meritocratic values in academics and the workforce are compromised by this, finally, restraining advancement in research and reducing it. Plagiarism hinders the progress of research since it allows for false claims of authorship. This could mislead the scientific community, waste funds on fruitless research projects, and impede actual progress in the field.

To uphold the principles of integrity, authenticity, and intellectual honesty in the seeking of knowledge, it is imperative to prioritise addressing academic plagiarism. Working within the academic education sector, I dedicated my research efforts to the identification of paraphrased text, a pivotal aspect of PD. This undertaking is not only central to maintaining the credibility of educational institutions but also holds significant relevance in various domains of NLP, such as machine translation, information retrieval, and text generation.

In terms of machine translation, where the identification of paraphrases involves finding equivalent terms in different languages, this process enhances the quality of translated text. It becomes essential to ensure that the translated material accurately captures the intended meaning of the source content.

The identification of paraphrases also proves crucial in information retrieval systems. By recognising paraphrased variations of user queries within the document corpus, these systems

can provide more comprehensive search results. This contributes to an improved user experience, facilitating efficient access to relevant information.

Moreover, natural language generation (NLG) systems aim to generate diverse and coherent sentences, and the ability to identify paraphrases helps prevent the generation of redundant or repetitive content. This enhances the overall effectiveness of NLG in creating meaningful and varied textual outputs.

1.4. Research Problem

The availability and growth of tools and NLG models that are used to paraphrase text could enhance machine translation, information retrieval, 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. In educational contexts, occurrences of academic plagiarism have increased, as it has been detected in diverse student tasks, spanning reports, assignments, projects, and beyond (Elkhatat et al., 2023). According to Alsallal et al. (2013), one of the worst types of research misconduct is academic plagiarism, and it has a negative impact on academic integrity.

Plagiarism is defined as using someone's written work without giving a reference to the source or claiming the ideas are taken from the work of others (Maurer et al., 2006). The copying of many words from the source, regardless of giving a reference, is also considered an act of plagiarism (Bär et al., 2012). The modification of sentences in such a way that the original structure of the sentences without acknowledgement is used by the author also falls into the category of plagiarism. According to Ventayen (2023), the current state of artificial intelligence (AI) models makes it possible to create highly coherent and contextually suitable paraphrasing, raising concerns about the potential for generating plagiarised material. In addition, Becker et al. concluded that it is difficult to differentiate artificially paraphrased text from human-written text (Becker et al., 2023). This introduces potential risks related to academic dishonesty and plagiarism, with possible significant academic and legal consequences, as highlighted by (Foltýnek et al., 2019).

The opportunities for engaging in technologically enabled academic misconduct are growing, and with them are the tools for spotting and stopping it. The development of these

tools has been an active field of study in computer science and NLP, including PI as a way to detect plagiarism.

Paraphrased text is an output text that preserves the meaning of the input text in other forms of text (A. Gupta et al., 2018). Bhagat and Hovy defined paraphrasing as a means of conveying the same meaning but with different sentence structure and wording (Bhagat & Hovy, 2013). These definitions count verbatim as a non-paraphrased case. Formally, let us have two different texts, A and B. If the information, φ, that can be derived from A can also be inferred from B and vice versa, then A is a paraphrase of B (Formula 1): α represents a given domain or background knowledge (Burrows et al., 2013)

$$(A \wedge \alpha \mid \varphi) \iff (B \wedge \alpha \mid \varphi) \qquad \text{Formula 1.1}$$

From these definitions, it is obvious that PI is implicitly part of PD. Both PI and PD have assumed tremendous importance for academic institutions, researchers, and publishers for the preservation of academic integrity (Bach et al., 2014). PI is a method that aims to measure the degree of similarity of sentences and phrases with the source and verify the semantic similarities between sentences (Das & Smith, 2009; Fernando & Stevenson, 2008). PI also helps determine whether two pieces of text carry the same meaning, which plays a vital role in natural language applications such as information retrieval (H. Li & Xu, 2014), answering questions (Barron-Cedeno & Vila, 2013; Y. Yang et al., 2015; Ben Aouicha et al., 2018) and other NLP downstream tasks mentioned previously.

Attempts to solve the problem of PI in past studies were focused mainly on comparing words in sentences (Wan et al., 2006; Vrublevskyi & Marchenko, 2020), or sentence to sentence (Nguyen et al., 2019; Devlin et al., 2019) or phrases in sentences (Arase & Tsujii, 2021). These studies achieved robust results. However, comparing each sentence in the suspicious document to all sentences in the source documents is not efficient for long texts. Moreover, they focus on paraphrased texts at the sentence-length and sentence-level paraphrases (Ganitkevitch et al., 2013; R. Yang et al., 2019; Prentice & Kinden, 2018; Hu et al., 2019), and paragraph-level (He et al., 2020; Asghari et al., 2021) , but utilise sentence-level paraphrase methods only. These approaches consider the meaning of each sentence independently; they do not determine any semantic relationships between sentences, which is harder to achieve 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 Foltýnek et al. (2019), plagiarists reuse paragraphs, not sentences, the most frequently. Thus, the need to provide insight into the efficient paragraph-level detection algorithms is clear, and it is one of this thesis objectives.

Paragraph-level paraphrasis includes sentence reordering, sentence splitting, and/or sentence merging. The initial work in this area is presented in (Al Saqaabi et al., 2022), where ML and DL algorithms are applied to detect paraphrasing (focusing on the paragraph-level), details in Chapter 5; however, this work is limited by the fact that very few suitable datasets are available for this type of research. Thus, a new dataset must be created to investigate detection algorithms' efficiency in recognising auto-paraphrased text at the paragraph-level, as detailed in Chapters 4 and 5.

This research targets the introduction of a new approach to create a paragraph-level dataset (Chapter 6), after proving that detecting paraphrased text at the paragraph-level is more accurate and robust compared to detecting paraphrased text at the sentence-level (Chapter 5). Paragraph-level paraphrases detection considers both inter-sentence and intra-sentence relationships, which guarantees that the text's meaning remains unaffected (Chapters 5 and 6). Moreover, this work examines the efficiency of state-of-the-art transformer learning models in detecting auto-paraphrased text at the paragraph-level (Chapter 8).

1.5. Research Questions (RQ)

The research questions are formulated based on the research problem and the specific gaps found in the literature (Chapter 3). It focuses on how to detect paraphrased text and distinguish between the source and auto-paraphrased text at the paragraph-level. This is achieved by 9 sub-questions under the umbrella research question:

- **RQ:** *How can the effectiveness of machine learning and deep learning paraphrase identification algorithms be affected by the variety of text length and paraphrasis levels?*

To assist in addressing this broad research question, the following sub-research questions are developed:

- ***RQ1:*** *How does the length of a piece of text affect the accuracy of the paraphrase identification approach used?*

This question is considered the cornerstone, as answering it will highlight the importance of detecting paraphrasing at the paragraph-level rather than focusing solely on sentences. This RQ aims to explore the impact of text length on the performance of PI methods. The findings from this RQ will help refine PI strategies for texts of different lengths. This RQ is addressed in Chapter 5.

- ***RQ2:*** *What features are most effective for paraphrase identification across different levels of paraphrasing and varying text lengths?*

This RQ investigates the types of features that contribute most effectively to PI across different levels of paraphrasis, including sentences and paragraphs. It aims to determine which features are best suited for each paraphrasis level and how they can be combined to improve overall detection accuracy. This RQ is addressed in Chapter 5.

- ***RQ3:*** *Which of the three novel paragraph-level paraphrasing algorithms (SALACs) proposed preserves the source paragraph's meaning most effectively?*

After determining the most appropriate text length for PI as a result of RQ1 and RQ2, the next RQs investigate in deep the PI at paragraph-length and paragraph-level paraphrasis. RQ3 compares the accuracy of three novel paraphrasing algorithms through both human and automatic evaluation while ensuring that the source paragraph's meaning is preserved. This RQ is addressed in Chapter 6.

- ***RQ4:*** *Is there a correlation between the similarity score assigned by human evaluators and the automatically generated coherence score used for paraphrase generation by the paragraph-level algorithms (SALACs)?*

This RQ examines the relationship between coherence scores assigned by human evaluators and those generated automatically. By exploring this correlation, the research aims to evaluate the reliability of automated coherence scoring systems and their alignment with human judgment. The findings will provide insights into the consistency and validity of automated metrics for assessing textual coherence. This RQ is addressed in Chapter 6.

- ***RQ5:*** *Is there a correlation between the automatically generated paraphrased paragraph's semantic similarity score and the human-written paragraph's semantic similarity score?*

This RQ examines the relationship between the semantic similarity scores generated automatically for paraphrased paragraphs and human-written paragraphs. The aim is to highlight whether the semantics of the text are affected after applying the paraphrasing approach. This RQ is addressed in Chapter 6.

- ***RQ6:*** *How does the paraphrasing quality of SALAC algorithms vary across multiple domains?*

This RQ investigates whether the performance of the SALAC algorithms remains consistent when applied to texts from different domains or if there are noticeable variations in paraphrasing quality. Since domain-specific language characteristics may influence the effectiveness of paraphrasing, this analysis aims to determine whether the algorithms can generalise well across various text types. By comparing paraphrases generated for texts from multiple domains, the author will assess whether domain-specific nuances affect coherence and semantic preservation. This RQ is addressed in Chapter 7.

- ***RQ7:*** *Are there domain-specific challenges in paraphrase quality as perceived by human evaluators?*

While automated evaluation metrics provide an objective assessment of paraphrase quality, human evaluation remains crucial in understanding the perceived coherence and faithfulness of paraphrased texts. This research question explores whether human evaluators identify specific challenges when assessing paraphrases from different domains. Factors such as loss of meaning or decreased readability may be more prominent in certain domains than others. By analyzing human feedback across multiple domains, the author aims to uncover recurring patterns and challenges that might limit the generalizability of the SALAC algorithms. These insights will contribute to refining the models and improving their adaptability across diverse text types. This RQ is addressed in Chapter 7.

- ***RQ8:*** *How effectively can autoencoding models discriminate between the source (human-written) and machine-paraphrased text generated by the paragraph-level method, without requiring a direct comparison between the two?*

This RQ explores the effectiveness of autoencoding models in distinguishing between source and auto-paraphrased texts at the paragraph-level. The objective is to evaluate the capability of these models to detect structural and lexical changes introduced through the novel paraphrasing approach. This RQ is addressed in Chapter 8.

- ***RQ9****: How effectively can state-of-the-art autoregressive models discriminate between the source (human-written) and machine-paraphrased text generated by the paragraph-level method, without requiring a direct comparison between the two?*

This RQ investigates the effectiveness of state-of-the-art autoregressive models in differentiating between source and auto-paraphrased texts at the paragraph-level. The aim is to assess the models' ability to capture and respond to structural and lexical modifications introduced during paraphrasing, comparing the findings to the result of the RQ6. This RQ is addressed in Chapter 8.

To address most of these questions, a dataset that satisfies appropriate requirements should be available. For the first two research questions (RQ1 and RQ2), ML and DL methods are applied in two of the most common datasets in the PI domain, namely, Microsoft Research Paraphrase Corpus (MSRP) and Webs Crowd Paraphrase Corpus 2011 (Webis-CPC-11). The result of this study highlighted the need to create a dataset consisting of paragraph-level paraphrases. To solve this problem, three algorithms are developed and joined with transformer models, then their outputs are evaluated by human and automatic metrics (RQ3-RQ7). For the last two research questions (RQ8 and RQ9), detecting algorithms are applied to classify whether a paragraph is human-written (source) text or auto-paraphrased text, utilising state-of-the-art LLMs.

1.6. Research Objectives (RO)

To address the research questions mentioned above, the author has set the following objectives:

- **RO1:** To investigate how the text length affects the classification algorithm's results in the PI task. Accordingly, short texts refer to sentence-length samples, mid-length texts to paragraph-length samples, and long texts refer to full documents, also sentence-level and paragraph-level paraphrases are considered. This addresses RQ1 and it is explored in Chapter 5.
- **RO2:** To select which hand-crafted features and word embedding are more effective on sentence-length and paragraph-length text in terms of PI task. This focuses on RQ2 (described in more detail in Chapter 5).
- **RO3:** To create an extensive dataset that consists of paragraph-level paraphrases using state-of-the-art transformer models. This dataset aims to serve as a valuable resource for subsequent applications in NLP and is intended to answer most of the research questions of this thesis, specifically RQ3-RQ9, as further explained in Chapters 5-7.
- **RO4:** To develop three novel algorithms that aim to generate paragraph-level paraphrases by altering the paragraph structure while preserving its semantics. Furthermore, investing the LLM's capabilities to modify the paragraph lexically without affecting its meaning. This is essential for addressing RQ3-RQ5, as discussed in Chapters 5 and 6.
- **RO5:** To assess the generalisability of the SALAC algorithms across many domains by analysing their paraphrasing quality in a variety of domain contexts. This investigation seeks to find out whether specific domains present greater challenges for automatic paraphrasing and whether the algorithms consistently preserve meaning across different text types. The study aims to determine if human evaluators recognise particular domain-specific difficulties that may not be entirely reflected by automated evaluation metrics. Understanding these variations will yield significant insights into the adaptability of the SALAC algorithms and their effectiveness in generating quality paraphrases across various domains. This addresses RQ6 and RQ7, which are explored in Chapter 7.
- **RO6:** To assess techniques employing cutting-edge DL models for the identification of paraphrased text at the paragraph-level, aiming to improve PD methods. This is crucial in addressing RQ8 and RQ9, as discussed in Chapter 8.

1.7. Research Contributions

This thesis contributes to the domain of PI by highlighting the impact of text length and handcrafted features on the PI ML and DL approaches. Then, constructing an extensive dataset containing paragraph-level paraphrase samples. These samples are generated using cutting-edge transformer models, then evaluated by human and auto metrics. After that, the study delves into the exploration of detection algorithms. The anticipated outcomes of this research aim to support PD methods employed in the educational sector. Furthermore, the dataset has the potential to provide advantages in other domains, including machine translation, summarisation, and data augmentation.

Contributions are as follows:

- This research presents a novel framework that investigates the impact of text length and paraphrasing levels on PI. By separating dataset samples based on text length and paraphrasing levels (sentence, paragraph, or passage), then feeding them into ML and DL models. The methodology integrates multiple models and features into a cohesive system, analysing their performance across subsets. Chapter 5 investigates RQ1 and RQ2.
- This study also enhances the efficiency of ML and DL algorithms by eliminating the need to compare each sentence individually. This leads to improved detection accuracy and F1-score, even when dealing with the complexities of paragraph-level paraphrasis. Chapter 5 investigates RQ1 and RQ2.
- This research contributes, to the best of the author's knowledge, by building the first large-scale paragraph-level paraphrases labelled dataset (ALECS-SS) from diverse Wikipedia domains, specifically designed for PI tasks. The dataset is extensive, containing a significant number of labelled samples, and serves as a valuable resource for the research community. Chapter 6 investigates RQ3 and RQ5.
- Three novel algorithms (SALACs) for reordering the sentences of a source paragraph without altering its meaning are developed, taking into account both intra- and inter-sentence relations. These approaches are necessary because traditional paraphrasing techniques often fail to maintain the semantic integrity of a paragraph when sentences are reordered. By addressing this issue, the proposed algorithms ensure that the semantic similarity to the source paragraph is preserved. Both automatic and human

evaluations demonstrate that these algorithms can successfully reorder sentences while maintaining the meaning of the source paragraph, making them valuable tools for paraphrase generation. Chapter 6 investigates RQ3 and RQ5.

- The generalisability of SALAC algorithms is assessed in different domain contexts to evaluate their adaptability and robustness across various disciplines. This assessment is essential, as the quality of paraphrasing might vary based on domain-specific linguistic complexities. This study offers deeper insights into the impact of language characteristics on paraphrase generation through the integration of human evaluation, inter-annotator agreement analysis, and readability metrics. The results indicate that although SALAC algorithms successfully maintain semantic integrity and coherence, enhancements are required to tackle domain-specific issues. Chapter 7 examines RQ6 and RQ7.
- Building on the current trends in NLP focused on detecting AI-generated text, this research takes a novel approach by specifically targeting the identification of paraphrased paragraphs generated using LLMs. A significant computational effort is applied to process and analyse the results, providing new insights into the effectiveness of detecting paraphrased content at the paragraph-level. This contribution offers a deeper understanding of how LLMs are effective in paraphrase detection, making it a valuable advancement in the field. Chapter 8 investigates RQ8 and RQ9.

1.8. Thesis Outline

- **Chapter 1:** Introduction: This chapter outlines the research problem, discusses the motivations behind the study, defines its scope, and presents the research questions, research objectives, and contributions.
- **Chapter 2:** Background: This chapter provides an overview of the main technical concepts relevant to the research. It introduces key methods in natural language processing, including conventional machine learning, deep learning, and large language models, to establish the theoretical foundations of the study.
- **Chapter 3:** Literature Review: This chapter reviews related work with a focus on plagiarism detection (PD) and paraphrase identification (PI). It highlights the main approaches, discusses their strengths and limitations, and identifies the research gap that this thesis seeks to address.

- **Chapter 4:** Methodology: This chapter outlines the methodology used to address the research questions of the thesis, discussing various data sources and the selected detection algorithms. Following that, the performance evaluation metrics, ethical considerations and the research's conceptual framework are presented.
- **Chapter 5:** A Paraphrase Identification Approach in Paragraph Length Texts: This chapter focuses on analysing the text length in terms of affecting the accuracy of ML and DL algorithms. The experimental study applied on short, mid, and long text as in sentences, paragraphs, and documents, respectively. In addition, it explains which features are more suited for each text length. The chapter includes sections on pre-processing, feature extraction, classification approaches, as well as the results and their discussion.
- **Chapter 6:** Dataset Creation and Evaluation: This chapter presents an in-depth analysis of the creation of the ALECS-SS dataset. It details three algorithms designed to reorder the sentences in paragraphs. Following this, the chapter explains the masked approach used to lexically modify the paragraphs before the dataset evaluation, which encompasses both human and automatic assessments. The chapter concludes with a thorough discussion that addresses the research questions 3,4, and 5 proposed in this thesis.
- **Chapter 7:** Multi-Domain Evaluation of Auto-Paraphrase Generation at Paragraph-Level: Insights for Education and Plagiarism Detection: This chapter explores the validity and adaptability of SALAC algorithms across many disciplines by evaluating their effectiveness in generating high-quality paraphrases in diverse domain contexts, which will be utilised in assessing the PI approaches in the next chapter. The study examines the influence of domain-specific linguistic variants on paraphrase quality, coherence, and semantic preservation through the analysis of both human and automated evaluations. The findings highlight the strengths and limits of SALAC algorithms, emphasising their relevance in educational contexts while identifying areas for enhancement to boost their robustness across various domains.
- **Chapter 8:** A Comparative Study on Identifying Human-Written vs. Machine-Generated Paraphrases Using Pre-Trained Models and Paragraph-Length Texts: This chapter seeks to explore the efficacy of both autoencoder-based and autoregressive LLMs in differentiating human-written paragraphs from those that have been machine-paraphrased. In contrast to Chapter 5, where detection algorithms operate by comparing

two texts—the source and the potentially paraphrased version—this chapter focuses on training LLMs to independently classify human-authored and machine-paraphrased text without relying on paired comparisons. This shift in methodology emphasises the models' ability to discern patterns and nuances intrinsic to the text itself, thereby addressing a more challenging and realistic scenario in PI tasks.

- **Chapter 9:** Conclusion: Summarising the main contributions and findings in this thesis.

Finally, the workflow of the thesis is illustrated in Figure 1.1.

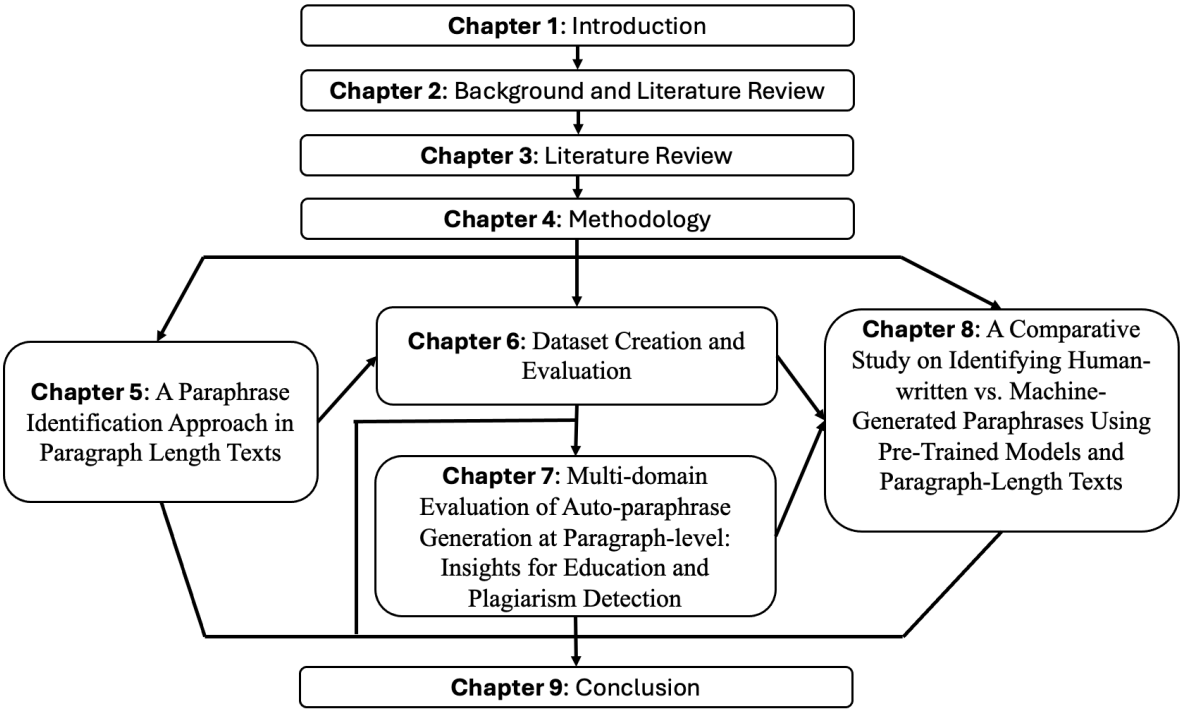

Figure 1.1 Thesis summary workflow

CHAPTER 2: BACKGROUND

2.1. Introduction

This chapter presents a comprehensive background which is critical for gaining a thorough understanding of the subject matter discussed throughout this thesis. It outlines the main theoretical background and provides brief definitions of the concepts related to the NLP approaches: 2.2. Text Pre-processing, 2.3. Text Representation, 2.4. Machine Learning, 2.5. Deep Neural Network and 2.6. Transformer Learning Models.

From the computer science point of view, NLP is a field of study and application that investigates how computers can interpret and modify natural language text in order to perform useful tasks (Chowdhury, 2005) . NLP approaches consist of learning and understanding words and recognizing the patterns in which they occur (Yin To et. al.., 2020). There is a wide range of NLP applications in a variety of sectors of study, such as sentiment analysis, question answering, plagiarism detection, semantic similarity, and paraphrased identification. These important applications could be trained as text classification tasks.

Text classification aims to automatically classify documents or texts into predefined categories according to their content (Kowsari et al., 2019). Text classification can be divided into two main branches supervised and unsupervised approach. According to Gupta (2011), information about the right classification provided by an external mechanism is vital in a supervised approach, whereas an unsupervised approach does not need an external reference. Another significant point of text classification is the number of categories that are considered in the task. Let us suppose that we have a set of documents defined as:

$$D = \{d_1, d_2, \dots, d_n\}$$

A set of categories is defined as:

$$C = \{c_1, c_2, \dots, c_m\} \text{ Where } n > m$$

If each document in D is mapped to only one label in C, this is called a single-label classification. Otherwise, it is referred to as multi-label classification (Vijayan et al, 2017). In this thesis, PI is implemented as a single label classification task where $C = \{0,1\}$

2.2. Text Pre-processing

Written text needs to be pre-processed before being used for classification purposes. This step generally aims to remove uninformed, or noise (misspelling, slang, etc.) content before converting the text into numerical data. Pre-processing steps include many techniques that should be selected depending on the data used and the nature of the problem. Firstly, it splits a string of text into distinct words, phrases, symbols, or other meaningful elements called tokens. Then, removing irrelevant punctuation and stop words, which are commonly used words like {'a', 'in', 'the', 'about'}. There isn't a single list of stop words that applies to every NLP task; in this thesis, the stop words list constructed by natural language toolkit (NLTK) in Python is used.

Additionally, the process involves normalization, which is converting all letters to a lower case, then lemmatizing each word. As some words can appear in various forms (e.g., singular and plural noun forms) while conveying the same semantic meaning, lemmatizing replaces each word to its meaningful root form depending on its context and part of speech (Korenius et al., 2004). In the pre-processing stage, each word stem to be represented by its base form; for instance, the base form of the word 'kindness' is 'kind'. The WordNet Lemmatizer that uses WordNet's built-in morphy function is implemented in this thesis, which returns the word unchanged if it does not exist in WordNet.

2.3. Text Representation

After cleaning the text, a crucial step called text representation is applied, which converts unstructured text into a structured feature. In this step, the text is represented by numerical vectors that enable the machine to understand and analyse the text and then extract useful information from it. In NLP, there are different approaches to vectorizing text, such as bag-of-words (BOW) which is a traditional method, and word embedding (Mikolov et al., 2013) and transformer models (Patil et al., 2023) which are deep learning-based representations. In the next subsections, the text representation methods utilized in this thesis are explained.

2.3.1. Traditional Methods

Bag of Words (BOW)

The BOW model represents a text using its individual words (1-grams), ignoring their order. It generates a vocabulary from a corpus of documents and tracks the frequency of each target word (Z. Liu et al., 2023). This simple model is easy to create and represents text through vectors with a fixed dimension, corresponding to the length of the vocabulary featured across all target documents. It generates flat vectors resulting in the loss of the source text structure, including sequences and word order. Enhancing BOW with n-grams (commonly 2-grams and 3-grams) adds depth to the representation, allowing the model to capture more contextual information and nuanced patterns within the text compared to using only single words. For the classification task, it considers documents to be similar if they exhibit a comparable distribution of specific words (Guozhu & Liu, 2018). Consequently, it does not account for the semantic or contextual aspects of the text.

Term Frequency-Inverse Document Frequency (TF-IDF)

The most common BOW technique is called TF-IDF, where TF refers to the number of times each text word appears in a document, and IDF refers to the number of times that word appears in the corpus. This results in the important terms being given more weight. In essence, words that are used more often are given less weight, while words that are used less often are given more weight. It helps to figure out how important each term is in a document. The main limitation of TF-IDF is that it fails to consider the similarity between words in a document because it treats each word as an independent index. Moreover, TF-IDF has a high sparse dimensionality. To reduce dimensionality, it is necessary to remove irrelevant features while keeping features that are important to achieve the target task (Forman,2003). This leads to the development of more complex models called word embedding.

2.3.2. Deep Learning-based Representation

Word embedding involves mapping each word to an N-dimensional real-valued vector (Bengio et al., 2000). Thus, words appearing closer in a vector space are expected to have similar meanings. In other words, words that are commonly found together in related contexts within a corpus tend to have vectors that are closer to each other. Several pre-trained models of word

embedding have been proposed to convert single words into interpretable inputs for ML algorithms such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington, Socher and Manning, 2014), FastText (Bojanowski et al., 2017) and the state-of-the-art transformer language models. These models, such as bidirectional encoder representations from transformers (BERT), applied a dynamic embedding technique where each word is represented based on the entire input sequence resulting in allowing the word's vector to vary depending on its context.

In this thesis, word embedding and sentence embedding methods, namely Word2Vec and Sentence-BERT (SBERT) (Reimers & Gurevych, 2019) are applied to calculate sentence vector (Chapter 5), BERT or a sub-version of it (Chapters 4-6).

Word2Vec

Word2vec is the most popular word embedding method developed by (Mikolov et al., 2013). It is semantically distributed representation of words into fixed-length dense and continuous-valued vectors based on a vast corpus of literature (Mikolov et al., 2013). This representation is generated by a three-layer neural network that considers the context of the word. The words and their surrounding contexts are mapped into a reduced-dimensional space, typically around 300 dimensions, with each word being associated with a vector (Church, 2017). Semantic similarities are gauged by measuring cosine distances within a reduced-dimensional matrix generated by Word2vec. These distances produce values ranging from -1 to 1, with values closer to 1 indicating stronger semantic similarity. There are two methods for training word2vec, which are continuous bag-of-words (CBOW) and Skip-Gram (Hunt et al., 2019). Simply, the CBOW model aims to predict a word based on its preceding words, whereas Skip-gram predicts words that are likely to appear near each word. In addition, the number of the neighbour's words that are considered during training can be controlled by adjusting a parameter called a sliding window (Suleiman et al., 2017).

Bidirectional Encoder Representations from Transformers (BERT)

BERT embeddings represent an important advancement in NLP, especially in understanding the context and meaning of words in a sequence. In contrast to conventional word embedding methods like Word2Vec, which produce fixed representations for words, BERT generates contextualised embeddings. This means that the representation of a word changes depending

on its surrounding words, which allows for a deeper understanding of its meaning in different contexts.

One of the defining features of BERT embeddings is their bidirectional contextualization. Unlike models that read text in a single direction, BERT processes sequences simultaneously from both left to right and right to left. This approach allows the model to capture the full context of each word, resulting in more accurate representations (Devlin et al., 2019). Furthermore, BERT employs a method called WordPiece tokenization. This technique breaks down words into smaller sub-word units, which helps the model effectively manage rare words and improves its ability to generalise across different word forms. This feature solved the out-of-vocabulary limitations commonly encountered in conventional DL word embedding methods.

Moreover, the foundational structure of BERT is founded on the transformer model which uses self-attention mechanisms to determine the relative importance of words in a given sequence. BERT improves its understanding of the text by giving more weight to important words and less weight to less important ones, resulting in enhanced contextual embeddings and contributing to BERT's overall effectiveness in NLP tasks (Devlin et al., 2019).

The method of obtaining sentence embeddings from BERT starts with correctly formatting the input text. This involves inserting special tokens, namely the [CLS] token at the beginning and the [SEP] token at the end of the sequence. The [CLS] token serves as a collected representation of the entire input sequence, encapsulating the sentence's overall semantics. The extraction of sentence embeddings involves accessing the output associated with the [CLS] token. This token is represented as a dense vector that captures the full meaning of the input sequence by aggregating information from all the processed tokens. While the standard approach is to use the representation of the [CLS] token, alternative methods, such as calculating the mean or weighted sum of all token embeddings, can be utilised depending on specific task requirements (Devlin et al., 2019).

Sentence-BERT (SBERT).

SBERT is a modified version of the pre-trained BERT model that leverages Siamese and triplet network architectures to produce semantically meaningful sentence embeddings (Reimers & Gurevych, 2019). In a Siamese network, two identical neural networks are used to generate

embeddings for different inputs, enabling the comparison of their similarity in a shared space. A triplet network extends this approach by comparing a text, a similar (positive), and a dissimilar (negative) inputs. This technique optimises the embedding space to minimise the distance between similar pairs and maximise it between dissimilar ones, which helps the model learn to produce embeddings that reflect the semantic relationships between sentence pairs. These embeddings are then compared using cosine similarity, which measures the angle between two vectors to determine their degree of similarity, regardless of their magnitude (Reimers & Gurevych, 2019). This method improves efficiency in sentence comparison tasks, making it suitable for applications like semantic search and PI. Subsequent research has explored both interpretability (Opitz & Frank, 2022) and efficiency improvements through layer pruning (Shelke et al., 2024), showing that SBERT continues to evolve while remaining widely used.

Specifically, SBERT uses a pooling strategy to generate a fixed size embedding from the BERT model's output. While BERT produces contextualised embeddings for each token in a sequence, SBERT pools these embeddings into a single vector representing the entire sentence. Common pooling techniques include calculating the mean or maximum of the token embeddings, ensuring that the final embedding encapsulates the entire semantic information while preserving computational efficiency.

2.4. Machine Learning

ML is a major area within artificial intelligence (AI) that involves developing statistical models and algorithms that enable computers to analyse data, learn from it, and make decisions. Unlike traditional programming, which relies on explicit instructions to perform tasks, ML models extract patterns from data to generate insights and draw valid conclusions. Common methods employed in ML include decision trees, support vector machines (SVM), logistic regression (LR), k-nearest neighbours, and neural networks, with the choice of method depending on the specific requirements and characteristics of the task. In PI, the goal is to assign input data to a predefined category or label. Therefore, the success of a model heavily depends on the quality and quantity of data, as well as the selection of relevant features. In this thesis, SVM and LR are applied for the experiments discussed in Chapter 5.

SVM

SVM is a supervised ML approach that is widely used for classification tasks. It operates as a non-probabilistic linear binary classifier, meaning that it aims to find the optimal boundary that best separates the data into two categories. The key idea is to identify a hyperplane that maximises the margin between the two classes, ensuring the best possible separation of data points (Vapnik, 2013). Figure 2.1provides a visual summary of SVM, illustrating support vectors, hyperplanes, and margins.

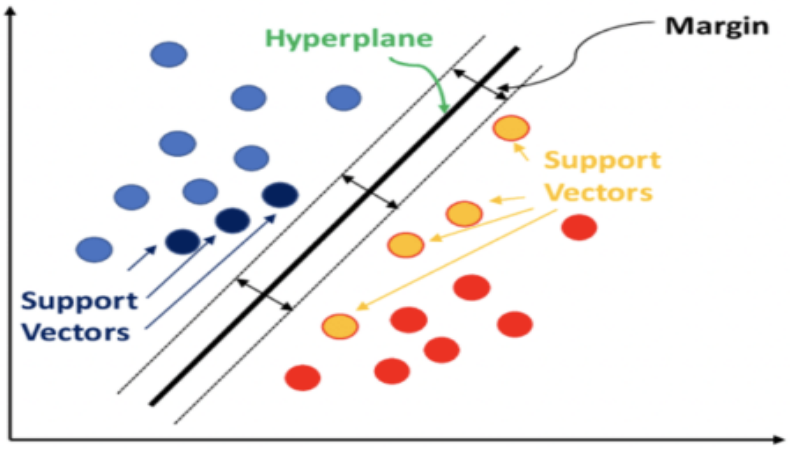

Figure 2.1 SVM

SVMs can handle non-linear classification problems using kernel functions, which transform the input data into higher-dimensional spaces. This transformation allows the SVM to find a linear hyperplane in the transformed space, even if the original data is not linearly separable. Common kernel functions include linear, polynomial, and radial basis functions, which are exponential in nature. By translating the data into a higher-dimensional vector space, SVMs can effectively distinguish between classes that are otherwise inseparable in the original lower-dimensional space (Saputro et al., 2019).

LR

Despite its name, LR is not a regression algorithm but a classification algorithm, frequently employed for binary classification tasks such as PI and PD. LR utilizes a logistic function to support ML processes that classify binary outcomes. This method is particularly effective in scenarios where the goal is to distinguish between two distinct categories or classes. By applying the logistic function, LR models can predict the probability of a particular outcome based on input variables. This approach is widely used to identify and analyse relationships or

comparisons between different variables, providing clear insights into how changes in one variable may impact the likelihood of a specific outcome (Hunt et al., 2019).

Shallow Neural Network

Neural networks consist of nodes (neurons) that work together, mimicking the interconnectivity of the human brain, representing a single-layer perceptron as in (Dagan et al., 1997) or a multi-layer perceptron (MLP) as in (Ruiz & Padmini, 1997). The number of layers selected depends on the task where single-layer perceptron is used for its ease of implementation and multi-layer perceptron is more advanced (Korde, 2012). If the neural network has one hidden layer, it is called a shallow neural network, as in Figure 2.2.

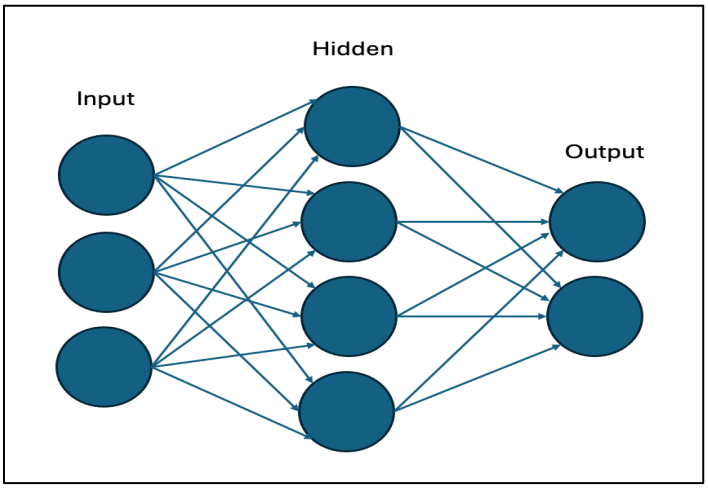

Figure 2.2 The basic structure of the neural network

2.5. Deep Neural Network

Unlike traditional ML methods that rely on manually crafted features, DL algorithms can automatically learn complex patterns from raw data inputs (Figure 2.3). This ability is particularly useful for handling unstructured data like images, audio, and text. Additionally, while ML models are generally easier to interpret and require fewer computational resources compared to DL models, DL excels in tasks where complex patterns and large datasets are involved. However, DL models typically demand significant computational power, often necessitating specialised hardware such as GPUs or TPUs due to their complex architectures and intensive training needs. Both ML and DL are rapidly advancing fields, driving innovations

across various sectors and enabling more sophisticated applications in AI, particularly in tasks involving classification like PD and PI.

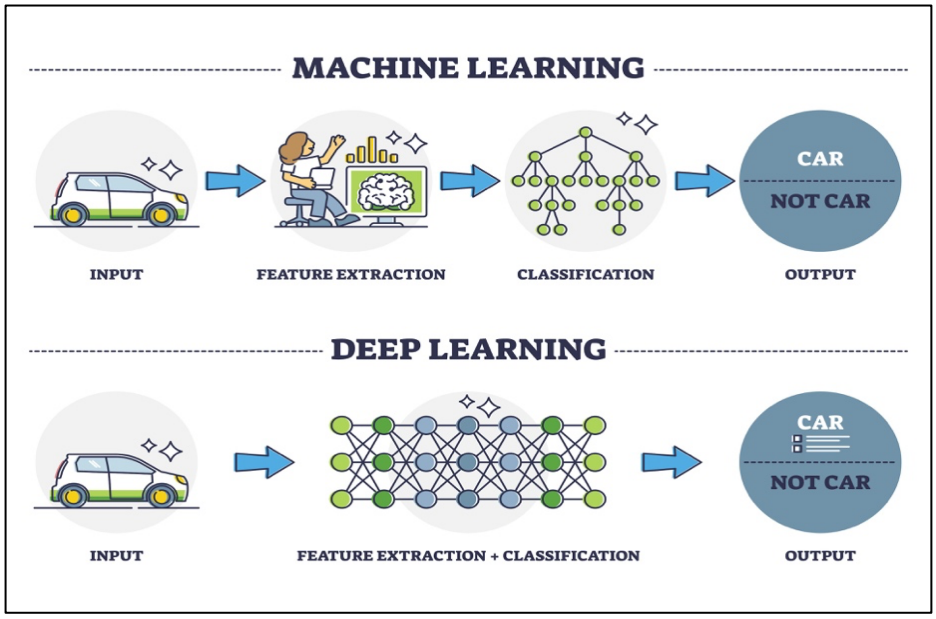

Figure 2.3 An overview of the difference between extracting features in traditional ML and DL. The Figure adapted from Turing.com[1].

The most fundamental architecture of deep neural networks that consist of more than one hidden layer are the recurrent neural network (RRN) (Servan-Schreiber et al., 1988) and convolutional neural network (CNN) (Lecun et al., 1998). Although CNNs were originally built for image processing, they have also been effectively used for text classification (Yin & Schütze, 2015). RNNs are powerful sequence models, and they represent one of the most widely used neural network architectures for language processing (Yin To et.al, 2020). Long-Short-Term Memory (LSTM) (Hochreiter & Jürgen, 1997) and gated recurrent units (GRUs) (Chung et al., 2014) are developed to leverage the shortness of RNN, which is losing long-term dependencies (Plesiak et al., 2020). The ability of DL algorithms to model complicated and non-linear relationships in data is critical to their effectiveness (LeCun et al., 2015) Generally, each neural network design has unique benefits and is chosen according to the particular needs of the required task.

[1] Turing.com. (n.d.). *Ultimate battle between deep learning and machine learning*. Retrieved April 7, 2025, from https://www.turing.com/kb/ultimate-battle-between-deep-learning-and-machine-learning

2.6. Transformer Learning Models

In contrast to RNNs and CNNs, transformer models have revolutionised NLP by introducing a groundbreaking modelling paradigm that eliminates the need for recurrence and relies exclusively on a specialised form of attention known as self-attention with an encoder-decode architecture (Figure 2.4). Vaswani et al. (2017) emphasise the significance of the attention mechanism in their paper, stating that self-attention is a mechanism that connects various positions within a sequence to generate a representation of the sequence (Vaswani et al., 2017). While recurrent models process data sequentially, the transformer uses self-attention mechanisms (orange blocks in Figure 2.4) to assess and weigh the importance of each word in a sequence relative to all other words simultaneously. This allows the model to capture long-range dependencies and relationships within the data more efficiently. By avoiding recurrence, transformers can parallelise computations across the entire input sequence, significantly enhancing processing speed and scalability, particularly for tasks involving large datasets and long sequences. Thus, as this thesis study focuses on paragraph-level paraphrase identification with a large dataset, transformer models were implemented, and the results were analysed in Chapters 5 and 6.

In terms of the encoder, it is responsible for processing input sequences by extracting features and capturing the relationships across the sequence to create a comprehensive representation of the input data (Cho et al., 2014). On the other hand, the decoder uses similar mechanisms to generate output sequences based on the encoded representations received from the encoder (Cho et al., 2014). From Figure 2.4, it is important to recognise that the decoder is employed for generating word predictions, whereas the encoder is utilised for acquiring textual representations or embeddings.

Additionally, transformer models aggregate knowledge from pre-trained data networks. This innovative approach to transfer learning eliminates the need to train the model from scratch with vast amounts of data (Arase & Tsujii, 2021). This approach leads to splitting the model into a body and a head. The body is responsible for learning general features from the input data, and these learned patterns are not specific to any task but can be applied across various tasks. The head, on the other hand, is a task-specific network that uses the features extracted by the body to perform particular tasks, such as classification. Therefore, the head is customised to meet the specific requirements of the task at hand; for example, in a classification

task, the head might be a fully connected layer that outputs class probabilities. As a result, the model can identify patterns using a smaller amount of data and previously acquired knowledge (Z. Yang et al., 2019). The transformer models used in this thesis (Chapter 6 and Chapter 7) are explained in the following sections.

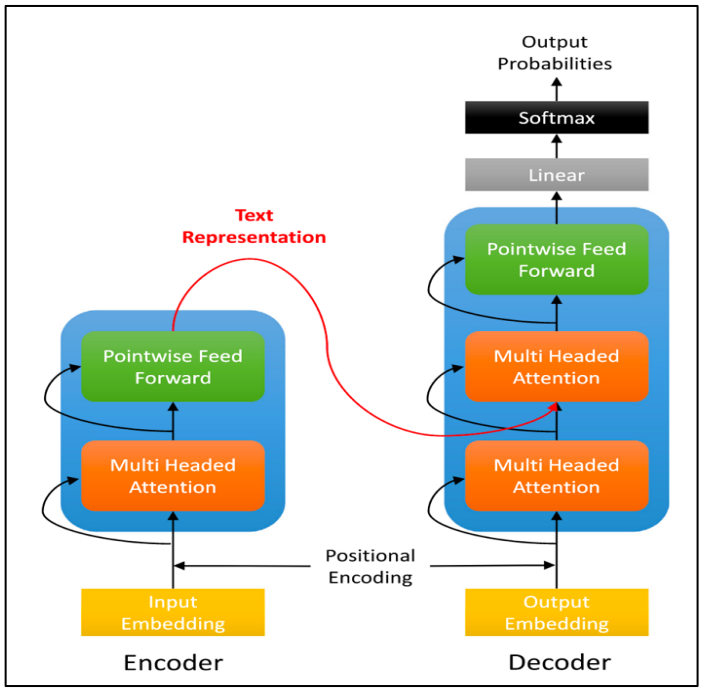

Figure 2.4 Self-attention with an encoder-decode architecture sourced from (Vaswani et al., 2017)

2.6.1. Auto-encoder LLMs

LLMs such as BERT, robustly optimised BERT pretraining approach (RoBERTa), and long-document transformer (Longformer) consist of encoder-only transformer infrastructure trained using a bidirectional approach that considers both the left and right contexts of each token simultaneously. This is achieved through the masked language modelling (MLM) objective.

MLM is a crucial technique used in training auto-encoder language models. In MLM, a portion of the input tokens in a sentence is randomly masked after tokenizing the input text. These selected tokens are replaced with a special [MASK] token, a random token, or left unchanged, according to a predefined probability distribution (usually 80%, 10%, and 10%, respectively). Subsequently, the model analyses this modified input and attempts to predict the original tokens. This process allows the model to learn contextual representations of words by considering tokens on both sides of the masked token to make predictions. By predicting the

masked tokens based on the surrounding context and through iterative training on large text corpora, the model learns to develop deep contextual embeddings. Thus, the model reconstructs the corrupted input with accurate and contextually relevant representations, capturing both syntactic and semantic variations in language.

Bidirectional Encoder Representations from Transformers (BERT)

BERT was introduced in 2018 by researchers from Google AI. It is a pre-trained transformer language model. It is a context-based model that reads the entire input sequence while considering both the left and right contexts (Devlin et al., 2019). Thus, the generated vectors capture contextual information from both directions. Additionally, BERT utilises WordPiece embeddings consisting of 30,000 tokens to tokenise input sequences. Each sequence starts and ends with special tokens ([CLS]) and ([SEP]), respectively. During the training stage, BERT employed a next-sentence prediction task where the model is given pairs of sentences and learns to determine whether the second sentence contextually follows the first in the training dataset. Moreover, BERT utilised MLM in a static manner (Figure 2.5), the positions of the tokens to be masked were determined during the preprocessing stage and remained fixed for the duration of training. The model aims to predict these masked tokens by considering the surrounding context (Devlin et al., 2019).

BERT architecture is available in two versions: BERT-base and BERT-large, each tailored for different scales of tasks and data. The BERT-base configuration includes 12 layers, each layer having 768 hidden states, distributed across 12 attention heads, and totalling 110 million parameters. In contrast, BERT-large is designed with 24 layers, each layer containing 1024 hidden states, spread across 16 attention heads, and comprising a total of 340 million parameters. These variations in architecture enable BERT to accommodate a range of complexities and requirements in NLP tasks, from standard applications to those demanding larger-scale processing and understanding.

A Robustly Optimised BERT Pretraining Approach (RoBERTa)

It is built on BERT's structure as an enhanced version realised by Facebook. RoBERTa includes several adjustments to the pretraining process, resulting in improved performance on downstream tasks compared to BERT. This involves increasing the size of the training dataset, increasing the batch size, and removing the next sentence prediction task (Y. Liu et al., 2019).

Additionally, RoBERTa uses a dynamic masking process (Figure 2.5), which differs from BERT's static masking strategy. In this dynamic method, a different mask is applied each time a sequence is fed into the model during training. This change helps the model gain a more comprehensive understanding of the text.

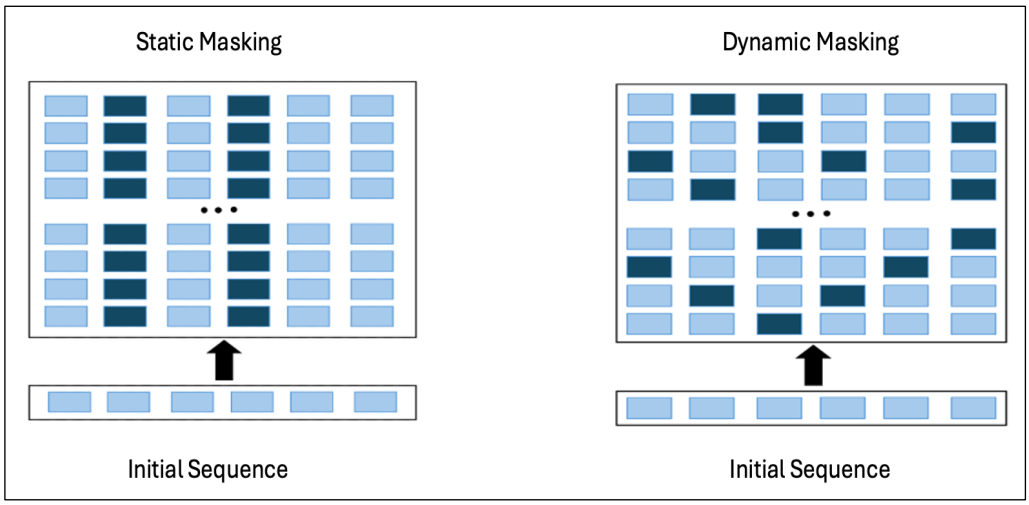

Figure 2.5 Static and Dynamic masking approaches

The Long-Document Transformer (Longformer)

It is developed with a novel approach to process long documents and text sequences, overcoming the drawbacks of traditional transformer models like BERT and RoBERTa. These models struggle with processing long sequences due to their full self-attention mechanism. Longformer addresses this limitation by introducing several significant changes that enable the model to handle long-range dependencies efficiently. Specifically, Longformer replaces the dense, full self-attention mechanism used in standard transformers with a sparse attention mechanism. This modification allows Longformer to attend to a fixed number of tokens regardless of input length resulting in significantly reducing computational and memory requirements.

Moreover, Longformer integrates global attention with sliding window attention to balance the capture of both local and global dependencies. Global attention allows specific tokens to attend to others across the entire sequence while sliding window attention enables each token to focus on its immediate context within a local window (Beltagy et al., 2020). This approach enforces sparsity in the attention matrix by applying masks: local tokens are restricted to attending only to nearby tokens within their window, while global tokens maintain access to

all tokens and are equally accessible to them. These features make Longformer an ideal choice for our research, particularly in evaluating its efficiency for paraphrase generation and the identification of long texts, such as paragraphs.

2.6.2. Autoregressive Language Modelling

An autoregressive language model is a type of statistical model used in NLP that predicts the probability of a sequence of words or tokens based on the previous tokens in the sequence. The fundamental principle behind autoregressive models lies in their sequential prediction capability. Each token's prediction depends on the model's internal state, which encapsulates information from previous tokens. This iterative process continues until the entire sequence is generated or a predefined stopping criterion is met. There are two manners of token prediction: forward or backward product (Figure 2.6). In Forward product, given a text sequence $X=(X_1,\cdots,X_T)$, the probability of the entire sequence X is the product of the probabilities of each word X_t given all the previous words (Equation 2.1).

$$\rho(X) = \prod_{t=1}^{T} \rho(X_t|X_{<t}) \tag{2.1}$$

Alternatively, the backward manner predicts each word based on all subsequent words in the sequence, starting from the last word X_t, Equation 2.2.

$$\rho(X) = \prod_{T}^{1} \rho(X_t|X_{>t}) \tag{2.2}$$

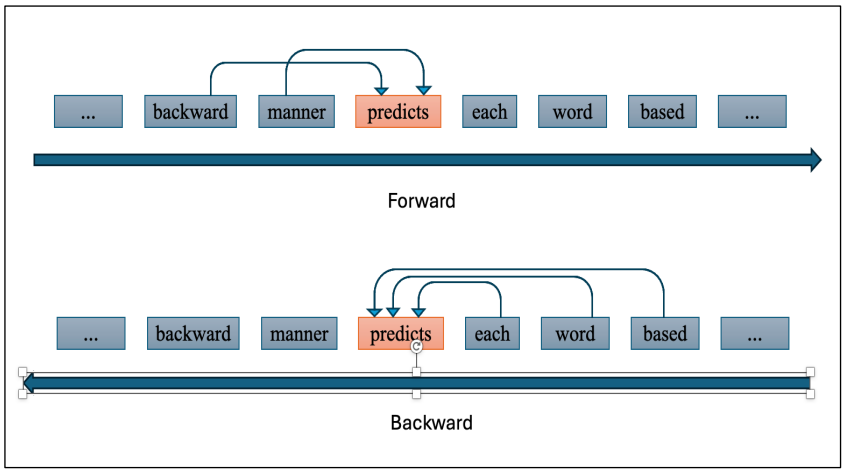

Figure 2.6 Token prediction in autoregressive models

Modern advancements such as transformer-based models like GPT have significantly enhanced the effectiveness of autoregressive language modelling by employing self-attention

mechanisms to capture long-range dependencies more efficiently. Autoregressive language models play a pivotal role in advancing various NLP applications, offering robust solutions for generating coherent text and understanding the intricate structures of natural language sequences.

Generative Pre-training Transformer Models (GPTs)

GPTs specialise in predicting the subsequent token within a sequence. Their self-attention mechanism employs forward autoregressive causal self-attention, allowing the model to focus solely on tokens preceding the current one and preventing future tokens from influencing the prediction. This feature makes GPTs particularly suitable for tasks involving generation. In the next paragraphs, a brief background on the latest developed GPTs, including ChatGPT is reported.

ChatGPT

ChatGPT revolutionised conversational AI when it was introduced in November 2022. This advanced language model, based on OpenAI's GPT-3 architecture, demonstrated remarkable proficiency in understanding and generating text, quickly attracting a vast user base within weeks of its launch. With its ability to understand context, generate human-like responses, and adapt to diverse conversational styles, ChatGPT set a new standard for chatbots and virtual assistants. Additionally, researchers are exploring its capabilities in NLP tasks (see section Chapter 3).

For more details, GPT-3 is huge, trained with 175 billion parameters using a vast dataset comprising 45 Terabytes of text (Brown et al., 2020). Developers noticed that increasing the model parameters improves its ability to understand natural language. Moreover, GPT-3 learned from multiple sources, such as web pages, scaling in three dimensions: model size, pretraining data, and pretraining computation. This enables it to perform well on unseen tasks without requiring specific training.

GPT-3.5

Building upon the success of GPT-3, OpenAI later produced GPT-3.5, an improved iteration that addressed certain limitations of its predecessor in terms of following user instructions and avoiding generating harmful text. Additionally, GPT-3.5 offered enhanced performance and

fine-tuned capabilities, further solidifying OpenAI's position at the forefront of NLP research. Furthermore, GPT-3.5 was trained on both code and text.

GPT-4

OpenAI continued to push the boundaries of AI with the release of GPT-4 in March 2023. This advanced iteration of the GPT series represented a significant leap forward in artificial intelligence capabilities. Unlike its predecessors, GPT-4 integrated not only text but also image inputs, significantly expanding its utility across diverse domains. This integration allowed GPT-4 to process and generate responses based not just on textual data but also on visual information, enhancing its ability to understand and interact with multimodal inputs in various applications.

2.7. Summary

This chapter outlined the theoretical background of this research, beginning with the role of NLP and its applications in tasks such as classification, plagiarism detection, and paraphrase identification. It then reviewed text pre-processing and representation methods, moving from traditional approaches like Bag-of-Words and TF-IDF to more advanced embeddings such as Word2Vec and SBERT. Machine learning techniques, including SVM and logistic regression, as well as shallow and deep neural networks, were discussed. Finally, the chapter examined transformer-based models, including BERT, RoBERTa, Longformer, and GPT, which represent the current state of the art. Together, these foundations provide the methodological basis for the previous work discusees in the following chapter and the novel proposed methods in the rest of this thesis.

CHAPTER 3: LITERATURE REVIEW

3.1. Introduction

The ongoing wave of technology innovations and easy access to vast amounts of information, content, and paraphrasing tools have affected education and academic integrity. In other words, it leads to increased plagiarism, which has a direct negative impact on education, journalism, publishing, and other fields. Consequently, the research to cope with the challenge and detect verbatim and obfuscation plagiarism is growing under the PD and PI fields.

This chapter is broken up into sections and covers the studies and datasets that are currently available in the two domains of PD and PI. The classification methods for PD and PI are reviewed in Section 3.2, with a focus on existing studies addressing both intrinsic and extrinsic plagiarism. It mainly explores the approaches and models used specifically for identifying paraphrases at both the sentence and paragraph levels. These methods are categorised into traditional ML techniques and DL approaches, including neural networks and transformer-based LLMs. Additionally, the datasets are evaluated based on whether they are constructed of sentence-level or paragraph-level paraphrasis considering the text length and other factors, Section 3.3. It is worth noting that most recent efforts focus on auto-generated and paraphrasing at the sentence-level while less work investigates the paragraph-level paraphrases and paragraph length as demonstrated by the existing work review provided towards the conclusion of this Chapter.

3.2. Classification methods for PD and PI

3.2.1. Plagiarism Detection (PD) approaches

Plagiarism could be defined as using someone else's written work without citing the original author or claiming that the ideas are your own (Maurer et al., 2006). Plagiarism may further involve the use of considerable chunks from the source without citing the source (Bär et al., 2012). Plagiarism also includes sentence modification where the author uses the original language pattern without giving credit to the source (Ventayen, 2023). Thus, the detection of

plagiarism presents a challenge, requiring advanced methodologies and technologies to identify instances of textual duplication accurately.

According to Ehsan et al. (2019), PD methods are divided into two main categories which are intrinsic plagiarism detection and extrinsic plagiarism detection methods. The intrinsic method is implemented to detect parts of the text that are inconsistent, while the extrinsic method can match suspicious passages in a text to the source(s) detecting exact verbatim copying and paraphrased text (Jirapond Muangprathub et al., 2021). In the following sections, the published research under each of the plagiarism types that employed NLP techniques is reviewed.

3.2.1.1. Intrinsic Plagiarism Detection Methods

Each author has a unique style of writing texts. From this fact, analysing the writing style of an author is used as a technique to detect potential plagiarism within a document. It mainly doesn't require the availability of a document repository; however, sometimes the document can be checked for stylistic changes by comparing it to prior work by the same author. The features are extracted by checking the author's unique writing style and fingerprinting. Mostly, researchers implement the stylometry concept and/or semantic feature to the text segments. Stylometry is the statistical examination of literary style differences between authors (Saini et al., 2021) , which includes, for example sentence length, word frequencies, and sentence structure. While the semantic feature considers information that represents the vocabulary richness and the semantic context of the document (Cheng et al., 2011). This quantitative characteristic can be used to identify similarities and variations indicating plagiarism with different techniques such as n-grams, Vector VSM, stylometric features, and MLP networks (Manzoor et al., 2023). Then, compare the stylistic signature of each document fragment to each other throughout the document or to the estimated signature of the original author using metrics like cosine similarity, ML or DL models (Bensal, 2013) (Manzoor et al., 2023).

In terms of n-gram overlap approaches, the gram could be represented by sequential patterns of characters or words. (Bensalem et al., 2014) divided each document into fragments using the sliding window technique. Then, every n-gram's frequency is determined by taking into account how often it appears in every fragment to build an n-gram class document model. This method is a supervised classification-based approach that is examined on three datasets to

identify if the fragments are original or plagiarised by applying ML algorithms. Other work focused on leveraging the semantic connections among words to reveal hidden patterns by considering the term frequency, seeking to identify notable variations in the writing style of a document (Oberreuter & Velásquez, 2013). Moreover, (Polydouri et al., 2020) extracted 11 features, a collection of statistic and stylometric features, after splitting each document into segments. Then they applied ML classifiers utilising the Scikit-Learn python library, concluding that the random forest classifier brings the highest F1-score when applied to a balanced dataset compared to other classifiers. A balanced dataset refers to a dataset where each class or category of data is represented in approximately equal percentages. Furthermore, (AlSallal et al., 2019) integrated deep latent semantic and stylometric analyses features to identify the author even though the lack of the reference collection. They examined their method on a corpus of English novels with an MLP approach resulted in a high accuracy, namely: 97%.

3.2.1.2. Extrinsic Plagiarism Detection Methods

Extrinsic PD methods rely on comparing a suspicious document to a set of documents selected from a large reference collection. This process involves retrieving all related documents of digital resources including web articles and datasets. These retrieved documents are referred to as candidate retrieval or resource retrieval which serves as the basis for comparison in the detection process. This phase requires to be accurate in a way that minimises computational overhead without missing a related document. Next, in order to determine which portions of the source documents are comparable to which portions of the suspicious document, the comprehensive data analysis phase conducts thorough pairwise document comparisons (Chang et al., 2021). Notably, the second phase (data analysis) cannot identify source documents missed in the first phase (candidate retrieval) (Hagen et al., 2015) which is considered as a limitation of extrinsic PD methods.

A lot of work has been done to enhance one or both of the extrinsic plagiarism phases: candidate retrieval and data analysis. In the case of improving the candidate retrieval stage, (Ehsan & Shakery, 2016) implemented a topic-based segmentation algorithm approach that considers the keywords to segment the texts into fragments. Then another model was applied to retrieve documents with the most similarity segments to the suspicious passages. This approach of considering the term or keywords resulted in missing some of the related

documents when the author uses synonymous words. To address the limitations of previously mentioned approaches, (Roostaee, Sadreddini, et al., 2020) proposed a fusion model that combines concepts and keywords in the candidate retrieval phase. With more focus on dropping the candidate retrieved documents size, they had expanded their research by designing a knowledge-base word-embedding and local weighting technique that took into consideration the synonyms, plural and singular forms, or other tenses of a word. (Roostaee, Fakhrahmad, et al., 2020).

In the context of data analysis, the researchers clarified what type of re-used text they intended to detect which is mostly verbatim plagiarism or paraphrased text. In this section, a brief overview of existing work in detecting literal plagiarism is offered, while the next section covers the research that has been done on obfuscated plagiarism as it recently received more focus from researchers. The objective is to review the techniques proposed in the fields of PD and PI within a single section, to highlight the research gaps that this thesis aims to address.

The research of detecting verbatim plagiarism mainly focuses on extracting the degree of overlap between the suspicious and candidate documents (Sánchez-Vega et al., 2013). The early research compared each sentence in the suspicious document to all the sentences in the resource documents (Karen & Park, 2002). This led to missing the cases where a student copies a part of a sentence. To overcome this limitation, researchers break down the sentences into fragments or words (Lancaster & Culwin, 2004) looking for partial overlap between suspicious and resource documents. The most efficient method in this scenario was the use of n-gram matching that breaks a text into all of its unique words and counts the number of times each word appears (Krisztián et al., 2002). Then distance metrics are used to determine how similar texts are to one another such as Euclidean and Cosine distance. To improve the result, (White & Joy, 2004) implemented a pre-processing step by stemming words and removing stopwords and/or common words.

The most recent works applied TF-IDF and WordNet which detect minor changes and modifications additionally to the exact copy thus we discussed them under the PI approaches section. According to Barron-Cedeno & Vila (2013) and Vila et al. (2014), the modification includes change of word order, addition or deletion and change of modality counts as a type of paraphrasing.

3.2.2. Paraphrase Identification (PI) Approaches

PI as a main part of extrinsic PD measures the lexical, syntactical, and semantic features of natural language concentrated on one or more granularity levels. Word-level where words are compared to determine whether two words are synonyms (Border, 1997), sentence-level using the sentence as the unit of comparison (Yin & Schütze, 2015), and hybrid of word-level and sentence-level (Agarwal, 2018). These studies are mainly examined on MSRP, Quora Question Pairs (QQP) and PAN datasets. Less research was conducted on the paragraph-level where terms and concepts were used for measuring semantic meaning and intra-sentence relations due to the absence of an appropriate dataset. This literature review is focused on traditional and trend techniques that are used to recognise paraphrasing showing that there is a critical gap in the existing literature of paragraph-length and paragraph-level paraphrases identification studies.

Various types of PI methods can be categorised according to the underlying techniques and methods used. Traditional ML techniques such as logistic regression, decision trees, and SVM use labelled paraphrase data to train models with manually constructed features or representations. In addition, the use of neural network designs for PI is becoming popular which includes models such as Siamese networks, RNNs, LSTM, and CNNs. Moreover, transfer learning techniques are emerging as the forefront approach in NLP, where they use massive datasets to fine-tune transform language models that have already been trained to capture the language words and phrases contextualised representations such as BERT, RoBERTa, Longformer, ChatGPT, and GPTs. Accordingly, this section is divided into three subsections which are ML methods, neural network methods, and transfer learning methods. It's worth mentioning that some works might belong to multiple categories concurrently. Nevertheless, they are mentioned under the most appropriate section to avoid needless repetition.

3.2.2.1. Machine Learning (ML)

A more straightforward, comprehensible, and computationally efficient solution is provided by traditional ML techniques, which are frequently used for PI. In some of these works, features are extracted by considering lexical features by utilizing the BOW techniques like n-gram overlap features (Ferreira et al., 2018), metrics like Bleu (Ji & Eisenstein, 2013; Wan et al.,

2006), syntactic features, and semantic features from external knowledge such as the lexical database. It organises words into sets of synonyms and provides semantic relationships between them like WordNet (Nguyen et al., 2019).

Lexical and Syntactic Approaches

(Cordeiro et al., 2007; João et al., 2007) implemented experiments that extract lexical features from text, by applying a variety of metrics such as Bleu, Sim, word simple n-gram overlap, and edit distance which calculates how many characters or word insertions, deletions, and replacements are required to change one string into the other. They evaluated the experiments on the MSRP and the Knight and Marcu Corpus (KMC), where the paraphrased sentence is a shortened or summarised version of the original one. The Sim metric presented the highest accuracy (ACC) after removing the equal and quasi-equal samples from the dataset (Cordeiro et al., 2007). For more investigations on the metrics' efficiency, they defined two types of paraphrasing, which are symmetrical paraphrasing (SP) and asymmetrical paraphrasing (AS). Symmetrical sentence pairs contain the same information, while in asymmetrical paraphrasing, at least one sentence has more information (João et al., 2007). The result showed that the Sim metric is efficient for AS, while the Logisim metrics, based on the theory of exclusive lexical links between pairs of short text, is better for SP. Rather than implementing a specific threshold value to do binary classification, these metrics also were fed as text features extracted from the Webis-CPC-11 dataset, which has longer text samples, to a classifier such as SVM and k-nearest neighbours (Burrows et al., 2013). Regardless of the accuracy of these models, the Sim metric is suitable only for short texts, because of its demands on computing time.

As the mentioned BOW method addresses each word independently leads to a shortcoming in recognizing the word order, synonyms, and sentence structure which are vital textual features in PI (X. Wang et al., 2018). Thus, researchers include syntactic features that consider structural similarity such as parsing and dependency features.

(Ferreira et al., 2018) conducted a study that combined the lexical and syntactical features. In detail, they measured the lexical features from a BOW, and syntactic features from the resource description framework based on dependence tree. It mainly tackled two specific issues: sequences with the same meaning, but different terms, and the word-order problem. Additionally, they evaluated different ML algorithms, namely RBFNetwork, BayesNet, C4.5, and SMO on short text and the results concluded that RBFNetwork and BayesNet algorithms

outperform others, with accuracies of 75.13% and 74.08%, respectively, ACC measures the method's performance discussed in more detail in Chapter 4. Despite the fact that it did not improve overall outcomes, it significantly recognised the meaning of sentences that shared the same words but in a different order.

(Wan et al., 2006) designed an approach that considered 17 syntactic dependency features, to examine their effects on the accuracy of different ML algorithms, namely naive bayes learner, decision tree, SVM, and K-nearest neighbour, to indicate dissimilarity between a pair of sentences. They claimed that dependency and N-gram features enhanced the classifier to recognise falsely paraphrased cases. In addition, avoiding lemmatisation in the pre-processing step was shown to keep the signs of differences in meaning and focus between sentences. However, more of the correctly paraphrased cases were identified as negative, decreasing the overall accuracy of the approach. They evaluated their experiment on a partial MSRP because some cases led to stopping the parsing script. To leverage the limitations of this study, (Ji & Eisenstein, 2013) considered the same features of (Wan et al., 2006) and implemented them on the whole MSRP corpus. Additionally, they developed a metric that computed the discriminability of features between sentences, called term frequency kullback leibler divergence (TF-KLD). It counts the probabilities that appeared on paraphrased and non-paraphrased sentences, to re-weight features before factorisation, to obtain latent representations of the text. It outperformed TF-IDF by 4% in ACC and 1% in F1-score on MSRP, both of each are measure the method's performance discussed in more detail in Chapter 4. Moreover, they combined other features, such as unigram and bigram, overlapping fine-grained features with TF-KLD, which raised the ACC from 72.75 to 80.41. TF-KLD improved discriminatively distributional features while reducing others. However, the main drawback of TF-KLD models is their inability to assign weights to words that are unseen in the training corpus.

(Ji & Eisenstein, 2013) and (Wan et al., 2006) noted the need for more investigating on another dataset to consider long text such as paragraphs, however these studies examined only on the MSRP corpus, where the maximum length of a sentence is 36 words (B. Dolan et al., 2004). In addition, using dependency trees to solve problems restricted an approach to single sentences (Nguyen et al., 2019).

Knowledge-Based and Graph Approaches

Some researchers preferred using ontology-based methods for detecting conceptual relationships and recognising unseen words. These type of methods considers the semantic relationship between different units of texts by utilizing dictionaries in a form of graph-based models (T. Zhang et al., 2019). HowNet and WordNet are the two commonly used dictionaries for ontological relationship which provide textual analysis and similarity comparison between different fine-grained texts (Eisa et al., 2015).

(Ul-Qayyum & Wasif, 2017) suggested an approach called ParaDetect to demonstrate the improved impact of taking WordNet into account. They implemented SVM with semantic heuristic features provided by WordNet to identify paraphrasing. Two techniques are examined, namely monotonic alignment as longest common subsequence and non-monotonic alignment based on part-of-speech (POS) and BOW. The results show that non-monotonic alignment features that supported by WordNet were better than monotonic alignment. However, the result is affected negatively by the high lexical similarity of training samples that leads to classifying true paraphrases as non-paraphrased cases. Additionally, it examined on MSRP which consists of sentence length and sentence-level paraphrases as discussed in section 3.3 below.

Furthermore, the ontology-based methods involved in the fuzzy-semantic method which is a detection approach focused on sentences restructuring within given texts. Many studies implemented fuzzy-sematic techniques for the calculation of sentence similarity index (Alzahrani & Salim, 2010), (D. Gupta et al., 2014) and (Ezzikouri et al., 2017). The sentences matching with original text were given score “1” and ones different from the source were assigned “0”, while the values in the range of “0” and “1” indicate partial presence in the set (Alzahrani & Salim, 2010). In detail, words that occur within the synsets, sets of synonymous words, of each other in the WordNet are assigned a fuzzy value of 0.50. The overall fuzzy similarity between similar sentences is then calculated using word-to-word similarity values.

Moreover, (Mohebbi & Talebpour, 2016) built graphs from the WordNet database with POS tags and used maximum matching parameter for the calculation of the similarity index. They reported the ACC of the proposed method to be 76.88%. In the same context, semantic role labelling (SRL) was proposed by (Osman et al., 2012) where WordNet and POS were applied for calculating the word similarity scores. The SLR method was useful for phrases and words, but they failed to determine the similarity index for relatively large sentences (T. Zhang et al.,

2019). To cope with this issue, structure-based methods were developed by (Vani K & Deepa Gupta, 2017), which compare the text in a paragraph-by-paragraph manner. Although these techniques are helpful for identifying paraphrases based on grammatical structures, they did not take brief sentence structures and word-by-word analysis into account (Alzahrani et al., 2011). In addition, it evaluated on sentence-level paraphrases corpora namely: plagiarised short answers (PSA) and PAN corpus respectively, discussed in section 3.3 below.

From another perspective, the application of graphs for the representation of such relationships was described to be useful for comparing textual features involving grammatical structures, sentence order, and word arrangements within sentences (Chae et al., 2013), and converting unstructured text into structural data (Momtaz et al., 2016). (T. Zhang et al., 2019) proposed graph-based method considered the term frequency and author keywords to construct graph nodes capturing the hierarchical document structure. The maximum common subgraph was used for measuring the similarity of two graphs. This similarity was used to identify the documents with similar themes but using different topic words. Indeed, the corpora used in this experiment were created manually, with certain limitations due to manual data handling and the exclusive focus on the Chinese language. Moreover, (Momtaz et al., 2016) developed a graph-based approach for similarity detection in Persian texts. They created a graph from the unique words in the sentences where each word was represented into a node with edges linked it to 4 words before and after it. After constructing the graphs, they applied an iterative method to identify nodes (words) with similar characteristics. Graphs with comparable numbers of nodes were marked as similar, and the PlagDet score—a comprehensive metric that combines precision, recall, and granularity to evaluate plagiarism detection performance—was then applied to determine the similarity level based on the number of similar graphs found in the texts. They evaluated their PD model on two Persian alignment datasets. They showed that their graph-based model could detect plagiarism with 90% Plagdet score on PAN2015 and 87% on PAN2016. However, it has a notable limitation which is the investigated documents should be used with specific lengths resulted in restricting its applicability to texts of varying lengths. In other words, it cannot be implemented to documents with various lengths. Additionally, the developed approach was tested only on Persian documents, thus there is a need for testing across different languages as well considering paragraph-level paraphrases. Furthermore, (Momtaz et al., 2016) used solely graphs without consideration of the application of neural networks for the detection of text-based plagiarism. The proposed graph-based approach could be enhanced by combining it with the convolutional neural network, which could offer a better

accuracy level in terms of detection of similarity scores in words, phrases and sentences within different types of texts, as shown by some other research cited in the next section.

Word Embedding and Semantic Representation

From different viewpoints, some researchers take the advantage of word embedding pre-trained models into consideration. These models convert each word into a vector to understand its meaning and identify paraphrases that convey similar meanings but may have different wordings or structures. From this perspective, (Vrublevskyi & Marchenko, 2020) concatenated the word embedding features with dependency tree features to show that this combination can be useful for identifying paraphrases. In the same context, (Nguyen et al., 2019) developed an algorithm based on external knowledge and word embedding. They have applied the continuous bag of words CBOW and Skip-Gram models to extract interdependent features based on pre-trained word embedding, namely word2vec. CBOW predicts a target word based on its context, and Skip-Gram does the opposite, predicting context words according to the target word. As a part of the methodology, more features were also included that help to measure semantic relatedness based on external knowledge resources such as WordNet. Features extracted from short sentences with and without pre-processing step. Then, SVM is involved for the classification task. It examined on MSRP, SemEval and P4PIN datasets achieving high ACC 84.17, 83,73, and 95,22, respectively, showing that word embedding offered a new approach to the PI task, obtaining excellent results by addressing the traditional approaches' semantic similarity problem (C. Zhou et al., 2022).

Furthermore, (Kenter & de Rijke, 2015) focused on word2vec and Glove semantic sentence representation. Specifically, they extracted the sentence features by word alignment and word embedding beside the saliency weighted semantic graph. They mainly measured the semantic similarity at the sentence level ignoring the importance of the word order in the PI task. Although applying word alignment to extract syntactic and semantic relations between words of the sentence and feeding them into the SVM model as features yielded significant results, (Kenter & de Rijke, 2015) indicated that their approach exhibited limitations when applied to text exceeding sentence length. Moreover, findings highlighted that concatenating of pre-trained word embedding models obtain better scoring than WordNet-based approaches.

As implementing word embedding shows enhancement of PI methods results, (Vrbanec & Meštrović, 2020 ; Vrbanec & Meštrović, 2021) conducted two studies on different models of

sentence semantic representation such as word2vec, Glove, USE and Fast-Text Their experiments were done on three datasets namely MSRP, Webic-CPC-11 and C&S. Because of the pairs of sentences that are semantically unrelated and very similar lexically, no specific model outperforms others on all datasets, however USE provides high ACC and F1-score on Webic-CPC-11 and C&S. These studies didn't take into consideration the paragraph-length or paragraph-level paraphrases.

3.2.2.2. Neural Networks

DL has several advantages over conventional ML techniques in the areas of PD and PI. DL is superior to traditional methods at identifying intricate patterns and representations in large datasets. This feature makes it possible for DL models to identify minute details in written content, leading to enhanced accuracy in detecting instances of plagiarism and paraphrasing. Furthermore, in comparison to conventional ML algorithms, DL architectures like RNNs and transformers may efficiently exploit contextual information and semantic linkages across phrases and texts, enabling more sophisticated analysis.

In the PI field, researchers employ DL models to detect semantic similarity mainly in short text highlighting the efficiency of DL models over ML on PI tasks. (Hunt et al., 2019) compared the accuracy of two ML models with three different deep neural network models. Their results illustrated that all DL models' accuracies outperform LR and SVM models. The lowest ACC of DL models was obtained by Siamese NN (~62), while the best ACC is (~82) from LSTM.

Furthermore, (Yin & Schütze, 2015) proposed a LR classifier and a new architecture of the CNN model called BI-CNN-MI that applied different sizes of filters on whole and parts of sentence pairs building similarity metrics. The main purpose was to detect paraphrases that only can be recognised at the sentence-level. However, they concluded that removing sentence-level features did not have a significant effect on the overall results when calculating all four lower levels of granularity using N-gram features (n = [1,2,3,4]). Rather than implementing different sizes of filters as a feature extractor and similarity metrics, (Yin et al., 2016) presented the Attention Based Convolutional Neural Network (ABCNN) to extract semantic features considering word-level and phrase-level. Attention is used to learn counter-biased sentence representation. In addition, word2vec is used to represent each sentence vector at the initial step. According to Yin et al. (2016), ABCNN may show a better result if training it on a large

dataset, which is examined on a small dataset namely MSRP. In addition, (X. Zhang et al., 2017) developed a CNN model considering the word, phrase, and sentence semantic measuring levels. Text is represented in a parse tree where the node is the word embedding vector. Then CNN with dynamic filter and MLP are examined resulted in the CNN model can deeply recognise the relation between sentences. This model relied on a highly accurate parser thus it is not appropriate for processing long texts (Nguyen et al., 2019).

In general, CNN is efficient for PI tasks because of its filter technique which can be used as a feature extractor on multiple levels of granularity (word, phrase, N-gram and sentence). On the other hand, CNN needs a huge dataset for training while the existing paraphrase datasets are small, so most of the researchers pretraining their models on an unlabelled text called unsupervised pre-training. This pre-training leads to avoiding overfitting (Yin & Schütze, 2015); however, it demands more resources (X. Zhang et al., 2017).

(R. Yang et al., 2019) suggested a RE2 model construction of N-blocks which starts with encoding sequences after representing it in a word vector. Then, an alignment layer is involved to measure the semantic similarity between the current word and its equivalent on the corresponding position of the second sentence. (B. Li et al., 2021) ignored the word order and took advantage of word alignment information that uses monolingual alignment tools. They considered the unsupervised pre-training method on RE2. Their results outperform the (R. Yang et al., 2019), however their approach examined on short sentences at the word-level by considering the QQP dataset where its mean sentence length is 6 words as discussed later in section 3.3.

In addition, (Chi et al., 2020) utilised an available NLP toolkit to extract the dependency features from pairs of sentences. These features show the relationships between parent and child nodes in a dependency tree, categorizing them into two groups based on whether they are shared or unique across both sentences. This categorization is aimed at representing the similarity and dissimilarity between the sentences. Subsequently, neural networks leverage these features to predict the classification outcomes. Their approach yielded improvements of 1% in ACC and 2% in F1-score compared to methods employing dependency features for detecting sentential paraphrases. They evaluated their method on the MSRP dataset, which consists of short text.

In terms of applying graphs with neural networks, (Huang et al., 2019) developed a novel graph-based neural network outperformed other graph neural network (GNN) models for the classification of textual information in the areas of consumption of memory, extraction of nuanced features from the text in a better way and supporting the online testing. They have used the small text windows, focused on the extraction of key text features from the small text windows rather than building the graphs from the whole corpus. The issue with the graph built for the whole corpus is that they used so many edges and nodes which not only consume a lot of memory but also are not able to catch nuanced information from the individual sub-text within the main text. This was possible due to the application of the message passing mechanism (MPM) which enabled the collection of information from adjacent text nodes and updating information embedded in the original nodes.

Hence, the (Huang et al., 2019) methodology of building graphs from the given text based on the graph-based neural network coupled with the MPM technique can come in handy in processing the natural language. Another important strength of GNN model developed by (Huang et al., 2019) is that it used all words within the text as nodes and creates a link between the texts using the MPM tool, which is better at connecting the words and paragraphs within each in order to extract the fundamental features of text compared to other GNN models. However, the GNN model was developed using the small windows in texts and tested on three non-binary classification datasets. Thus, it needs to be validated specifically on PD and PI tasks.

(X. Liu et al., 2020) Utilised the tensor graph convolutional networks as a tool for the classification of textual material, and focused on extraction of semantic, syntactic, and sequential information based on the contexts of texts. As part of the methodology, they built the tensor graph convolutional networks for extraction of sequential information, followed by development of intra-graph propagation which was mainly employed for aggregation of information from the adjacent nodes within a single graph. After this step, the inter-graph propagation was constructed to create the sequential relationship and harmonisation between the constructed graphs. After extensive experiments on classification benchmark datasets excluding PD and PI datasets, they found that the proposed tensor graph neural network effectively represented, harmonised, and integrated information from various graphs, outperforming other sequential learning models in learning different textual features (X. Liu et al., 2020).

Heterogenous graph neural network (HGNN) as used by (Huang et al., 2019) and (X. Liu et al., 2020) was efficient in extracting the lexical, syntactic, and semantic features of the text represented them on its edges. Additionally, it considered the co-occurrence of words and sentences represented them in different types of nodes. These characteristics of HGNN lead to achieved high performance in text classification, especially in document clustering which tends to determine the document theme. However, it's noteworthy that these approaches have not been examined in tasks related to PI possibly due to the limited availability of datasets containing long texts required for this task.

Although these studies discussed in this section were done on more than one of public paraphrase dataset, they did not take into account the variation in the number of words in each sample, nor the type of paraphrasing applied to different datasets. More importantly, they consider typical samples as paraphrased cases that do not state the paraphrase definition provided in Chapter 1.

3.2.2.3. Transformer Models

Another related strategy in a DL setting is transfer learning, in which the machine gathers up knowledge from a trained data network. This novel way of learning eliminates the method of training the model from scratch with an enormous amount of data (Arase & Tsujii, 2021) like in RNN and CNN models. The model can therefore recognise a pattern using a small number of data samples and previously learned information.

Transfer Learning and BERT

The most common transformer model which is also now used as a baseline in classification tasks is BERT. It builds with the aim of overcoming the limitation with sentence representation models that are based on a unidirectional encoder. BERT consists of an encoder only that was trained by implementing an MLM and next sentence prediction (NSP) (see Chapter 2). BERT is fine-tuned for particular downstream tasks after pre-training on a vast corpus of text data by adding an additional output layer. Its pre-trained knowledge is adjusted through this fine-tuning process to meet the unique needs of tasks such as PI. It has been demonstrated to achieve high outcomes on a wider array of sentence-level and token-level NLP tasks. Specifically in the PI task, it evaluated on MSRP with 89.30% and on QQP with 72.10% of F1-score (Devlin et al.,

2019). This high result raises the machine prediction ACC to be closer to human performance. This efficiency of BERT attracts researchers to explore it.

(Ko & Choi, 2020) implemented multi-task learning to enhance the BERT model's accuracy on the PI task. They additionally applied a whole word mask rather than the original mask which is a sub-word mask. They compared their study to others that used sentence vector representation, showing that BERT outperformed them. As the BERT model presents a generic representation for NLP tasks, (Arase & Tsujii, 2021) considered phrasal paraphrase to develop a transfer fine-tuning called PPBERT. At the input layer, the phrase alignment runs to obtain a set that contains pairs of spans for each source and targets sentential paraphrase pair. By applying mean-pooling, the representation of these pairs is generated and then concatenated to be represented by a single vector. To keep focusing on the efficiency of this representation method, a simple classifier was selected. It evaluated on many sentence pairs modelling tasks including PI. The PPBERT's result on MSRP doesn't outperform (Ko & Choi, 2020), however this method is cost-effective because it only relies on paraphrases having phrase alignments. Furthermore, (Xu et al., 2020) created a model called Lexical, Syntactic, and Sentential Encoding that extracted the dependency structure of paired sentences by applying Stanford Parser and the sentential and lexical features by using BERT. These features were fed into the relational graph convolutional networks (R-GCNs) in order to obtain the sentence vectors needed to determine the semantic similarity of the sentences. They improved the F1-score by 1.0 on QQP and 1.7 on MSRP over the (Devlin et al. 2019). However employing this approach is not feasible for paragraphs due to its reliance on the Parser tool, which is optimised for shorter texts at the sentence length (Nguyen et al., 2019). In a corresponding situation, (Shree & Jayita, 2023) examined the efficiency of BERT as a classifier model compared to ML models. Specifically, they applied TF-IDF for feature extraction and then inputted these features into BERT, SVM, and an ensemble model. The ensemble model combined predictions from Random Forest, Naive Bayes, and SVM classifiers using a majority voting approach. Their findings revealed that BERT yielded superior ACC and F1-score compared to the top-performing ML classifier, surpassing it by 14% and 7%, respectively.

Variants of BERT

As observed, BERT surpasses alternative sentence representation models by producing a contextual vector representation. This capability allows it to differentiate word meanings across diverse contexts, rather than assigning equal importance to each word irrespective of context

(Devlin et al., 2019). Consequently, this breakthrough in DL has led to the development of many other transformer models based on BERT's foundation, which focuses on optimizing the architecture of the BERT model to make it more efficient and effective, such as DistilBERT, RoBERTa, and the lite BERT for self-supervised learning of language representations (ALBERT) models.

DistilBERT was created by reducing the number of BERT's parameters by 40%. The model becomes more lightweight and computationally efficient while still retaining the essential knowledge learned during training, 97%. This reduction in parameters helps in speeding up inference by 60% and reducing memory requirements making the model more practical for deployment in resource-constrained environments (Sanh et al., 2020).

In regards to RoBERTa, (Y. Liu et al., 2019) proposed several enhancements on BERT which include training on a significantly larger dataset, increasing from 16GB to 160GB of training data. They also introduce a dynamic masking pattern, departing from BERT's static masking approach, where different masks are applied to the input tokens in each training iteration. This dynamic masking scheme encourages the model to learn more robust representations by preventing it from relying too heavily on specific masked tokens during training. Additionally, RoBERTa replaces the NSP objective with a strategy that involves training on full sentences, omitting the NSP component. Furthermore, RoBERTa trains on longer sequences, enabling the model to capture more comprehensive contextual information during training. These adjustments collectively aim to enhance the robustness and effectiveness of the RoBERTa model compared to its predecessor, BERT. Consequently, RoBERTa's performance on MSRP and QQP exceeds that of BERT by 1.9% and 0.9%, respectively which are same to the results obtained by ALBERT.

In terms of ALBERT, it factorises the embedding parameters by decomposing them into two smaller matrices. These matrices are then shared across all tokens, significantly reducing the total number of parameters required for embedding. By combining factorised embedding parameterization and cross-layer parameter sharing, ALBERT achieves a substantial reduction in the number of parameters compared to traditional BERT models. In addition, ALBERT uses sentence order prediction (SOP) as an alternative objective of NSP during pre-training. It learns to understand the contextual relationships between sentences and effectively captures the flow of information within a document. This helps the model generate more coherent and contextually relevant representations (Lan et al., 2020).

Researchers investigated these models within the realms of PI and PD as word embedding models that convert each word or sentence into a vector and/or a classifier models. (Vrbanec & Meštrović, 2023) compared the efficiency of BERT-base models on detecting plagiarism in the form of paraphrases evaluated on three PI corpora: MSRP, CS, and Webis-CPC-11, further elaboration on these corpora is provided in section 3.3. Their method focused on sentence representation using different approaches such as TF-IDF, word embedding models and transformer-base models. They represented each sentence into a vector and then used similarity or distance measure to detect paraphrased pairs such as cosine similarity where these embeddings can be paired to assess similarities between texts. They concluded that cosine similarity is the most efficient metric and the BERT family of models, including BERT, RoBERTa and DistilBERT, is highly effective in identifying short text. According to Incitti et al. (2023), Transformer models generate word embedding vectors that are significantly influenced by their context, thus effectively conveying semantic content.

Further work, (Reimers & Gurevych, 2019) developed SBERT which produces semantical sentence embeddings by utilizing Siamese and triplet network structures. Its performance was examined on many NLP tasks including PI by considering the MSRP dataset. The results of SBERT outperformed other word embedding models including Fast-text, Glove and BERT Embeddings in the PI task. In addition, (W. Wang et al., 2019) built StructBERT that introduced novel linearization strategies to incorporate language structures during pre-training. By leveraging both word-level and sentence-level ordering, StructBERT captures sequential dependencies more effectively. As a result, it outperformed most previous models across various natural language understanding tasks. Specifically, it surpasses BERT by 3.3% in the F1-score on the MSRP dataset and by 2.3% on QQP.

From another perspective, (Hany & Gomaa, 2022) examined how the combination of different types of similarity techniques can improve the ML classifier accuracy. They specifically integrated three categories of similarity scores: string similarity, embedding similarity derived from BERT, and semantic similarity utilizing WordNet and spaCy algorithms. They employed 168 string similarity techniques from the Abydos library, including Levenshtein similarity and Damerau-Levenshtein similarity. Additionally, they utilised eight of word embeddings from pre-trained models in the sentence transformers library, such as BERT-base-nli-mean tokens and all-mpnet-base-v2. Finally, they applied linear SVM as a classifier. They concluded that the more features and similarity algorithms are used, the more

advantageous the outcomes will be in the PI task. A related study, (Muneer et al., 2025)investigated the effectiveness of various classification methods in distinguishing between manually and auto-paraphrased sentences using both synthetic and real-world corpora. They proposed three sentence-level benchmark datasets for artificial paraphrases, and one based on real-life texts. The classification task was approached using traditional ML algorithms and transformer learning models where the DL models were excluded due to data limitations. The highest performance was achieved by integrating similarity features from traditional techniques—such as N-gram overlap, WordNet-based measures, and Kullback-Leibler divergence—with Sentence Transformer embeddings. These combined features were used as input to the ML classifiers, and the evaluation, conducted via ten-fold cross-validation, revealed that manual paraphrases remain significantly more challenging to detect, although some machine-generated paraphrases also proved comparably difficult.

In terms of reducing the model complexity, (Raffel et al., 2020) developed text-to-text transfer transformer model (T5) which could be recognised as a hydride model of bidirectional and auto-regressive models, however its architecture and primary usage make it more closely related to the bidirectional models. This transformer model provided a unified format that simplified the implementation of various NLP tasks. In detail, the same model architecture, training procedure, and decoding process can be applied to all tasks making the model more fixable. The text-to-text paradigm allows the researcher to view each task as converting input text into target output text which makes it easier to adapt the model to different types of problems without having to apply adjustments that are particular to each task. They evaluated T5's performance on a wide variety of English-based NLP problems including the PI task. The result on MSRP is 92.8% which surpassed BERT by 3.5% but fell short of StructBERT by 0.8% in terms of F1-score.

Advanced and Generative Models

In the same context, (Palivela, 2021) implemented T5 as a paraphrase generation and identification model. Firstly, they increased the data diversity by eliminating pairs that share over 60% of unigram elements or have very little semantic similarity from the used datasets (QQP and MSRP) before training the paraphrasing model. They configured the PI model hyperparameters using the paraphrase model hyperparameters in a way that avoided the need to retrain the model. They achieved 87.17% and 82.05% ACC on QQP and MSRP respectively, through determining the semantic similarity between sentence pairings after extracting their

sentence vectors. In addition, XLNet is an autoregressive model developed by (Z. Yang et al., 2019). This model is different from BERT because XLNet uses permutation language modelling while BERT implements MLM. XLNet maximises the expected probability over all possible permutations of the input sequence. This indicates that it trains on various permutations of the input sequence considering both the tokens that precede and follow each token in the sequence. The result of implemented XLNet on PI outperformed the BERT on two of the primary PI datasets. In particular, XLNet achieves an F1-score of 92.3% on QQP and 90.8% on MSRP, surpassing BERT's scores by 1% and 2.8%, respectively.

Despite the numerous benefits of pre-trained large models over traditional DL, they still face limitations such as struggling to adapt to new tasks without task-specific training. As a result, researchers have shifted their attention to creating more sophisticated models like generative LLMs which can handle unseen tasks without the need for task-specific training such as GPTs. Consequently, GPTs become state-of-the-art in the fields of AI and NLP. These models, especially the ChatGPT, have shown remarkable capacity in understanding and generating human-like text in a variety of tasks. Thus, GPTs significantly influenced the landscape of NLP and continue to be at the forefront of research and development in NLP.

These models have advanced capabilities in generating text that is fluent, detailed, and exceptionally natural in tone making it increasingly challenging for faculty members to differentiate between human-written and AI-generated content (Abd-Elaal et al., 2022). Due to this ability of GPTs, the researcher focuses on exploring detecting auto-generated text over utilizing GPTs as tools for identifying auto-paraphrases. For instance, (Alamleh et al., 2023) developed a method to differentiate human text from machine-generated text using handcraft features and an ML classifier. Specifically, they implemented TF-IDF and 11 of ML classifiers including SVM, feedforward neural network and BERT. TF-IDF features capture the importance and relevance of individual terms within a given text. They concluded that random forest provides the best ACC with 93.50% on distinguishing between human-written and ChatGPT generated text. For further clarification, (Elkhatat et al., 2021) highlighted the difficulties the auto-generated detecting tools have when trying to distinguish AI-generated content, particularly when using more sophisticated LLMs to create the text. Thus, (Perkins et al., 2023) employed prompting techniques, utilizing OpenAI's tool namely GPT4 through ChatGPT Plus, to generate 22 unique experimental submissions, aiming to examine the challenge of the AI detectors identifying AI-generated content. They concluded that the

Turnitin AI detection tool only detected a mean proportion of 54.8%, despite the fact that AI methods were employed to generate 100% of the material content for all submissions. Additionally, several studies have suggested a significant decrease in ACC when text is paraphrased using automated tools (Anderson et al., 2023; Weber-Wulff et al., 2023; Krishna et al., 2023; Mitchell et al., 2023). Stated simply, detecting auto-paraphrased content is more challenging than detecting auto-generated text.

In terms of PI, there has been limited research conducted with the aim of detecting text that has been paraphrased by humans or machines. (Kim et al., 2024) conducted an assessment of BERT, RoBERTa, and ChatGPT across various classification tasks, including PI. Their methodology involved the application of these models to the QQP and MSRP datasets. The outcomes of their investigation revealed that ChatGPT exhibited the lowest ACC and F1-score on both datasets, whereas RoBERTa achieved the highest performance scores. In detail, they argue that even the most basic BERT model performed much better than ChatGPT in recognising sentences that are semantically identical, with a margin of 24% in ACC and 17.7% in F1-score on MSRP and 2% in ACC and 0.7% in F1-score on QQP. Thus, ChatGPT performed best in simpler text structures, such as user questions in the QQP, where it can accurately identify similarity with results that are comparable to BERT-base accuracy. However, ChatGPT's performance significantly decreased on the more complex MSRP dataset, particularly when contrasted with the fundamental BERT model. When analysed by class, ChatGPT performed as accurately as BERT-base in the “paraphrased” class, but it performed noticeably worse in the “Not Paraphrased” class, suggesting that it is not sensitive to semantic distinctions between sentences. This study presents a thorough examination of ChatGPT's performance on PI task, notably does not acknowledge how the length of the text and the level of paraphrases affect the outcome. To enhance the RoBERTa performance on PI, (Amin et al., 2023) implemented a method which combines the prompt tool explanation of the similarity of sentence pairs to the RoBERTa classifier. The result showed that the combination of ChatGPT and RoBERTa provided better results than implemented RoBERTa individually on binary classification tasks. Expanding on the challenges of paraphrase detection, (Kartelj et al., 2025) conducted a comprehensive study employing a diverse set of classification models on datasets encompassing varying text lengths, ranging from single sentences to full-length documents. The corpora used in this research consist of human-written texts from various domains and their corresponding paraphrased versions generated by GPT models. Interestingly, human-authored texts tended to be longer in terms of both word and sentence count, while GPT-4

occasionally produced longer outputs when the original human-written input was very brief. Their approach incorporated advanced feature extraction techniques tailored to identify paraphrasing patterns, utilizing BOW and character-level n-grams as input features. A total of 19 classifiers were tested, alongside a commercial detection system called ZeroGPT. Results showed that the classifiers achieved high ACC across conditions, with a minimum ACC of 95%, attributed to the distinct output patterns of ChatGPT's autoregressive generation model.

From our point of view, one work (Wahle, Ruas, Kirstein, et al., 2022) has been done to detect auto-paraphrased text at paragraph-length but at sentence-level paraphrasis using SpinnerChief, BERT and GPT3 models. They implemented eight machine classifiers on their datasets concluding that the best F1-scores were extracted by implementing auto-regressive models (T5 and GPT3).

Therefore, DL holds the potential for developing more accurate and advanced PD systems that can adapt to the subtle differences in language and deceptive paraphrasing techniques. However, all the work mentioned in this chapter focused on sentence-level paraphrasis that occurs in short text and showed robust results. Addressing the effect of text length on paraphrase identification and developing a detection algorithm for paragraph-level paraphrases are two of the main contributions of this thesis.

Table 3.1 Outline of previous studies in PI

Source	Method	Features	Classifier	Dataset
Wan et.al, 2006	ML	17 Features Include BLEU and N-gram overlap	SVM	MSRP
Ul-Qayyum and Altaf, 2012	ML	Semantic-heuristic Features POS	SVM	MSRP and X1999
Cordeiro et.al, 2007	ML	10 metrics include Bleu, Edit, N-gram overlap and Sum	Threshold	MSRP and KMC
Vrublevskyi, and Marchenko, 2020	ML	6 Features include Bleu, dependency tree and IDF	SVM	MSRP
Kenter and De Rijke, 2015	Hyprid ML and DL	Word2vec Glove Saliency weighted	Non liner SVM	MSRP
Ferreira et al, 2018	ML	BOW dependency tree	BayesNet	MSRP
Ji and Eisenstein, 2013	ML	TF-KLD, TF-IDF and Wan 2007 Features	Threshold	MSRP
Vrbanec, and Meštrović, 2020	Hybrid ML and DL	Semantic sentence representation includes Word2vec Fast-Text and Glove	Threshold	MSRP, C&S and Webis- CPC-11
Vrbanec, and Meštrović, 2021	Hybrid ML and DL	Word2vec and Glove	Threshold	MSRP, C&S and Webis- CPC-11

Nguyen et.al 2019	Hybrid ML and DL	Name Entity Word2vec	SVM	MSRP, SamEvel and P4P
Zhang et.al, 2017	DL	CNN	Fully connected layer	MSRP
Yin et.al, 2015	DL	Bi-CNN-MI	LR	MSRP
Yin et.al, 2016	DL	ABCNN	LR	MSRP
Devlin et.al, 2018	DL	BERT	-	MSRP
Aruse and Tsujii, 2021	DL	PPBERT	Fully connected layer	MSRP
Ko and Choi, 2020	DL	Paraphrase-BERT	-	MSRP
Palivela, 2021	TM	Word Embedding (T5)	Threshold	QQP MSRP
Xu et al., 2020	DL (GCN and BERT)	contextual features: combination of position encoding (syntactic structure information)and syntactic features	Fully connected layer	QQP MSRP
Hany & Gomaa, 2022	Hybrid Approach	string similarity, semantic similarity and embedding similarity	Linear SVC	MSRP
Muneer et al., 2025	Hybrid Approach	Combining similarity metrics from traditional techniques with embeddings generated by the Sentence Transformer model	ML	MSRP subset of QQP Artificial case paraphrases corpora Article Rewrite Corpus
Kartelj et al., 2025	Hybrid Approach	bag-of-words and character-level n-grams	20 classifiers	PhD abstracts from different universities

3.3. Datasets

The task of identifying paraphrases requires a dataset. It is the model's instructional resource where its collection contains both paraphrase and non-paraphrase examples. This aids the model in learning the linguistic patterns and the various ways that the same concept can be expressed. Additionally, the dataset's function goes beyond model training to include a model evaluation that is accomplished by using a subset of the dataset called the test set or validation set. This assessment process highlights the model's strengths and weaknesses in terms of its capacity to identify paraphrases. Moreover, the dataset makes it possible to compare several models consistently and helps to improve the model. By examining the model's performance on the dataset, the researcher can spot its weaknesses and make necessary improvements. Furthermore, the researcher can determine which models perform best for the PI task through analysing the outcomes of multiple models tested on the same dataset. In other words, a dataset is essential for developing and optimising PI models. It serves as the foundation for the development and enhancement of these models. Thus, in this section a review of various PI

datasets is offered. This review is based on the type of paraphrases, the length of the text, train and test sets size, and whether the datasets were manually or artificially created. However, corpora that are not in English are excluded, as this thesis aims to provide solutions for detecting plagiarism and paraphrases in English text.

The primary datasets utilised for training and assessing PI algorithms are mostly constructed of sentence-length and sentence-level paraphrasis that consider only intra-sentence relations, such as Microsoft Research Paraphrase Corpus (MSRP) (Dolan & Brockett, 2005), Quora Question Pairs (QQP)[2] (Puvvada et al., 2017), PAN (Potthast, Barrón-Cedeño, et al., 2010) and (Potthast et al., 2011) Plagiarised Short Answers (PSA) (Clough & Stevenson, 2011), Paraphrase for Plagiarism (P4P)[3] (Barron-Cedeno & Vila, 2013), which expanded with negative samples crating (P4PIN) (Sánchez-Vega et al., 2019), and Webis Crowd Paraphrase Corpus (Webis-CPC-11) (Burrows et al., 2013). The following paragraphs present an in-depth analysis of these datasets, as well as several less commonly used datasets primarily created for PI or PD.

The MSRP corpus consists of 5801 pairs which have been manually categorised as either paraphrase, meaning the sentences convey the same or very similar meaning, or non-paraphrase, meaning the sentences convey different meanings, and split into a train set and a test set (Dolan & Brockett, 2005). The data was collected from online news sources using heuristics to identify candidate document pairs and sentences. Each sample in this dataset consists of less than 35 words which makes it not suitable for investigating the paragraph length paraphrases task (Dolan & Brockett, 2005).

In terms of QQP, question titles from a website where users post enquiries and get responses are separated into groups for duplicates and non-duplicates which also consist of short text and sentence-level paraphrasis samples. Even though it is a large dataset (over 400,000 question pairs) that makes training DL models in PI possible, it includes slang, mathematical formulas, abbreviations, typos, etc. which can all be considered noise. Interestingly, even though the extensive QQP dataset contains labels generated by humans, these labels were not specifically designed for PI tasks. They were released as part of the PAN workshop series focusing on PD

[2] https://www.kaggle.com/c/quora-question-pairs/
[3] https://clic.ub.edu/corpus/en/parafrasi-en#

(Puvvada et al., 2017). Furthermore, the average length of its text is just 6 words, which limits its usefulness in detecting plagiarism at paragraph-length.

When it comes to PAN[4], it is a plagiarism analysis, authorship identification, and obfuscation detection dataset. It is a set of documents that are frequently utilised for research evaluating authorship attribution techniques and PD algorithms. PAN released many versions such as PAN-pc-10 which considers different languages to meet the demands of researchers working in multilingual research and AN-PC-11 corpus that represents automatic plagiarism. Moreover, PAN is intently designed to include a variety of plagiarism styles, including exact copying, paraphrasing, and obfuscation, in order to evaluate the performance of various PD methods. Additionally, PAN's samples are often manually annotated to identify the portions that have been plagiarised as well as the plagiarism type that has been applied to each sample (translating from a Spanish or German source document, or by randomly relocating words and replacing them with a comparable lexical term) (Potthast, Barrón-Cedeño, et al., 2010). Although it consists of a wide range of text documents including articles, essays, news articles, and academic papers, it consists of sentence-level paraphrases highlighted the purpose of creating this dataset, which is identifying plagiarised portions of text within documents and the corresponding source (resource retrieval in PD). In more detail, let's suppose we have a set of potential source documents (D) and a set of suspicious documents (Dq) that may contain plagiarised portions. Text fragments from documents (d) within D were randomly selected and used in some of the documents in Dq in order to simulate plagiarism. In the case of PAN-PC-10, there are approximately 70,000 plagiarism segments in Dq, 40% verbatim plagiarism content while the rest employed different obfuscation techniques, including paraphrasing. Most of these plagiarism cases are created artificially by computing tools. Only 6% of it was written by humans via Amazon Mechanical Turk (Potthast, Barrón-Cedeño, et al., 2010). While in PAN-PC-11 the number of cases that are manually or automatically obfuscated is increased to represent 71% of the dataset with 8% paraphrased manually (Potthast et al., 2011).

As the PAN dataset was mainly created to evaluate PD algorithms, (Barron-Cedeno & Vila, 2013; Sánchez-Vega et al., 2019) extracted only positive and negative paraphrased examples of PAN-PC-10 for the PI task, respectively. (Barron-Cedeno & Vila, 2013) selected pairs of source and plagiarised fragments that are less than 50 words in length end up with only 847

[4] https://pan.webis.de/index.html

paraphrase pairs. Additionally, they annotated linguistic units including words, phrases, clauses, and sentences with 20 paraphrase tags which represent the paraphrase distribution into P4P corpora. They were aiming to produce a resource that covers all possible forms of paraphrasing that could occur. To make this dataset useful in PI, (Sánchez-Vega et al., 2019) including the negative paraphrases samples of PAN to P4P called it P4PIN.

From another perspective, (Clough & Stevenson, 2011) created a PSA dataset which is specially produced for plagiarism detection in an academic context (Computer Science). The authors asked participants to wilfully reuse another document in a way typically viewed as inappropriate. There are four different levels of plagiarism in this corpus namely: near copy, light revision, heavy revision, and non-plagiarism. The dataset includes a total of 95 publications that display varying degrees of plagiarism in length 200 – 300 words including only 19 paraphrased examples. These documents are compared to five original documents that were obtained from Wikipedia. Although it consists of long text, it represents sentence-level paraphrases. In addition, it is recognised as a small dataset which makes it unsuitable for training a deep-learning model.

In relation to Webis-CPC-11, (Burrows et al., 2013) provided 7859 text which was paraphrased by Mechanical Turk crowdsourcing. The corpus consists of 4067 accepted paraphrased pairs, meaning that one piece of text is a paraphrase of the other and 3792 non-paraphrased pairs. The samples were randomly selected from about 7000 books that were provided by the Gutenberg project. The text length in this data set is largely varied from sentence-length to a long article-length which could be considered as a drawback (see Table 3.2). In terms of paragraph length, Webis-CPC-11 has in total of 1339 positive and negative samples that were paraphrased at paragraph-level and labelled by humans. Most of these datasets have a limitation on their size which makes training neural or Transformer-based models difficult. To solve this limitation, many of datasets have been created using a variety of techniques for the PI task such as PARADE and PPDB.

PARADE was created of computer science concepts from online user-generated flashcards and employed clustering techniques to categorise term definitions into groups. They selected one as the source and labelled the other as a paraphrased text, utilizing a four-label system for manual annotation (He et al., 2020). Another noteworthy resource, PPDB, constituted an extensive automatic collection of paraphrases, totalling 220 million pairs (Ganitkevitch et al., 2013). In a different way, (Kanerva et al., 2021) collected text automatically from two separate

sources: headlines from news addressing the movies or TV episodes, and alternate Finnish subtitles for the same movies or TV programs. They then labelled this collected text manually. Moreover, (Hu et al., 2019) constructed a dataset by featuring sentence-level paraphrasis generated through machine translation. This involved translating the text into another language (Czech) and subsequently translating it back to the original language (English). Thus, the quality of the paraphrased generated text was affected by the efficiency of the translation model used. Although these sources were created with a huge number of samples making training DL models possible for the PI task, they didn't consider the paragraph length or paragraph-level paraphrases.

Despite the differences in style and content quality of the mentioned datasets, they all consist of sentence-level paraphrases or have limitations on their size which makes them not suited for PI at the paragraph-level paraphrases.

Recently, a few datasets with paragraph length texts have been created; however, the type of paraphrasing utilised remains focused on intra-sentence level paraphrasis, consequently addressing within-sentence semantics, that ignores inter-sentence semantic associations. (Asghari et al., 2021) proposed HAMTA, a Persian (aka, Farsi) monolingual plagiarism detection corpus. They implemented paraphrasing techniques to extract content from the source papers (Wikipedia documents) and then insert it as plagiarised fragments into the suspect documents. The length of documents varies between 30-300 words. They applied three paraphrase operations, namely Random Text Operations, Semantic Word Variation and POS-preserving Word Shuffling. These operations generate intra-sentence level paraphrasis. In a more recent effort, (Kurt Pehlivanoğlu et al., 2024) introduced ParaGPT, a large-scale paraphrase dataset comprising 81,000 machine-generated sentence pairs. This dataset includes 27,000 synthetic reference sentences produced by ChatGPT, with paraphrases generated using three large LLMs: ChatGPT, GPT-3, and T5. All paraphrases operate at the sentence-level, and the dataset emphasises lexical and syntactic diversity. The use of synthetic reference sentences was a deliberate choice, allowing for a controlled and reproducible setup where all reference inputs are consistently generated. However, since the source sentences are also artificially generated, the dataset is unsuitable for training plagiarism detection models. Instead, it is primarily used to analyse the paraphrasing capabilities of various LLMs and highlight their performance differences. In the same context, (Wahle et al., 2021) constructed a paraphrase dataset by paraphrasing content drawn from Wikipedia, academic theses, and arXiv articles.

Unlike (Kurt Pehlivanoğlu et al., 2024), which employed autoregressive generation models, Wahle et al. relied on autoencoder-based LLMs for paraphrase generation. Although their dataset was built from full paragraphs, the paraphrasing process focused solely on intra-sentence relations by paraphrasing each sentence independently.

On the other hand, (Lin et al., 2021) and (Qiu, 2022) explored paragraph-level paraphrasis through sentence reordering after back-translating the text. They note the potential errors introduced by relying on automated translation, especially when using synonyms that lack contextual validity (Prentice & Kinden, 2018). These studies (Lin et al., 2021; Qiu, 2022) used graph models to determine the optimal sentence order of paraphrased text, neglecting source sentence relationships and impacting semantic coherence.

To sum up, most of the datasets in PI and PD were created by applying different algorithms to paraphrase each sentence independently. 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. Thus, the research presented in the current thesis aims to overcome these limitations by using Sentence Order Prediction (SOP) on the source text, generating three distinct sentence orders per paragraph based on intra-sentence and inter-sentence semantic similarities. Paraphrased paragraphs are then created using advanced Transformer-based models rather than back-translation approach. Notably, this thesis introduces the first dataset for PI training that incorporates paragraph-level paraphrasis through Transformer-based models, see Chapter 6.

Table 3.2. Key numerical features of primary corpora in PD and PI.

Corpora	Size	Positive	negative	Words	Max	Mean	Type of paraphrase	Primary task
MSRP	5801	3900	1091	211,206	34	19.29	Sentence-level	PI
QQP	404,351	149306	63% 255045	400,000	272	6.7	Sentence-level	duplicate detection problem
SAP	100	62	38	-	300	-	sentence-level	PD
PAN-PC-10	27073 documents and 68 558	60% include exacta copy and	40% without plagiarism	-	5000	-	34%: at paragraph-length (50-150)	PD

	plagiarism cases	Obfuscation plagiarism					the rest is (300-5000)	
PAN-PC-11	26 939	82% 61 064 plagiarism cases	18%	-	1150	-	35% at paragraph-length, the rest longer	PD
P4P	847	847	0	-	50	-	sentence-level	PI
P4PIN	6708	0	6708	292,050	90	44.58	-	PI
Webis-CPC-11	7520	4067	3453	4,928,055	4993	320.34	Sentence-level and paragraph-level	PI
HAMTA	-	-	-	-	300	-	Sentence-level	PD

3.4. Summary

This it underscored the advancements achieved by researchers in PI and PD, highlighting the robust experiments done to tackle these tasks. However, a notable gap remains in the literature: the majority of the existing studies primarily concentrate on sentence-level paraphrasing. Although these studies offer significant insights, they overlook the complexity involved in identifying paraphrases at the paragraph-level. This offers an opportunity to explore the effective application of ML and DL algorithms in paragraph-level paraphrasing tasks.

The limitation in the existing research can be linked to the absence of suitable datasets that facilitate such investigations. The datasets utilized in published research often lack the required structure for evaluating paragraph-level paraphrases, as previously mentioned in this chapter. This obstacle has probably restricted the advancement of more sophisticated methods for PI and PD at the paragraph-level, hence underscoring the necessity for expanded research efforts and improved dataset availability in this field. This thesis aims to bridge the gap between sentence-level and paragraph-level paraphrase identification through innovative methodologies and a new comprehensive dataset.

The following chapter outlines the methodology employed in this thesis, explaining how it contributes to and enhances the fields of PI and PD.

CHAPTER 4: METHODOLOGY

4.1. Introduction

This thesis focuses on emphasizing the importance of detecting the need for a dataset comprised of paragraph length content and paragraph-level paraphrases. The development and construction of such a dataset represent one of the key contributions of this research. Moreover, the thesis analyses the ability of LLMs such as auto-encoder and GPTs to distinguish between paraphrased paragraphs, rather than solely focusing on the identification of paraphrased words or sentences. This shift in focus from sentence-level to paragraph-level analysis is essential for understanding the broader context in which large models operate, significantly enhancing plagiarism detection's effectiveness.

In this chapter, an explanation of the research methodologies employed to address the research questions and achieve the outlined objectives is provided. The chapter begins with a thorough description of three datasets that have been collected from different domains and are utilised in the experiments conducted throughout this thesis (sections 4.2.1, 4.2.2, and 4.2.3). This is followed by a brief discussion of the methodology adopted to create the ALECS-SS dataset (section 4.2.4), including the specific processes and criteria that were followed to ensure its relevance and effectiveness. Furthermore, the implemented classification methods and the evaluation metrics are reported in section 4.3 and section 4.4, respectively. Additionally, the ethical considerations that were carefully taken into account during this research are explained in section 4.5. Finally, the chapter concludes with a detailed outline of the overall process and the experimental framework of the research, as described in section 4.6. This final section serves to integrate the various parts of the thesis and provide a clear roadmap for understanding the experimental approaches and their outcomes.

4.2. Datasets

4.2.1. MSRP

The MSRP is a significant dataset in the field of NLP, specifically designed for the purpose of identifying paraphrases. This corpus, created by Microsoft Research, has become an essential resource for assessing different models and algorithms that aim to identify semantic similarities between sentences. The numerical details of this dataset are provided in Chapter 3.

An important characteristic of the MSRP is its focus on sentence length, where pairs usually consist of sentences that have similar lengths. This component is critical for ensuring that the dataset provides an unbiased foundation for model evaluation since it eliminates biases induced by notable variations in sentence length. The balanced length distribution allows for a more accurate assessment of a model's ability to identify paraphrases based on semantic similarities rather than outer-layer text length differences. Additionally, human annotators labeled sentence pairings to ensure the dataset's annotations were highly trustworthy. This robust annotation strategy elevates the dataset's status as a valuable resource for training and evaluating paraphrase identifying algorithms. The MSRP has made a significant contribution to NLP research by defining a consistent baseline for PI tasks. It is commonly used to evaluate a wide range of models, from basic ML approaches such as Support Vector Machines (SVMs) to more complex neural network architectures. Improvements in model performance on the MSRP are frequently reflective of advances in the field of paraphrase detection. Thus, this dataset is used to evaluate the performance of the implemented algorithms for PI in this thesis (Chapters 4 and 7).

4.2.2. QQP

The QQP is one of the largest publicly available resources for paraphrase identification and has been widely adopted as a benchmark in NLP research. It is distinguished for its collection of questions of varying lengths and formats, reflecting the diverse ways in which users pose questions on the platform. This dataset, obtained from the Quora platform, provides a reliable baseline to evaluate algorithms that attempt to determine whether pairs of questions are paraphrases of one another. While QQP originates from Q&A forums, where users may duplicate questions or provide semantically overlapping answers, the underlying task of

determining whether two texts convey the same meaning through different wording parallels the challenge of paraphrase-based plagiarism. The numerical details of this dataset are provided in Chapter 3.

The QQP dataset is distinguished for its collection of questions of varying lengths and formats. This variety reflects the various ways users pose questions on the platform, presenting a unique challenge for models trained to recognize paraphrases. Because of variations in question length and structure, models must be able to detect semantic equivalence across a wide range of question formats, complicating paraphrase recognition.

The annotation process for the QQP dataset is very similar to the MSRP, it was performed by human evaluators. This dataset is extensively utilized to evaluate diverse models, and enhancements in model performance on the QQP often indicate advancements in accurately recognizing semantic similarities between questions. This dataset is employed in this thesis to evaluate the results in Chapter 8.

4.2.3. Webis-CPC-11

This dataset, created by the Webis research group, is a crucial resource for evaluating algorithms that detect semantic similarities or paraphrases between documents. Additionally, expert annotations further boost the dataset's reliability, ensuring high standards of accuracy and consistency in its labeling. Detailed statistics concerning the dataset are provided in Chapter 3.

A key characteristic of the Webis-CPC-11 dataset is its diverse range of document lengths and sources, providing a richer and more varied evaluation context for algorithms. Unlike many datasets that focus on short, uniform sentence structures, Webis-CPC-11 includes short, mid and long texts, challenging models to manage complex variations in length and structure. This variety prevents models from developing biases toward specific text types, allowing for a more comprehensive evaluation of their paraphrase detection capabilities. Thus, the Webis-CPC-11 dataset is utilised to measure the effectiveness of the paraphrase detection algorithms at different levels of text length in this thesis (Chapter 5).

4.2.4. ALECS-SS

The ALECS-SS dataset represents one of the primary contributions of this thesis. After thoroughly evaluating ML and DL models using the previously discussed datasets, MSRP and Webis-CPC-11, the results demonstrated that these models perform better when dealing with texts in the paragraph length range (refer to Chapter 5). Nevertheless, a notable limitation was the size of these datasets, which restricted the models' performance evaluation, especially for longer texts. To address this gap, the ALECS-SS dataset is created, a significantly larger collection of texts that specifically focuses on paragraph length and paragraph-level paraphrases content (Chapter 6).

The development of this comprehensive dataset was enabled by advancements in LLMs, which can comprehend and paraphrase text with capturing the broader contextual meaning. These models made it possible to create human-like paraphrases that capture both meaning and structure effectively. The state-of-the-art LLMs are implemented to ensure that the generated paraphrases are both high-quality and diverse, a process that is detailed further in Chapter 6. The methodology involved assessing the semantic similarity between sentences within a paragraph and then reconstructing them based on the source's sentence order probability while accounting for inter-sentence relationships, resulting in alterations to the syntactic structure of the paragraphs. Then, a masking technique was applied to paraphrase individual words, thereby modifying the paragraph's lexical composition. Despite these changes, the process crucially preserves the underlying semantic meaning of the paragraphs, ensuring that its core message remains intact.

In addition to the dataset's creation, an evaluation process was conducted to rigorously test the quality of the generated paraphrases (see Chapter 6 and Chapter 7). This evaluation ensured that the paraphrased content closely mirrored human-written text semantically. ALECS-SS is later used in Chapter 8 to assess the ability of various models, including LLMs and regression-based models like GPT, in distinguishing between machine-generated paraphrased texts and human-written paragraphs. This dual evaluation not only showcases the quality of the dataset but also highlights the strengths and limitations of current paraphrase detection models when applied to more complex, longer texts.

4.3. Text Classification

PI is commonly recognised as a binary classification task within the field of NLP, where an algorithm analyses text inputs and returns a binary output of either 0 or 1. In this task, the model is presented with two text samples, designated as Text A and Text B, and assesses whether they convey equivalent semantics. If the model identifies Text B as a paraphrase of Text A, it assigns a value of 1; otherwise, it returns a value of 0. This fundamental approach is clarified in Chapter 1, which introduces the concept and its significance in NLP. This conventional binary classification approach is appropriate for models created with the MSRP, QQP, and Webis-CPC-11 datasets, each comprising three essential elements: Text A, Text B, and a label denoting whether the texts are paraphrases. This approach represents the standard scenario in PI tasks, as outlined in the literature review (Chapter 3). It has been widely adopted for evaluating paraphrase detection models across various datasets.

However, with the recent advancements in text generation technologies, a new scenario has been introduced in PI research. In these new challenges, the algorithm is tasked not only with identifying paraphrases but also with distinguishing between the source text and the paraphrased text. This adds a layer of complexity since the model must differentiate between source and paraphrased content without offering pair information. Both the traditional and the newer scenarios are explored in this thesis: the first is examined in Chapter 5, while the latter is investigated in Chapter 8. By addressing both approaches, this thesis offers a comprehensive analysis of PI in the context of modern NLP challenges.

4.4. Evaluation Metrics

In PI, evaluating model performance is crucial for illustrating their effectiveness in distinguishing between paraphrased and non-paraphrased text. A variety of evaluation metrics are employed to assess the ACC and robustness of PI models. These metrics offer a quantitative evaluation of the model's effectiveness in identifying paraphrases, considering many factors including precision, recall, F1-score, and overall accuracy. A thorough analysis of the key evaluation metrics in the context related to Paraphrase detection is presented below, highlighting the categorization of samples into two groups: positive and negative. The definitions of these categories are as follows:

- True Positive (TP): Cases where the model correctly identifies a paraphrase.

- True Negative (TN): Cases where the model correctly identifies a non-paraphrase.
- False Positive (FP): Cases where the model incorrectly labels a non-paraphrase as a paraphrase.
- False Negative (FN): Cases where the model fails to identify a paraphrase and labels it as a non-paraphrase.

4.4.1. Accuracy (ACC)

ACC is a commonly used evaluation metric for PI models that represents the ratio of correctly predicted samples to the total number of samples. It considers both paraphrased and non-paraphrased samples within the dataset and is often its main metric examined.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \quad (4.1)$$

ACC is especially advantageous when the dataset has an equal split in paraphrases and non-paraphrases, as it provides a quick assessment of the model's overall performance. In cases with imbalanced datasets, when non-paraphrase pairs overwhelm paraphrases, ACC might be misleading. A model that consistently predicts “non-paraphrase” for any sample may achieve a high ACC, despite its failure to correctly identify paraphrases, which is a critical aspect of the task.

4.4.2. Precision (P)

Precision measures the percentage of true positive samples to the total samples the model classifies as positive. Essentially, precision reflects the model's confidence in its positive predictions, making it a crucial metric in situations where false positives (incorrectly identifying non-paraphrased text pairs as paraphrases) incur a high cost.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad (4.2)$$

Precision is required in scenarios like PD when misclassifying unrelated text as a paraphrase can yield substantial consequences. Nonetheless, high precision alone may be problematic if the model shows overboard conservatism in identifying paraphrases, leading to the missing of many true paraphrase samples, which is where recall comes into play.

4.4.3. Recall (R)

Recall, or sensitivity, measures the percent of true paraphrases correctly detected by the model relative to the total actual paraphrases in the dataset. It illustrates the model's ability to recognise paraphrase pairs, making it crucial in situations that failing to detect a paraphrase (false negatives) could result in substantial costs. However, a model with a high recall may also produce a high number of false positives due to its overly eagerness in predicting paraphrases. Therefore, recall makes sure that fewer paraphrases are missed; however, it must be balanced with precision to prevent the overall prediction accuracy from being affected.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad (4.3)$$

4.4.4. F1-score

F1-score combines precision and recall into a single metric in a way that balances the trade-offs between them. It is especially advantageous in datasets that are imbalanced, as it guarantees that neither precision nor recall are disproportionately prioritised, as both false positives and false negatives must be considered.

$$\boldsymbol{F1} = \boldsymbol{2} * \frac{\mathbf{Precision} * \mathbf{Recall}}{\mathbf{Precision} + \mathbf{Recall}} \quad (4.4)$$

The F1-score is beneficial for understanding the model's performance when both identifying and avoiding paraphrase pairs are equally important.

In this thesis, the primary focus is placed on the F1-score due to the imbalanced nature of the datasets used for classification experiments. In these situations, the F1-score provides a balanced measure that considers both precision and recall. Unlike ACC, which can be misleading in imbalanced datasets, the F1-score ensures adequate accounting for both false positives and false negatives.

While this research primarily uses the F1-score to evaluate the models' performance, other evaluation metrics like ACC are also reported. This provides for an in-depth evaluation of the algorithms, particularly in comparison to previous research. Many existing works in the field tend to prioritise ACC, often neglecting the importance of the F1-score in scenarios where the dataset is imbalanced. By including both metrics, this thesis aims to provide a more nuanced

comparison with prior works, highlighting the significance of using the F1-score alongside ACC to better assess model performance.

4.5. Ethical Considerations

In this research, ethical considerations were rigorously followed throughout the process of establishing the ALECS-SS dataset. The dataset consisted of Wikipedia text and was labelled by university students. Informed consent was obtained from all participants, with a clear explanation provided regarding the project's purpose, their role in the labelling tasks, and the intended use of their contributions. Participation was entirely voluntary, and assurances were given that opting out would not affect academic standing. To protect privacy, all contributions were anonymised, with no personally identifiable information linked to their work or disclosed in the final dataset. According to the Durham University ethics committee, formal ethical approval was not required for this study.

To ensure fairness and protect the interests of the students involved in the labelling process, the tasks were carefully balanced with their existing academic commitments. The labelling tasks were structured to be intellectually stimulating and beneficial to their research skills, without imposing an excessive burden on their time. For PhD students participating outside formal academic requirements, proper recognition of their contributions was provided, and they were credited accordingly.

To address potential bias, the students were provided with thorough training to ensure consistency and objectivity in the labelling process. The dataset was carefully monitored for any signs of bias, and efforts were made to minimise it throughout the project. Recognizing the potential for misuse, clear guidelines for the dataset's use were established, and its limitations were acknowledged in the thesis (Chapter 6).

4.6. Conceptual Framework

The research goal is to detect and identify paraphrased text at the paragraph-level, addressing a gap identified through the literature review in Chapter 3. Selecting an appropriate research methodology is important and must align with the particular issue being investigated. Based on the thesis objectives and research questions, the research framework is structured to begin with experiments that incorporate handcrafted features from texts of varying lengths into ML

and DL models, utilizing widely recognised datasets for PI tasks, such as MSRP and Webis-CPC-11, Figure 4.1. These experiments follow a binary classification approach, where two texts are compared to generate a label (Chapter 5). The findings from these experiments underscored the need for a paragraph-level dataset which used for examining the detection algorithms in Chapter 8, Figure 4.2.

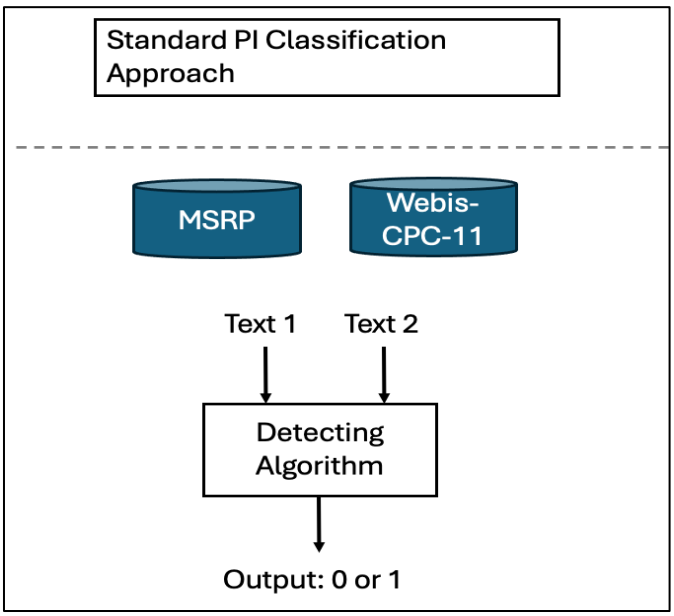

Figure 4.1 Flowchart of a study that is explained in Chapter 5

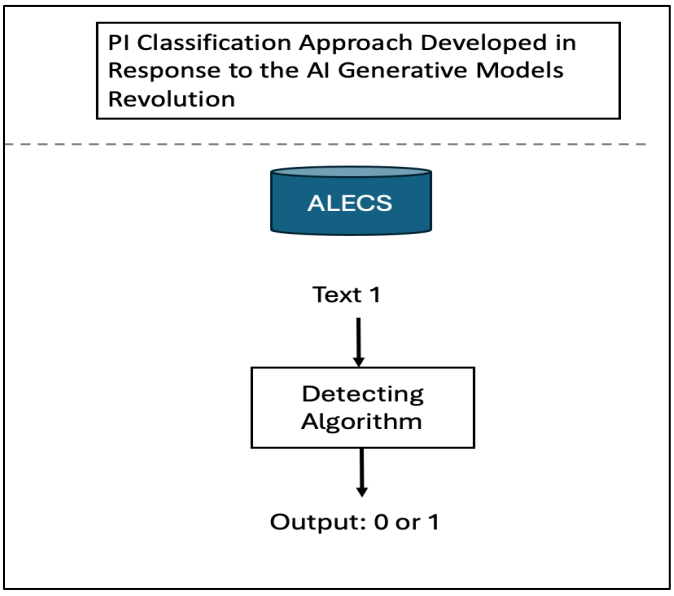

Figure 4.2 Flowchart of a study that is explained in Chapter 8

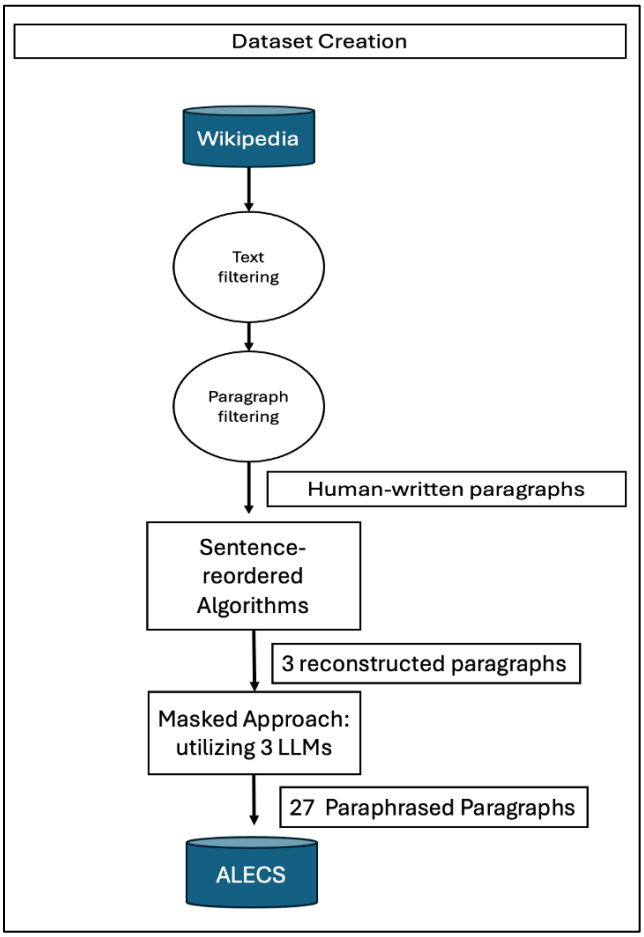

Figure 4.3 ALECS-SS dataset creation process Chapter 6

Subsequently, it is created through a complex process using Wikipedia content Figure 4.3, called ALECS-SS (Chapter 6). This new dataset is then used to evaluate state-of-the-art models' ability to differentiate between machine-paraphrased and human-written paragraphs. These experiments are also applied as a binary classification task, taking a single paragraph as input and returning a label (Chapter 8) Figure 4.2. The MSRP and QQP datasets are also employed for this purpose. The overall experiments architecture of the thesis is shown in Figures 4.1, 4.2 and 4.3.

4.7. Summary

The existing research has primarily focused on short texts and sentence-level paraphrasing. However, recent advancements in LLMs have made it possible to analyze the context and semantics of longer texts. In this thesis, the focus is on studying paragraph-level paraphrasis, taking into account both intra-sentence and inter-sentence relations. To address the research

gaps identified in Chapter 3, a series of experiments were conducted, and their methodologies are summarized in this Chapter.

The initial phase of the research involved the implementation of various ML and DL classification algorithms. These experiments highlighted the limitations of existing datasets, particularly in training transformer models, underscoring the need for a new dataset. Subsequently, the datasets used in the study were introduced, followed by a detailed explanation of the development of ALECS-SS.

After the dataset is created, several classification methods are applied to differentiate source paragraphs from the machine-paraphrased text at the paragraph-level. The experiments demonstrated high ACC, particularly in detecting challenging cases of auto-paraphrased samples, where the proportion of paraphrased tokens was minimal. This performance highlights the effectiveness of the developed models in handling complex paraphrase identification at the paragraph-level. The following chapters discuss these experiments in more detail.

CHAPTER 5: A PARAPHRASE IDENTIFICATION APPROACH IN PARAGRAPH-LENGTH TEXTS

5.1. Introduction

When students submit their work, institutions must ensure that it is free from plagiarism. Given the limitations of human capabilities in terms of scalability, such as the time required for thorough checks and maintaining consistency, ML and DL techniques are employed for tasks like plagiarism and paraphrase detection. Plagiarism as mentioned in the introduction refers to the use of another person's written work without proper citation or presenting others' ideas as one's own (Maurer et al., 2006). In some cases, even when a reference is provided, extensive word-for-word copying from the source is also deemed plagiarism (Bär et al., 2012). Additionally, rewriting sentences while maintaining the source structure without proper attribution is categorised as plagiarism as well. On the other hand, paraphrasing is the process of conveying the same meaning as the source but using different text structures and vocabulary (Bhagat & Hovy, 2013). This definition sets paraphrasing apart from verbatim reproduction, which involves copying the text exactly as it appears and is clearly not considered paraphrasing. As such, paraphrasing presents a significant challenge in the realm of PD. Detecting instances of paraphrasing requires sophisticated techniques, as it involves recognising when ideas are expressed in a new form but still closely mirror the source. Consequently, identifying and addressing paraphrasing is often regarded as one of the most complex aspects of PD.

Generally, efforts to tackle the issue of PI have concentrated on comparing individual words within sentences (Vrublevskyi & Marchenko, 2020; Wan et al., 2006), analysing phrases within sentences (Arase & Tsujii, 2021), or contrasting entire sentences with one another Nguyen et al., 2019). While these methods have yielded solid results, they are not well-suited for processing longer texts. Specifically, comparing each sentence in a potentially lengthy suspicious document to every sentence in the source texts proves to be an inefficient approach, highlighting the need for more scalable solutions in PI.

Given the challenges associated with detecting paraphrasing, the objective is to create a method specifically designed to identify paraphrases at the paragraph-level. This approach shifts the focus from analysing individual sentences to treating entire paragraphs as the fundamental units of comparison. By examining paragraphs as cohesive blocks of text, this chapter aims to streamline the detection process and improve efficiency. Instead of comparing every sentence in a document to all sentences in the source material, the current study will evaluate the overall structure and meaning of paragraphs. This will not only simplify the detection process but also enhance the accuracy of identifying paraphrasing, aligning with the complexities discussed in the previous chapters.

In this chapter, ML and DL classification models are implemented. The classification models operate by processing two pieces of text simultaneously and determining whether they convey the same information. If the texts are semantically equivalent, the model outputs a '1'; if they are not, it returns a '0'. The primary objective of this experiment is to explore how the length of the text influences the model's effectiveness in accurately detecting paraphrasing, providing insights into the scalability and sensitivity of the ML and DL models' performance in terms of the PI task. To investigate this, the dataset samples are categorised based on their length, organising them into distinct groups.

Specifically, ML approaches that primarily rely on handcrafted features are implemented, alongside advanced DL models for sentence representation, such as word2vec and SBERT. These methods are chosen due to their proven effectiveness in PI tasks (see Chapter 3). It is important to acknowledge that while transformer-based models are highly effective, they present significant challenges, particularly in terms of time consumption, computational resources, and high memory usage. Given these limitations, the focus of this work is on alternative approaches that have consistently yielded strong results in the past. The goal is not only to leverage these established methods but also to enhance and refine them for improved performance. In doing so, this chapter aims to strike a balance between model efficiency and accuracy, building on prior successes while addressing the limitations of more resource-intensive models.

For the purpose of this study, the following definitions are provided:

- *Sentence-level paraphrasing:* Refers to the scenario where the meaning of a single sentence is paraphrased into exactly one other sentence, as seen in MSRP.

- *Paragraph- level paraphrasing*: Refers to the paraphrasing of a block of text consisting of multiple sentences, where the paraphrase may involve a different number and/or order of sentences. This can be observed in datasets like the Webis-CPC-11.
- *Passage-level paraphrasing:* Refers to the paraphrasing of a multi blocks of text, where the paraphrase may involve a different number and/or order of blocks. (This type is out of this thesis's scope)
- *Sentence-length level:* Defines a short text length, consisting of fewer than 50 words.
- *Paragraph-length level:* Represents a mid-length text, typically around 100 words, which is considered the average length of a paragraph (Larock MH et al., 1980).
- *Passage-length level*: Refers to a longer text that spans more than a single paragraph, containing 150 words or more.

With these definitions in mind, the primary research questions (RQs) for this chapter are formulated accordingly.

- *RQ1: How does the length of a piece of text affect the accuracy of the paraphrase identification approach used?*
- *RQ2: What features are most effective for paraphrase identification across different levels of paraphrasing and varying text lengths?*

5.2. Method

Previous research has explored various pre-processing techniques, such as the removal of stop words and word lemmatisation (Wan et al., 2006), as well as similarity measures like cosine similarity, soft cosine, and Euclidean distance (Vrbanec & Meštrović, 2021). Additionally, studies have examined the use of pre-trained word embedding models (Vrbanec & Meštrović, 2020) in the context of PI tasks. Building on this foundation, the influence of input text length on the accuracy of ML and DL models in determining the optimal number of words necessary to convey sufficient semantic information is investigated in this chapter. Specifically, this experiment aims to address whether shorter texts, such as individual sentences, medium-length texts like paragraphs, or longer texts consisting of multiple passages or extended paragraphs, provide the most meaningful semantic detail for ML and DL models to effectively identify paraphrases. Furthermore, the study explores which features are best suited for paraphrase detection across varying text lengths and paraphrasing levels. It is investigated whether certain

types of features are more effective for identifying paraphrases in shorter sentence-based inputs compared to those found in longer paragraph-based texts. The analysis is designed to uncover the relationship between text length, semantic content, and feature selection in improving model performance on PI tasks.

5.2.1. Dataset

In this experiment, the focus is placed on two widely used datasets in the field of PI: the MSRP and the Webis-CPC-11. These datasets serve as benchmarks for evaluating PI models. However, it is important to note that they differ significantly in terms of the distribution of their labelled categories.

MSRP is known for being imbalanced, meaning that the dataset contains a disproportionate number of positive and negative samples. Specifically, 67% of the samples in MSRP belong to the positive class, indicating that the majority of the text pairs in this dataset are labelled as paraphrases. This imbalance can affect model training, as the classifier may become biased toward predicting the majority class.

On the other hand, Webis-CPC-11 offers a nearly balanced dataset, with 51.75% of the samples being acceptable paraphrased pairs and 48.25% classified as non-paraphrased pairs. This balanced distribution allows for a more even representation of both classes, which can be beneficial for training models that need to generalise well across both paraphrased and non-paraphrased instances.

Most PI experiments to date have primarily focused on the MSRP, which is composed of sentence-level paraphrases. As a result, the majority of research in this area primarily evaluates sentence similarity, which limits the scope of findings to relatively short text segments. However, in real-world applications, paraphrasing often occurs at a higher, more natural level, such as paragraphs (Wahle, Ruas, Foltýnek, et al., 2022), as is more commonly seen in the Webis-CPC-11. This makes Webis-CPC-11 a valuable resource for studying paraphrase detection beyond the sentence-level, offering a more realistic reflection of paraphrasing in real-world scenarios.

A key point of comparison between the MSRP and Webis-CPC-11 datasets is the difference in text length, which plays a crucial role in this study. While MSRP limits its scope to sentences

with a maximum length of 36 words (Figure 4.1. a), the Webis-CPC-11 dataset contains passages that can reach up to approximately 1,000 words in length (Figure 4.1.b). The substantial variation in text length within the Webis-CPC-11 corpus provides an opportunity to investigate how text length influences the performance of ML and DL models in PI tasks. This range allows for a deeper analysis of how different lengths of text impact model accuracy and effectiveness.

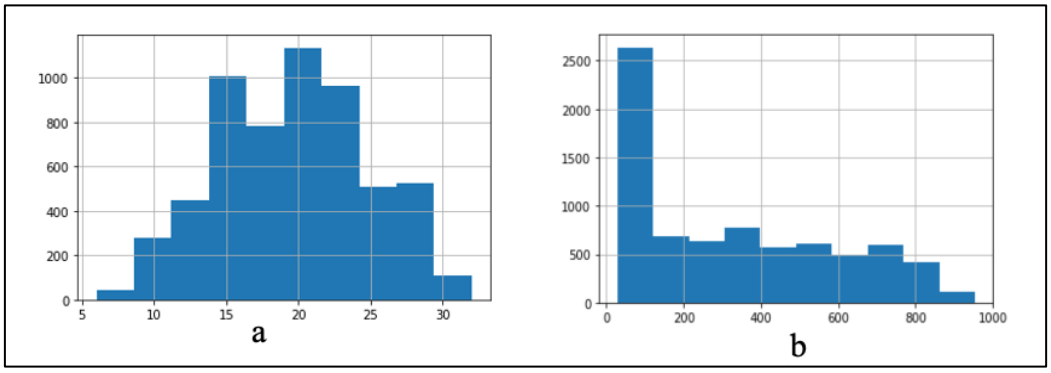

Figure 5.1 The X-axis represents the number of words in a given text, whilst the Y-axis represents the number of samples in MSRP (a) and Webis-CPC-11 (b)

To fully explore the impact of text length on model performance, the Webis-CPC-11 dataset has been further divided into three sub-corpora based on text length: short, medium, and long (as discussed in section 5.3.4 below) after implementing pre-processing techniques. This stratification allows for a systematic investigation of the effect of different text lengths on both traditional ML models and DL models, with the aim of identifying the optimal length of text for effective PI.

5.2.2. Method of Feature Extraction

For this study, the focus is on selecting the most relevant features for PI that can effectively transform text into numeric representations. The features chosen for this task are those that have been widely used in previous research, as discussed in Chapter 3, including TF-IDF, BLEU, dependency trees, N-gram overlap, and Word2Vec (Cordeiro et al., 2007; João et al., 2007; Hunt et al., 2019; Ji & Eisenstein, 2013; Kenter & de Rijke, 2015; Nguyen et al., 2019; Vrbanec & Meštrović, 2020; Vrbanec & Meštrović, 2021; Wan et al., 2006). Each of these features has proven effective in capturing different aspects of text similarity and structure.

However, it is important to note that most of the prior studies have mainly utilised the MSRP, which is focused on sentence-level paraphrasing. In contrast, this research shifts the focus to a more complex task: paragraph-level paraphrasing, as seen in the Webis-CPC-11 dataset. By moving beyond sentence-level comparisons, this study aims to explore how features perform when applied to longer text segments, where the relationships between sentences and the overall structure of the text play a more significant role. Thus, the use of paragraph-level paraphrasing presents new challenges and opportunities for improving the accuracy of PI tasks, especially when compared to the sentence-level paraphrases found in the MSRP dataset.

5.2.2.1. Bleu and N-gram overlap features.

The BLEU metric, originally developed to automatically evaluate the quality of machine translation systems (Papineni et al., 2002), operates by comparing a translated text against a reference source text. It does this by calculating the overlap of N-grams, which are continuous sequences of words, between the two texts. Then, João et al. adapted this metric for use in PI tasks (João et al., 2007). In this adaptation, the BLEU score is computed by counting the N-gram overlaps between two sentences, where N is set to 4. This is expressed mathematically in Equation 5.1, where C_n represents the ratio of matching N-grams to the total number of N-grams in a given sentence.

$$\boldsymbol{C_n} = \sum_{ngram} \frac{\boldsymbol{Count_{match}\,(ngram)}}{\boldsymbol{Count(ngram)}} \tag{5.1}$$

Equation 5 .1 captures the N-gram overlap feature, which is integral to implementing BLEU for PI tasks. BLEU for PI is further modified by introducing the brevity penalty (BP) factor, which adjusts for variations in sentence length, as shown in Equation 5.2.

$$BLEU_{adapted} = BP * [\prod_{n=1}^{N} C_n]^{\frac{1}{N}} \tag{5.2}$$

This adaptation of BLEU has been successfully applied in several studies to measure paraphrase similarity and has demonstrated effectiveness comparable to more complex methods, such as those based on dependency tree structures (Vrublevskyi & Marchenko, 2020; Wan et al., 2006). However, in this study, the focus remains solely on BLEU for a few reasons. One key reason is that while dependency tree features have been shown to perform well on shorter text segments, they are less effective when applied to longer texts (Kenter & de Rijke,

2015). Given that this research explores PI across a range of text lengths, including short, medium, and long passages, BLEU's simplicity and adaptability make it a more suitable choice. Moreover, Equation 5.1 is implemented to extract N-gram overlap features, allowing for a more efficient and scalable approach to PI over varying text lengths.

5.2.2.2. Word2vec

Word2Vec is a powerful word embedding technique that transforms words from a vast corpus into fixed-length, dense, continuous-valued vectors. The primary advantage of these vectors is that cosine similarity can be used to measure the degree of semantic similarity between words. Essentially, words with similar meanings will have vectors that are closer in their multidimensional space, while words with different meanings will be further apart. This vector-based representation is generated through a neural network model that considers the surrounding context of each word, enabling it to capture not just the word's meaning, but also its relationship with other words in the corpus.

There are two primary methods for training Word2Vec models: CBOW and Skip-Gram (Hunt et al., 2019). CBOW predicts a target word based on the surrounding context, while Skip-Gram does the reverse by predicting the context words from a target word. In this study, a pre-trained Word2Vec model with 300 dimensions is used, where each word is represented as a 300-dimensional vector. Although these models are inherently designed to represent individual words, they are often extended to represent entire sentences or documents by aggregating the vectors of all the words within the text. A common approach to achieve this is by taking the mean of the word vectors across the text. However, as noted in (Kenter & de Rijke, 2015), averaging word vectors does not always provide an accurate representation of a sentence or passage, particularly in cases where one text contains terms absent from the other.

To address this issue, instead of calculating the mean of word vectors, the vectors of all words within the text are summed to generate a more robust representation of the entire text. This method, known as sent2vec (Pagliardini et al., 2018), is utilised to enhance the overall text representation and is expressed mathematically in Equation 5.4, where ***d*** represents the vector dimensions and ***n*** represents the number of words in the paragraph:

$$sen2vec = \sum_{i}^{d} \sum_{j}^{n} w_{ij} \tag{5.4}$$

The next step involves determining the similarity between the vector representation of the source text and the paraphrased text. While cosine similarity was used in previous studies by (Vrbanec & Meštrović, 2020; Hunt et al., 2019) to measure the degree of overlap between two vectors, a different approach is adopted in this study. Instead of using cosine similarity, the vector of the paraphrased text is subtracted from the vector of the source text. This subtraction provides a measure of the distance between the two vectors, reflecting the degree to which the meaning of the paraphrased text overlaps with or diverges from that of the source. By focusing on the differences between the vectors, this method offers a precise understanding of the semantic similarity between the texts, capturing subtle shifts in meaning that may be missed by other approaches.

5.2.3. Classifier

In this chapter, several ML algorithms commonly applied to PI, treated as a binary classification task, have been tested. Algorithms such as LR and SVM are evaluated for their performance in determining whether a pair of texts represents a paraphrase. While multiple algorithms are explored, only the best-performing models are reported in detail in the following sections.

Furthermore, to incorporate cutting-edge approaches known for achieving high accuracy across various NLP tasks, the performance of the SBERT model is also examined in this study. SBERT is a state-of-the-art DL model specifically designed to produce high-quality sentence embeddings, and it has demonstrated superior performance in multiple downstream tasks, including paraphrase detection. Here, SBERT is applied not just to sentence-level paraphrasing but also to more complex paragraph-level paraphrasing, with an emphasis on how text length influences its effectiveness. The study investigates SBERT's capabilities across different text lengths, ranging from short sentences to longer passages, to assess its adaptability and accuracy in capturing semantic relationships across varying text sizes.

5.3. Experiment

To investigate how text length influences the accuracy of using individual or combined features in ML and DL models for PI, this study utilises two datasets: the MSRP and the Webis-CPC-11. MSRP is selected for its focus on short texts, while Webis-CPC-11 offers a diverse range of text lengths, making it ideal for a more comprehensive analysis.

The samples from Webis-CPC-11 are first pre-processed to remove empty entries and pairs where the source and paraphrased texts significantly differ in length. Additionally, identical text pairs are excluded to align with this work's definition of paraphrasing, as outlined in Chapter 1. According to this definition (referenced in formula 1.1 in Chapter 1), two texts in a pair must differ in wording but retain the same meaning. After these adjustments, the dataset is renamed Webis-CPC-21, following the naming convention of the original dataset, where "11" refers to the year of creation, 2011. This new dataset reflects the adjustments made to ensure it adheres to the study's criteria for paraphrasing.

Tables 5.1 and 5.2 illustrate the Webis-CPC-11 and Webis-CPC-21 datasets. The total number of Webis-CPC-11 samples is reduced by 3,989 after removing identical and highly similar samples. Similarity is measured using TF-IDF, followed by cosine similarity to assess the similarity between text pairs. All samples with 90% or greater similarity are removed. Notably, these samples included both positive and negative labels, as outlined in the original paper on Webis-CPC-11, which indicated that samples are labelled as negative not only when relevance is lacking but also when excessive similarity is present. As a result, both highly similar positive and negative samples are removed to create Webis-CPC-21. Moreover, these tables provide numerical information for each dataset, detailing the total number of short, medium, long, and extra-long samples, as well as the positive and negative labels within each category.

Table 5.1*:* Number of positive samples (i.e., true paraphrase) and negative samples (i.e., non-paraphrase) in Webis-CPC-11.

Category	Positive	Negative	Total
Webis-CPC-11	4067	3453	7520
Short text	978	339	1317
Mid text	1207	705	1912
Long text	331	287	618
Extra long	1506	2091	3597

Next, Webis-CPC-21 is divided into three subsets based on text length: short, medium, and long texts. The short text subset includes samples with up to 50 words, while the mid-length text subset covers texts between 51 and 150 words, consistent with the typical length of an English paragraph, which averages around 100 words (Larock MH et al., 1980). The long text

subset includes samples with 151 to 500 words. This stratification allows for a more detailed analysis of how different text lengths impact model performance. Notably, the category of extra-long text is excluded from this study because its lengths ranged from 500 to over 1000 words, exceeding the maximum length considered by state-of-the-art models in this experiment, which is 500 tokens. Additionally, analysing texts with variations in length exceeding the 500-word range is deemed unnecessary for the objectives of this experiment.

MSRP, on the other hand, is retained in its original form due to its focus on short texts, with most samples containing fewer than 40 words. Although MSRP only includes short text data, it is widely used in state-of-the-art paraphrase identification research, making it a valuable point of comparison for this study. Even though the comparison between MSRP and Webis-CPC-21 is not entirely equivalent due to differences in text length, including both datasets allows for insights into how models perform across varying text lengths.

Both the MSRP and Webis-CPC-11 datasets consist of paraphrased and non-paraphrased text pairs, labelled as positive and negative, respectively. Table 5.2 provides a breakdown of the number of positive and negative samples in each dataset.

Table 5.2: Number of positive samples (i.e., true paraphrase) and negative samples (i.e., non-paraphrase) in each category.

Category	Positive	Negative	Total
Webis-CPC-11	4067	3453	7520
Webis-CPC-21	2690	841	3531
Short text	931	227	1158
Mid text	1085	254	1339
Long text	446	154	600
Extra long	228	206	434
MSRP	3900	1901	5801

5.3.1. Pre-processing

Pre-processing in this study involves both cleaning the data and converting it into a vector format before it is passed into a classifier. The data cleaning phase includes several steps aimed at enhancing the quality of the input text. First, irrelevant punctuation and common stop words

are removed. Stop words are frequently occurring words like “a,” “in,” and “the”. Since there is no universal stop-word list that applies to all NLP tasks, this study employs the stop-word list provided by NLTK in Python.

In addition to removing stop words, all text is converted to lowercase to ensure uniformity, as capitalisation is not essential for most NLP tasks. Following this, the words are lemmatised, a process that reduces each word to its root or base form, depending on its context and part of speech. Lemmatisation helps retain the semantic meaning of words while simplifying them to their most fundamental form. For this task, the WordNet Lemmatiser is used, which leverages WordNet’s built-in “morphy” function to identify the root form of each word. If the word is not found in WordNet, it remains unchanged. This method ensures that the text is in its simplest and most meaningful form, allowing the classifier to focus on relevant patterns and relationships between words.

5.3.2. Feature Sets

Since the focus is on evaluating how different features perform across various text lengths, experiments are conducted for each feature—TF-IDF, Bleu metric, sent2vec, and N-gram overlap—both individually and in combination. These experiments are applied to the original dataset, the modified dataset, and the sub-datasets with different text sample lengths.

5.3.3. Baseline Model’s Result

The findings reported in (Burrows et al., 2013) established the ground truth for the Webis-CPC-11 dataset. Specifically, the dataset's precision is 81, ACC is 84, and recall is 90. Although (Burrows et al., 2013) did not directly report the F1-score, it has been computed using the equation reported in section 4.4 of this thesis, yielding an F1-score of 85. These performance metrics are derived from the results of (Burrows et al., 2013), where ten distinct metrics are used as features for a k-nearest neighbour ML algorithm.

5.3.4. Results and Discussion

The evaluation of different features across various datasets reveals distinct performance patterns in PI tasks, particularly concerning text length (Table 5.3). In the Webis-CPC-11 dataset, the TF-IDF feature demonstrates superior performance, achieving the highest ACC of

87% and an F1-score of 82%. This suggests that statistical methods such as TF-IDF are highly effective in capturing important word-frequency relationships. Conversely, the Sen2vec model shows moderate success, with a 64% ACC and a 63% F1-score, indicating that while semantic representation has value, it may not completely capture the subtleties within this dataset that contains highly similar samples in both categories of label (positive, negative). Interestingly, the combination of all features results in a marginal improvement, with an ACC of 66% and an F1-score of 65%, which implies that for a variety length of text in Webis-CPC-11, combining features does not significantly enhance the model's overall performance.

After modifying the Webis-CPC-11 dataset by excluding identical samples and pairs with notable differences in length to create Webis-CPC-21, a significant improvement in results is observed. Specifically, exceptional performance is achieved by semantic-based features like Sen2vec, with an ACC of 80% and an F1-score of 88%. Strong performance is also demonstrated by N-gram overlap, which yielded an ACC of 83% and a 90% F1-score, indicating its effectiveness in capturing structural relationships between paraphrases, particularly within this dataset. However, when all features are combined, the ACC decreased slightly to 79% while the F1-score remained high at 88%. This suggests that although feature integration provides comprehensive coverage, it may weaken the advantages of individual features, particularly for long text category. Overall, the dataset modifications and the less variety in text-lengths subsets samples size contributed to improved results compared to the original Webis-CPC-11.

Table 5.3: ACC and F1 results for BLEU, TF-IDF, Sent2Vec, N-gram overlap, and All Features across six datasets. Bold = highest per feature; underline = highest per dataset.

Dataset	**Bleu**		**TF-IDF**		**Sen2vec**		**Ngram_overlap**		**All Features**	
Evaluation	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**
Webis-CPC-11	57	72	**87**	82	64	63	56	68	66	65
Webis-CPC-21	77	87	78	87	80	88	83	90	79	88
Short text	77	82	78	87	79	88	77	82	79	88
Mid length	**81**	**89**	83	**90**	**84**	**91**	**85**	**91**	**85**	**91**
Long text	73	84	73	84	75	89	78	86	75	85
MSRP	67	80	71	80	72	82	69	81	71	81

In the case of short texts, such as those found in the Webis-CPC-21 dataset, both TF-IDF and Sen2vec exhibit strong performance, with accuracies ranging from 78% to 79% and F1-

scores from 87% to 88%. These results indicate that statistical features like TF-IDF, as well as semantic representations, are particularly well-suited for short-text PI. Notably, combining all features in this case maintains similarly high results, with an ACC of 79% and an F1-score of 88%, indicating that a combination of features can capture diverse characteristics of short texts without significant performance trade-offs.

When analysing mid-length texts, Sen2vec and N-gram overlap emerge as the top-performing features. Both achieve high ACC (84-85%) and F1-scores (91%), indicating that semantic and structural features are particularly effective for texts of moderate length. In this case, combining all features yields results similar to those obtained from using the N-gram overlap feature alone, with an ACC of 85% and an F1-score of 91%. Generally, all features show high ACC and F1-scores, ranging from 81% to 85% and 88% to 91%, respectively. Thus, mid-length text is more effective for identifying paraphrases.

For long texts, the results indicate a slight to high decrease in performance across all features compared specifically to the mid-length texts. Sen2vec continues to perform best, with an ACC of 75% and an F1-score of 89%, emphasising that semantic representations are crucial for capturing the meaning of longer texts. However, combining all features results in a drop in both ACC and F1-score, suggesting that the combination may not provide a substantial advantage for longer text samples, where semantic representation is more critical.

In the MSRP dataset, which primarily consists of short sentences, both Sen2vec and TF-IDF perform almost similarly, with accuracies around 71-72% and F1-scores of 80-82%. This demonstrates that for sentence-level paraphrasing, both statistical and semantic features are highly effective. The combination of features, in this case, does not show significant improvement, achieving an ACC of 71% and an F1-score of 81%. This suggests that for short-text datasets like MSRP, individual features can perform as well as the combined methods, possibly due to the limited variation in sentence structure and content.

Furthermore, the results for each feature on the MSRP dataset are notably worse compared to those achieved on the short text category within the Webis-CPC-21 dataset. This discrepancy suggests that the superior results observed on Webis-CPC-21 may be because the text samples in MSRP are even shorter than those in the short text category of Webis-CPC-21, 36-50 words, respectively.

In terms of examining SBERT as a state-of-the-art transformer model for the PI task, the results in Table 5.4 indicate a mixed performance across different datasets and text lengths. For the Webis-CPC-11 dataset, SBERT achieves an ACC of 78% and an F1-score of 83%. While this reflects an improvement over individual features like N-gram overlap and Sen2vec, which achieved 56% and 64% ACC and F1-scores of 68% and 63% respectively, SBERT falls short of outperforming traditional ML features such as TF-IDF, which achieved 87% ACC.

In contrast, the Webis-CPC-21 dataset shows better results for SBERT, with an ACC of 79% and an F1-score of 88%. This performance closely aligns with Sen2vec (80% ACC and 88% F1-score) and N-gram overlap (83% ACC and 90% F1-score), indicating that SBERT is affected by the diversity of text lengths present in Webis-CPC-21. Despite this, combining all features from earlier models produced similar results, suggesting that SBERT performs comparably to these semantic and structural features in handling various text lengths.

SBERT demonstrates remarkable performance on short and mid-length texts. For short texts, it achieves 89% ACC and 94% F1-score, outperforming previous methods like TF-IDF and Sen2vec, which both peaked at 79% ACC and 88% F1-score. Similarly, for mid-length texts, SBERT reaches the same 89% ACC and 94% F1-score, surpassing earlier results from features like N-gram overlap and Sen2vec, which maxed out at 85% ACC and 91% F1-score. These results highlight SBERT’s effectiveness in capturing semantic and contextual information in shorter text segments, making it highly suitable for sentence and paragraph-length and paragraph-level paraphrases. However, SBERT struggles in detecting Long-paraphrased text, achieving only 69% ACC and a 79% F1-score. These figures are notably lower than the performance of Sen2vec, which achieved 75% ACC and 89% F1-score. This suggests that SBERT's ability to handle longer passages is limited compared to semantic-based models, which excel in conveying the meaning of extended text segments.

Finally, in the MSRP dataset, which consists of short sentences, SBERT performs well, achieving 72% ACC and 82% F1-score. This result mirrors the performance of the Sen2vec model, indicating that SBERT is well-suited for sentence-level paraphrase identification but does not significantly outperform simpler models on short-text datasets.

In summary, the experiments reveal that text length plays a crucial role in the performance of PI models. Semantic-based features like Sen2vec excel in capturing the meaning of longer texts, while statistical features such as TF-IDF are better suited for shorter text segments.

Combining multiple features tends to yield better results, particularly for mid-length texts, but for very short or long texts, individual features often perform more effectively. These findings underscore the importance of choosing features that align with the specific text length to optimise the performance of PI models.

Table 5.4. The results of SBERT across multiple datasets, with accuracy (ACC) and F1-scores reported for each.

Transformer model	**SBERT**	
Evaluation	**ACC**	**F1**
Webis-CPC-11	78	83
Webis-CPC-21	79	88
Short text	89	94
Mid Text	89	94
Long Text	69	79
MSRP	72	82

Additionally, SBERT shows strong performance in short and mid-length text categories, surpassing traditional feature-based models in these cases. However, its effectiveness drops when applied to longer texts, where methods like Sen2vec may be more suitable for handling extended passages. This suggests that while SBERT is highly capable in some PI tasks, especially those involving shorter text (short and mid-length text), it may not be the best option for longer documents, emphasising the need for model selection based on text length in paraphrase detection tasks.

For more investigation, the results in Table 5.3 show a notable improvement over the baseline system for the Webis-CPC-11 dataset, which serves as the primary dataset for this study due to its diverse range of text lengths and paraphrase types. The achieved F1-score of 90% represents a 5% increase in F1-score on the Webis-CPC-21 dataset when considering the N-gram overlap feature. Moreover, the accuracy on the Webis-CPC-11 dataset surpasses the baseline by 3%, while the F1-score is slightly lower by 3%. However, it's crucial to emphasise that these results are achieved with just one feature (TF-IDF) as opposed to the baseline system that employed 10 different features. This demonstrates that the approach, even with fewer features, can deliver competitive and efficient performance, particularly in handling the dataset's varied text lengths and representing paragraph-level paraphrases. The ability to

achieve strong outcomes with a restricted feature set emphasises the model's robustness in PI tasks.

Furthermore, previous studies have employed pre-trained word2vec models with cosine similarity or soft cosine measures, considering the mean of the all-words vectors, for the MSRP and Webis-CPC-11 datasets (Vrbanec & Meštrović, 2021). In contrast, the approach taken here involves converting each piece of text into a single vector by summing all the word vectors within the text and then subtracting the paraphrased sentence vector of the source sentence vector (V_1-V_2). This method captures the overall semantic substance, thus representing the comprehensive meaning of the text. As demonstrated in Table 5.5, this proposed method yields better results than using cosine similarity or soft cosine measures with Sev2vec on both the MSRP and Webis-CPC-11 datasets.

Additionally, it is thought that SBERT would do better than the ML and DL methods used in this chapter because it had been trained on a large unsupervised text dataset and could handle up to 512 tokens, which is longer than the text in the sub-categories. Although SBERT requires more computational resources and power to operate, it performs better only for the Short and Mid text sub-categories, as well as for its equivalent on the MSRP dataset (Table 5.5). On the contrary, the results for Webis-CPC-11 and Webis-CPC-21 show a decrease in ACC of 9% and 4%, respectively, when implementing SBERT. For the Long text sub-category, SBERT's performance declines significantly compared to the method that relies solely on the N-gram overlap feature. This comparison highlights the limitations of using transformer models like SBERT. Unlike these models, which demand significant GPU power and high memory capacity, the method employed does not require such extensive resources. Additionally, ML and handcrafted features provide a more straightforward and accessible alternative to state-of-the-art models that require pre-training and specialised equipment, which is not always easily accessible.

Finally, it is observed that the efficiency of both ML and DL models improves when the text length is balanced, neither too short nor too long. Therefore, feature engineers must account for the text length when extracting features from text segments. By carefully considering text length, model performance can be enhanced, ensuring greater effectiveness across varying text lengths.

Table 5.5 Word2Vec on MSRP and Webis-CPC-11 dataset with different measure, *v1*and *v2* refer to the text 1 and text 2 vectors respectively, bold font represents the highest ACC and F1-score

Source	Measure	MSRP		Webis-CPC-11	
		ACC	F1	ACC	F1
(Vrbanec & Meštrović, 2021)	**Soft cosine (v1, v2)**	71	**82**	52	67
(Vrbanec & Meštrović, 2021)	**Cosine (v1, v2)**	69	80	52	67
Proposed method	**(v1- v2)**	**72**	**82**	**64**	**63**

5.4. Summary

This chapter addresses RQ1 and RQ2 by investigating the impact of text length on model performance when measuring the semantic similarity between different texts and identifying which features are most effective for short, mid, and long text lengths. The experiments conducted demonstrate that mid-length texts are particularly adept at conveying the semantic meaning of natural language compared to both short and long texts. To assess this, three distinct features are considered following the preprocessing of the text. The experiments utilised two datasets that differ in terms of text length and types of paraphrases.

Overall, no single feature emerged as the best performer across all categories. Nevertheless, mid-length texts consistently achieved the highest ACC and F1-score, both when analysing each feature separately and when examining combinations of features. Additionally, results indicate that long texts generally yield better performance than short texts and, surprisingly, even surpass the performance of state-of-the-art transformer models in some cases.

Given the findings of this study, there is a significant need for a new dataset specifically dedicated to mid-length texts. The research highlights that mid-length texts are particularly effective in capturing semantic meaning and achieving high ACC and F1-score. However, the existing datasets do not provide an adequate representation of mid-length texts, which constrains the ability to fully explore and optimise DL models for this text length. A new dataset with a robust collection of mid-length texts would not only address this gap but also advance research in PI by extending the focus beyond sentence-length and sentence-level paraphrases. This dataset would enable more detailed investigations into paragraph-length and paragraph-level paraphrases, allowing for a deeper understanding of how DL models perform with longer

text segments and leading to more refined and effective PI methods. In the next chapter, details of a novel method for creating a such dataset are provided and evaluated.

CHAPTER 6: DATASET CREATION AND EVALUATION

6.1. Introduction

Paragraph-level paraphrasis involves several transformations, including sentence reordering, splitting, and/ or merging, in addition to sentence-level paraphrasis. These processes modify the text both lexically and semantically while preserving its original meaning, adding complexity to paraphrase detection at this level. The combination of these changes makes it more challenging to maintain coherence and meaning, requiring more sophisticated methods to detect and evaluate paraphrased content accurately. Foundational work in this area is done by Al Saqaabi et al. (2022), who developed an algorithm specifically to detect paraphrasing at the paragraph-level (Chapter 5). However, their study is limited by the absence of suitable datasets, which is crucial for advancing research in this domain. The lack of these resources caused obstacles to further investigation of PI at the paragraph-level, particularly given the increasing significance of advanced LLMs that facilitate the processing and generation of lengthy text.

Given that no publicly available datasets currently address paragraph-level paraphrasing with the integration of LLMs, this thesis seeks to bridge that gap by addressing sentence reordering in a more systematic manner. Inter-sentence diversity is ensured by implementing three different algorithms based on an LLM. Specifically, this research applies the SOP from the ALBERT re-training model (Lan et al., 2020). In this approach, reconstructed paragraphs are generated based on the semantic and contextual relationships between the source sentences. This technique ensures the preservation of the paragraph's meaning despite significant transformations in the sentence structure.

To further diversify, the output text is lexically paraphrased using three different Transformer-based LLMs, each employing varying levels of MLM probabilities. These models are included because they are assessed based on their ability to generate paraphrased text that is challenging for humans to distinguish (Wahle, Ruas, Kirstein, et al., 2022). This approach generates multiple paraphrased versions of each source paragraph, contributing to the creation of the ALECS-SS dataset. The ALECS-SS dataset serves as a significant resource for

examining the complexities of paragraph-level paraphrasing, offering a diverse range of samples for model training and evaluation.

According to Ventayen (2023), artificial intelligence models are capable of generating paraphrased text that is both highly coherent and contextually accurate. While these capabilities present numerous advantages for NLG, they also pose significant challenges in the realm of academic integrity and content originality. The potential for AI-generated paraphrased text to be misused for producing plagiarised material is a growing concern. Becker et al. (2023) further emphasise that the sophistication of these models makes it increasingly difficult to distinguish between artificially paraphrased content and text written by humans (Becker et al., 2023).

In light of these challenges, the development of a large-scale paragraph-level paraphrase dataset is not only beneficial but essential. Such a dataset is critical for training DL models capable of reliably detecting paraphrases, which represents a key step in identifying plagiarism—particularly in cases where AI systems generate paraphrased versions of source text without proper attribution, thereby converting paraphrasing into a form of plagiarism. ALECS-SS responds to this need by providing approximately 27 paraphrased versions of each source paragraph, encompassing a range of paraphrasing intensity as measured by the number of shuffled sentences and the proportion of altered tokens. This breadth makes ALECS-SS valuable not only for advancing PI but also for pedagogical purposes. For instance, it can offer students graded examples of paraphrasing appropriate to their level of progress or support formative feedback by flagging paraphrases that remain overly close to the original text and suggesting deeper reformulation or improved coherence. By leveraging ALECS-SS, this thesis seeks to strengthen paragraph-level paraphrase identification and to contribute to safeguarding the integrity of written content.

Given the focus on creating and evaluating a large-scale dataset for paragraph-level paraphrases, the research questions explored in this chapter are as follows:

- ***RQ3:*** *Which of the three novel paragraph-level paraphrasing algorithms (SALACs) proposed preserves the source paragraph's meaning most effectively?*
- ***RQ4:*** *Is there a correlation between the similarity score assigned by human evaluators and the automatically generated coherence score used for paraphrase generation by the paragraph-level algorithms (SALACs)?*

- ***RQ5:*** *Is there a correlation between the automatically generated paraphrased paragraph's semantic similarity score and the human-written paragraph's semantic similarity score?*

6.2. Methodology

The ALECS-SS dataset is developed by collecting paragraphs from social science domains retrieved from Wikipedia. Articles related to linguistics and law are excluded due to their paraphrasing requirements, such as requiring the models to be specifically trained in legal language. Following this selection process, the dataset comprises 391,205 paragraphs, each ranging from 50 to 151 words in length, aligning with the average paragraph length in English (Larock MH et al., 1980) (refer to Figure 6.1), and includes a minimum of three sentences (refer to Figure 6.2). This sentence requirement ensures that the paraphrasing methodology, which is based on inter-sentence semantics, is distinct from sentence-level approaches utilised in other datasets. The primary objective of establishing ALECS-SS is to develop a dataset for training state-of-the-art NLP models to differentiate between human-written and machine-paraphrased texts to detect plagiarism in academic writing at the paragraph-level.

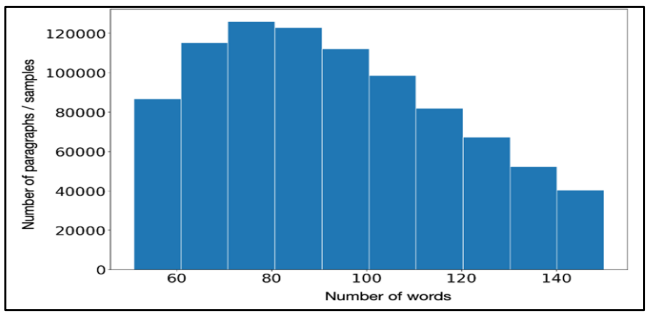

Figure 6.1 The ALECS-SS dataset described by the number of samples according to the word count in the paragraphs.

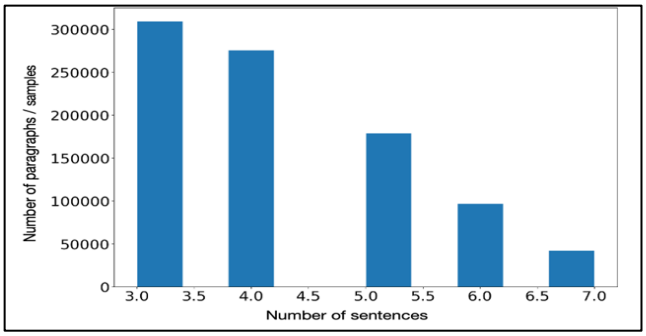

Figure 6.2 The ALECS-SS dataset described by the number of samples based on the count of sentences.

6.2.1. Inter-Sentence Paragraph Coherence Score

Evaluating sentence similarity and generating coherence scores has been extensively studied. Coherence is commonly defined through methods such as entity analysis (Elsner, M. & Charniak, E, 2011) and word co-occurrence analysis (Soricut & Marcu, 2006). Recently, LLMs have incorporated these tasks into their training processes. For example, BERT utilises NSP, while ALBERT employs SOP. This study employs SOP, as ALBERT demonstrates better performance in terms of measuring document coherence (Shen et al., 2021). It considers inter-sentence semantics and generates a coherence score based on the sentences' semantic vectors that represent the validity of the order between two sentences (Lan et al., 2020). In more detail, SOP is designed to predict whether one sentence logically follows another, integrating this task into ALBERT's training to enhance its understanding of contextual relationships between sentences. This framework operates by converting sentences into semantic vectors, which are then used to determine the probability of sentence order, resulting in a coherence score. This score is instrumental in assessing document coherence and improving text generation.

6.2.2. SALAC Algorithms

In this method, each paragraph is transformed into a fully connected directed graph (G) (refer to Figure 6.3) with sentences (S_1 *to* S_n; where n refers to the total number of sentences in the graph G) acting as nodes, as in Equation 6.1:

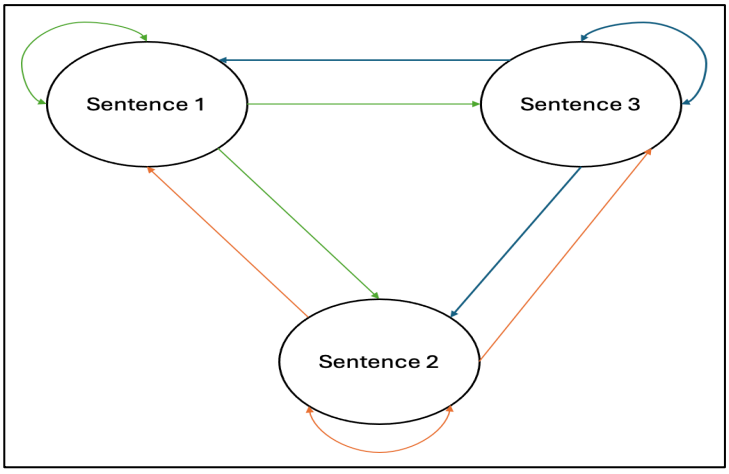

Figure 6.3: fully connected directed graph

$$V(G) = \{S_1, S_2, \dots, S_n\} \tag{6.1}$$

To achieve this, each sentence is first converted into a semantic vector that incorporates contextual information, utilising the ALBERT model. These sentence semantic vectors are then processed through the SOP framework. The output from the SOP framework provides a probability (*P*) indicating the likelihood of placing the *i-th* sentence (S_i) before the *j-th* sentence (S_j) within the paragraph. Equation 6.2 details this probability, which is referred to as the coherence score or semantic score in this thesis. The coherence score is used to determine the weight of the edge connecting the nodes in the graph.

$$P_{SOP}\left(S_s^i, S_s^j\right) = \begin{Bmatrix} P \geq \varepsilon, & i \neq j \\ 0, & i = j \end{Bmatrix} \tag{6.2}$$

It is noteworthy that the coherence score from sentence S1 to sentence S2 is not necessarily the same as the score from S2 to S1. This distinction is crucial, as it reflects the directional nature of sentence relationships within the paragraph.

Three algorithms are then implemented to reorder the sentences within a paragraph, with each algorithm proposing a path that visits each node exactly once, aiming to maximise the coherence score of the paragraph. After determining the optimal path, the sentences within the paragraph are reordered accordingly. Subsequently, human evaluation is conducted to interpret nuances in meaning, judge the relevance of sentence reordering, and understand how well the paraphrased content preserves the semantics of the source.

6.2.2.1. Inter-Sentence Reordering

Consider a fully connected directed graph G (refer to Figure 6.3) where nodes represent the sentences of a paragraph and edge weights are determined by the SOP probability. The objective is to generate a Hamiltonian path that visits each node exactly once without repetition (Dirac, 1952). It's important to highlight that in fully connected graphs, there are *n!* possible unique paths, where *n* is the number of sentences. To address this, three algorithms are developed, each proposing an optimal sentence sequence based on the semantic relationships between the source's sentences.

Algorithm SALAC1

This algorithm prioritises nodes based on the strength of the coherence scores, which are represented as the weights of the connections in the graph. The flowchart in Figure 6.4 outlines

the steps and conditions considered by SALAC1 to determine the optimal sentence order. For example, in a paragraph consisting of four sentences, as illustrated by the graph matrix in Figure 6.5, a coherence score of 0.7 indicates strong relationships between S1 and S2, as well as S2 and S4. Consequently, the generated path must ensure that S1 precedes S2, and S4 follows S2, placing S1 before S4. Other sentences can be inserted between these nodes without changing their established relationship. Additionally, the weakest coherence score in this scenario is 0.4, with other scores varying between this weakest value and the strongest. Notably, the diagonal values in the matrix are set to 0 to exclude paths from a sentence to itself.

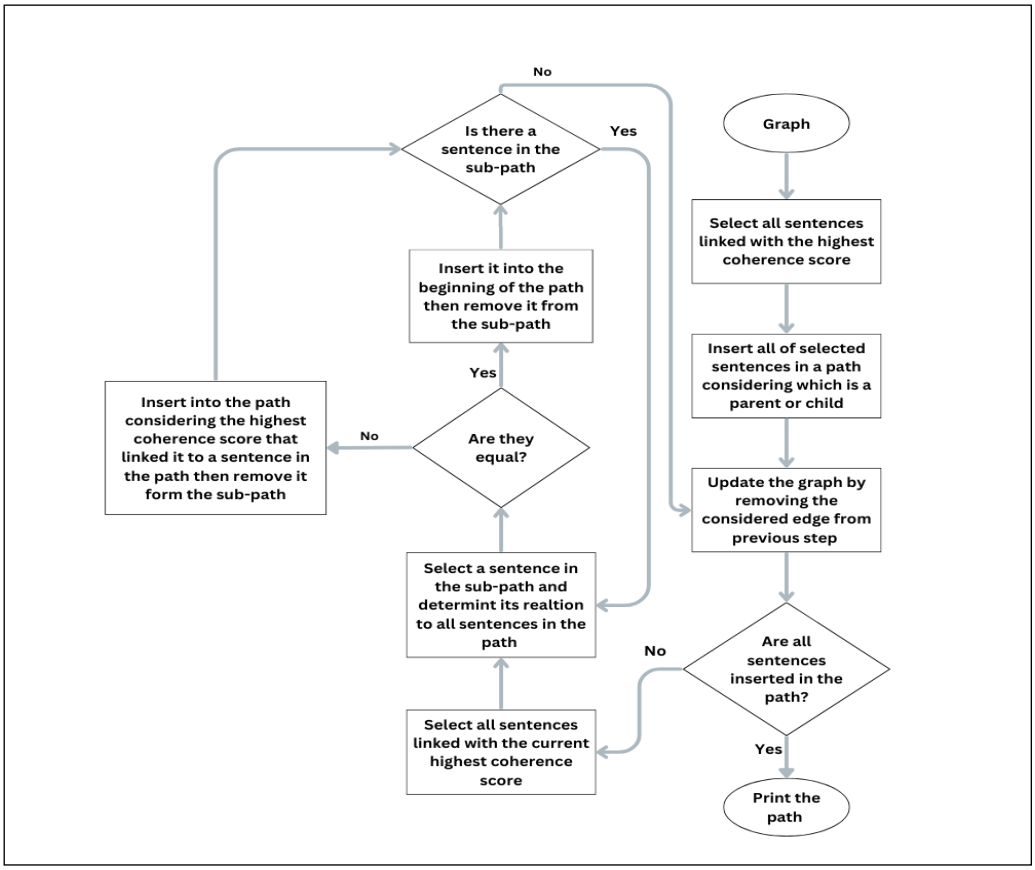

Figure 6.4: SALAC1 flowchart algorithm

	Sentence S1	Sentence S2	Sentence S3	Sentence 4
Sentence S1	0	**0.7**	0.6	0.5
Sentence S2	<u>0.4</u>	0	0.6	**0.7**
Sentence S3	0.6	0.6	0	0.6
Sentence S4	<u>0.4</u>	0.5	0.6	0

Figure 6.5: SALAC1 graph matrix example, scores in bold represent the strength coherence score while underlined scores represent the weakest coherence score

The first step, as illustrated in the flowchart (Figure 5.4), involves considering only the sentences connected by the strength of the coherence scores, such as S1-S2-S4 in the example. Following this, it is verified whether the path is complete or if there are any sentences not yet included. All coherence scores (graph edges) considered in the previous step are then removed, reducing the strength coherence score from 0.7 to 0.6 in this example. Subsequently, sentences linked by the updated strength coherence score are selected and inserted into the path based on their relationships with the sentences already in the path, taking into account their parent and child connections. In some cases, a node may have the same edge weight to all other nodes in the graph, allowing it to be positioned at any location in the path. For instance, in this example, S3 is connected to all sentences with a coherence score of 0.6, so it can be placed at the beginning, middle, or end of the path.

Another scenario shown in the flowchart occurs when a sentence has a strong connection to the second sentence in the path but a weak connection to the first sentence. In such cases, the sentence is treated as a parent to the second sentence (indicating it should be inserted before the second sentence) and as a child to the first sentence (indicating it must follow the first sentence).

Algorithm SALAC2

SALAC2 examines all possible paths in the graph and selects the path with the highest coherence score. For instance, if the path S1-S3-S2-S5-S4 is considered, where S1 is the first sentence of the source paragraph followed by S3, S2, S5, and S4 in sequence, the coherence score is calculated by summing the coherence values of the edges connecting each sentence to the subsequent one in the path (refer to Equation 6.3). Based on this calculation, SALAC2 identifies and suggests the optimal path with the highest paragraph coherence score.

$$COH = \sum_{i}^{n-1} P_{SOP}\left(S_s^i, S_s^{i+1}\right) \tag{6.3}$$

Algorithm SALAC3

SALAC3 evaluates all unique paths within the graph and identifies the path with the highest coherence score. It calculates this score by considering the relationships between parent nodes and their child nodes, Equation 6.4. Unlike SALAC2, which computes the coherence score by summing the weights of edges between each sentence and the one immediately following it,

SALAC3 measures the paragraph's coherence by aggregating the weights of all edges from a node to all subsequent nodes within the path. This approach allows SALAC3 to provide a different perspective on the overall coherence of the paragraph.

$$COH = \sum_{i}^{n-1} \sum_{j=i+1}^{n} P_{SOP}\left(S_s^i, S_s^j\right) \tag{6.4}$$

The SALAC algorithms generate three reconstructed paragraphs for each source paragraph, with one reconstructed paragraph produced by each algorithm. In the following section, the process for lexically paraphrasing each reordered sentence while considering intra-sentence relations is described, resulting in multiple paraphrased versions for each source paragraph.

6.2.3. Intra-Sentence Masking (Paraphrasing)

To build a dataset for paragraph-level paraphrasing, three state-of-the-art LLMs are employed as paraphrase-generation tools. Specifically, BERT (Devlin et al., 2019), often used as a baseline in NLP research, RoBERTa (Y. Liu et al., 2019) an extension of BERT designed for handling longer documents, and Longformer (Beltagy et al., 2020) developed primarily for long documents, are utilised. These models take as input the paragraph's sentences that have been shuffled by the SALAC algorithms. To increase the diversity of the ALECS-SS dataset, different levels of MLM probabilities are applied across all three LLMs. MLM involves masking specific words in the input sequences and prompting the model to predict the most likely words for text completion based on the semantic context (refer to Chapter 4). To ensure that the paraphrased content remains accurate and avoids the generation of incorrect information, named entities and punctuation marks such as brackets, digits, currency symbols, and quotation marks are excluded during the paraphrasing process.

The selection of transformer models is guided by their relevance to the task and their capacity to process text of varying lengths. Key factors influencing this selection included model architecture, the ability to capture contextual relationships, and suitability for handling paragraph-length text. These criteria ensured that the chosen models effectively addressed the specific challenges associated with paragraph-level paraphrase generation. In particular, models incorporating masked language prioritised, as this feature is essential for modifying the lexical layer of text segments. BERT is included as a baseline due to its extensive use and strong performance in sentence-level PI tasks (Chapter 3), serving as a reference for evaluating the performance of models designed for longer text segments. RoBERTa is selected for its

improved training methodology and dynamic masking strategy, which enhances contextual representations compared to BERT, as demonstrated in Chapter 3. Longformer is chosen for its capability to process extended text sequences while retaining the MLM approach. It was developed to mitigate the limitations of traditional transformers, which struggle with long text sequences due to the quadratic complexity of self-attention mechanisms. Longformer integrates a sliding window attention mechanism. This enhancement enables efficient linear scaling, allowing for the effective processing of significantly longer texts. Its ability to capture both intra-sentence and inter-sentence semantics makes it particularly beneficial for paragraph-level NLP tasks. Unlike other long-document transformers, Longformer optimises computational efficiency while preserving strong contextual representations, making it a suitable model for paragraph-level paraphrase generation and identification.

6.3. Dataset Evaluation

6.3.1. Human Evaluation

Manual evaluation studies are a standard practice in NLG tasks, often used to verify the effectiveness of the methods applied (Birch et al., 2009; Hämäläinen & Alnajjar, 2021). In this research, a human evaluation method is employed, as detailed below, to carefully assess the output generated by the SALAC algorithms.

The main objective of the evaluation is to assess the semantic similarity between the generated paragraph and the source paragraph. This ensures the preservation of the source text's meaning, even when the syntactic structure of the sentences is changed. Thus, the analysis is conducted to achieve this objective, focusing on identifying semantic differences that may arise due to sentence reordering. A random sample of 100 source paragraphs and their paraphrased versions generated by each SALAC algorithm is selected from the ALECS-SS dataset. In details, the process of selecting the 100 paragraphs from Wikipedia for the evaluation phase followed a structured procedure. First, paragraphs are extracted from psychology-related articles to ensure topical consistency and to provide text representative of academic content. These paragraphs are then filtered to retain only those containing at least three sentences and falling within the average range of paragraph length, thereby excluding very short or unusually long passages. The filtered set of paragraphs is incorporated into the ALECS-SS dataset. From this pool, a random selection algorithm is applied to ensure unbiased sampling, and 100

paragraphs were chosen to form the evaluation subset. This number was chosen based on the median sample size commonly used in NLG evaluations (van der Lee et al., 2021). Then, each paraphrased paragraph is reviewed by three independent evaluators to enhance the reliability of the evaluation as recommended in the evaluation literature (Van Enschot et al., 2017; Potter & Levine-Donnerstein, 1999). The evaluation process is based on majority voting, meaning that the final semantic similarity score between the source paragraph and the reconstructed paragraph is determined by the consensus of at least two out of the three evaluators. This method is designed to ensure an unbiased and accurate reflection of the output's quality.

A total of six evaluators is selected for this task, each highly proficient in written English and possessing a high level of education. This level of proficiency is considered essential given that the texts are drawn from Wikipedia articles, which are intended for a general readership rather than for domain experts. Specifically, the evaluators are therefore instructed to compare the semantic content of each generated paragraph with its source, focusing on whether meaning is preserved despite the structural changes introduced through sentence reordering. The decision not to use expert evaluators is deliberate, since the aim of the study is to assess paraphrasing quality at the paragraph-level rather than disciplinary accuracy, and the SALAC algorithms are designed to generate paraphrases without regard to text domain, as they are not fine-tuned or trained on specific disciplines but developed as a general framework for paragraph-level paraphrasing. Moreover, prior studies have shown that expert judgments often introduce bias and variance, making them less representative of general readers (Amidei et al., 2018b; Belz & Reiter, 2006). Thus, this human evaluation provides strong evidence that the SALAC algorithms maintain semantic integrity while producing syntactically distinct paraphrases. For more details, see Appendix A.

According to van der Lee et al. (2021), it is acknowledged that complex concepts, such as semantic similarity, cannot be sufficiently represented by a singular arbitrary rating. Instead, a more sophisticated approach, specifically a 5-point Likert scale, is considered suitable, as suggested by (Amidei et al., 2018; Potthast, Stein, et al., 2010). Therefore, the participants are instructed to assign a score to each reconstructed paragraph using a 5-point Likert scale, with each point on the scale representing a defined value of semantic similarity. The scale is designed as follows:

- **5: Almost identical**: The paragraph is nearly a perfect match to the source, with only negligible differences.

- **4: Very similar**: The meaning is very close to the source, with only minor alterations.
- **3: Similar**: Major changes to the meaning are present, but the general context is retained.
- **2: Dissimilar**: Significant changes to the meaning make it noticeably different from the source.
- **1: Extremely different**: The paragraph bears little to no resemblance to the source, with substantial changes in meaning.

The University's ethics committee approved the experiment, ensuring adherence to ethical standards, and the evaluation process took approximately three hours to complete.

6.3.1.1. Inter-Annotator Agreement (IAA) Correlation.

The classification of IAA values, as outlined by (Landis & Koch, 1977), is crucial for determining the reliability of the evaluations (Table 6-1). IAA values less than 0 are considered poor, indicating that the annotators' ratings are not consistent and may even be worse than random chance. When the IAA value falls between 0 and 0.2, the agreement is deemed slight. This range suggests minimal agreement between annotators. Values between 0.2 and 0.4 indicate fair agreement: although there is noticeable consistency in the annotators' ratings, it is not particularly strong, highlighting the need for improvement. Moderate agreement, represented by IAA values between 0.4 and 0.6, shows a decent level of reliability: annotators are moderately consistent, suggesting that while their evaluations are reliable, there is still room for enhancement. Substantial agreement is observed when IAA values lie between 0.6 and 0.8; this range indicates a high level of consistency among annotators, implying that their ratings are reliable and trustworthy. Finally, IAA values exceeding 0.8 denote almost perfect agreement. Such high values reflect an exceptionally high consistency among annotators, with their evaluations being nearly identical.

In NLG, Amidei et al. (2018) suggested that an acceptable range for IAA falls between 0.3 and 0.5, with higher values being more desirable. To evaluate the consistency among annotators in this study, the kappa (k)coefficient is applied as a statistical test for IAA. This test is conducted across groups of three evaluators assessing various generated paragraphs. The analysis revealed an acceptable IAA correlation of $k = 0.32$, in line with expectations for

complex tasks like evaluating semantic similarity. Amidei et al. (2018) noted that IAA tends to be lower in tasks involving language complexity, such as semantic similarity assessments.

Table 6.1: Data from (Landis & Koch, 1977)

IAA value	IAA interpretation
$\text{IAA} < 0$	Poor
$0 \leq \text{IAA} \leq 0.2$	Slight
$0.2 < \text{IAA} \leq 0.4$	Fair
$0.4 < \text{IAA} \leq 0.6$	Moderate
$0.6 < \text{IAA} \leq 0.8$	Substantial
$0.8 < \text{IAA} \leq 1$	Almost Perfect

Despite the initial IAA score, the correlation significantly improved to $k = 0.81$ when the 5-point Likert scale ratings are grouped into two broader categories based on their definitions: scores of 1 and 2 are combined into one group (Group A), and scores of 3,4 and 5 are combined into another group (Group B). This grouping highlighted a stronger consensus ($k = 0.8$) among evaluators when considering broader distinctions in their assessments. This notable rise in agreement is likely due to the binary conversion process, since most evaluators gave scores between 3 and 5 (see Table 6.2). By grouping these scores, the conversion made the evaluation criteria more straightforward, which helped minimise variation and resulted in a greater level of agreement. This approach reflects the naturally subjective aspect of human comprehension, as individuals often interpret semantic subtleties in different ways.

6.3.2. Automatic Evaluation

Two methods are employed to calculate the coherence score of the generated paragraphs, used to evaluate the outputs of the SALAC algorithms. The first method focuses on the direct relationship between consecutive sentences in the path (i.e., the paragraph), where the coherence score is calculated based on the link between a sentence and the one that immediately follows it. The second method takes a broader approach by evaluating the overall coherence of the paragraph. This method calculates the total coherence score by summing the relationships between each sentence and all the sentences that follow it within the paragraph. Both methods aim to provide a comprehensive assessment of the paragraph's structural coherence.

When humans read, they are more likely to notice the coherence between consecutive sentences, as described in the first method. This method evaluates how each sentence

transitions to the next, which directly affects the readability and flow of a paragraph. Readers typically process text linearly, making the connection between adjacent sentences critical for maintaining smooth and logical progression.

However, the second method, which considers the coherence of each sentence in relation to all subsequent sentences, provides a more holistic evaluation of the paragraph's structure. While it may not be as immediately noticeable to the reader, it plays a significant role in ensuring the paragraph maintains consistency throughout. This method captures deeper, more complex relationships between sentences, which helps to create a coherent overall narrative, especially in longer paragraphs. As this thesis considers a paragraph's coherence as a whole, the second method is more aligned with the broader context and organisation of ideas, making it particularly effective for evaluating the overall flow and meaning of the text.

In essence, while the first method excels at measuring sentence-to-sentence coherence, the second method offers a more comprehensive view of how well the entire paragraph holds together, which is essential for maintaining thematic and semantic consistency across multiple sentences. Thus, the second method is applied and considered in this chapter.

Generally, the analysis of the data collected from both human and automated evaluations is conducted from three distinct perspectives. First, the effectiveness of the paraphrase-generating algorithms (SALAC1, SALAC2, and SALAC3) is assessed to determine their operational accuracy and to identify which algorithm performed most accurately. Second, the correlation between the coherence scores assigned by human evaluators and those generated automatically is examined to understand how well automated metrics align with human judgment. Third, the relationship between the similarity score of the paraphrased paragraphs and the similarity score of the original human-written paragraphs is investigated to evaluate how closely the generated paraphrases match the source content. The findings from this analysis are discussed in the next section.

6.4. Results and Discussion

6.4.1. SALAC Algorithms Efficiency

The comparison of the algorithms' performance, as shown in Table 6.2, provides insights into their effectiveness based on human evaluations that produced 300 scores per algorithm.

Table 6.2: Distribution of 300 votes of the scores given by human annotators (with 1-5 ranging from: 1=’extremely different’; 5= ’almost identical’)

Score	1	2	3	4	5	A	B
SALAC1	1%	9%	27%	24%	39%	10%	90%
SALAC2	3%	15%	24%	28%	30%	18%	82%
SALAC 3	3%	13%	21%	23%	40%	16%	84%

For “Extremely different, Score 1” ratings, SALAC1 demonstrates superior performance, with only 1% of paragraphs falling into this category. In contrast, SALAC2 and SALAC3 each have 3% of paragraphs rated as extremely different semantically from the source, indicating that SALAC1 better preserves the meaning of the source text.

In terms of “Dissimilar, Score 2” ratings, SALAC2 produces the highest proportion at 15%, compared to SALAC3's 13% and SALAC1's 9%. This suggests that SALAC2 often results in paragraphs with greater deviations from the source material compared to the other algorithms.

For “Similar, with major changes, score 3”, SALAC1 and SALAC2 have similar proportions, with 27% and 24% respectively, while SALAC3 has a lower rate at 21%. This reflects a moderate level of similarity with significant alterations across all algorithms.

Regarding “Very similar, Score 4” results, SALAC2 achieves the highest proportion at 28%, whereas SALAC1 and SALAC3 are close, at 24% and 23% respectively. This indicates that SALAC2 is somewhat more effective at maintaining similarity with minor changes compared to SALAC1 and SALAC3.

In the category of “Almost identical, Score 5”, SALAC3 led with 40% of paragraphs rated as such, slightly surpassing SALAC1, which has 39%, and SALAC2, which has 30%. This highlights the superior ability of SALAC1 and SALAC3 to closely mirror the source text.

For further analysis, the scores are grouped into two categories according to their definitions “A” (dissimilar) and “B” (similar), SALAC1 achieves 10% of paragraphs in the “A” category and 90% in the “B” category. SALAC2 shows a slightly higher proportion in category A (18%) and 82% in B, while SALAC3 have 16% in A and 84% in B. Although all algorithms primarily yield paragraphs in the “B” category, indicating similarity, SALAC1 and SALAC3 show higher proportions of similar paragraphs compared to SALAC2. Nevertheless, SALAC2 consistently maintains a high degree of alignment with the source meaning.

In summary, SALAC1 emerges as the most effective algorithm in preserving the source meaning, as evidenced by its high percentage of “similar” ratings and low percentage of “dissimilar”, closely followed by SALAC3. SALAC2, while still effective, tends to generate outputs with more divergence from the source meaning compared to the other two algorithms. This difference may be attributed to the number of shuffled sentences processed by each algorithm, as SALAC2 often alters a greater proportion of the sentence order in the paragraph (see Figure 6.6).

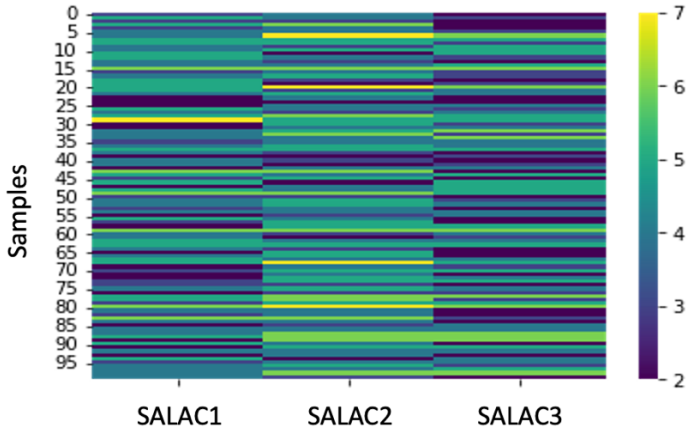

Figure 6.6 Shuffled sentence distribution for each algorithm. Colour indicates the number of shuffled sentences, with darker shades representing fewer shuffles and lighter shades representing more. The figure highlights variation in shuffling behaviour across different algorithmic settings.

6.4.2. Correlation of Human Ratings and Automatic Coherence Scores.

The relationship between the coherence scores assigned by human evaluators and those generated automatically is examined, as presented in Table 6.3. To quantify the strength and direction of this relationship, the Pearson correlation coefficient is utilised. This statistical measure is commonly employed to assess how closely two variables are related, producing a value that ranges from -1 to 1. A value of -1 indicates a perfect negative correlation, meaning that as one variable increases, the other decreases. Conversely, a value of 1 reflects a perfect positive correlation, where both variables increase in parallel. A value of 0 indicates no correlation between the variables. This method provided insight into how closely aligned the human-assigned coherence scores are with the automatically generated ones, offering a clearer understanding of the system's reliability in mimicking human judgment.

Furthermore, the number of sentences within a paragraph plays a crucial role in determining the paragraph's overall coherence score. This score is calculated based on the coherence between individual sentences within the paragraph. For instance, if we have a paragraph with 7 sentences, each having a coherence score of 4, the total coherence score would be 24. On the other hand, a paragraph with only 4 sentences, but with a coherence score of 6 for each sentence, would have a total score of 18. This can be misleading because, despite the second paragraph having a lower total score, its sentences are more coherent than those in the first example. Thus, it is decided to account for the number of sentences when assessing the correlation between the automatic paragraph coherence score and the human-evaluated score. Interestingly, as the number of sentences increased, the correlation between the automated and human-evaluated coherence scores improved, as shown in the results for paragraphs with seven sentences in Table 6.3. Moreover, the correlation also improved when the total coherence score is divided by the number of sentences, providing a more accurate reflection of the overall coherence by normalising for paragraph length. This adjustment helps mitigate the misleading effects of higher sentence counts on the total score.

In detail, the results in Table 6.3 show that the Pearson correlation values indicate varying levels of agreement between the coherence scores assigned by human evaluators and the automatically generated scores across different SALAC algorithms. For the 5-point Likert scale, SALAC3 shows the strongest correlation (0.157) across all samples, while SALAC1 and SALAC2 present weaker correlations (0.050 and 0.003, respectively). However, when focusing on samples with seven sentences, SALAC1 and SALAC3 display significantly higher correlations (0.409 and 0.427, respectively), suggesting that these algorithms are more effective with longer paragraphs, while SALAC2 remains relatively low at 0.055.

When scores are grouped, SALAC3 continues to outperform, with correlations of 0.181 for all samples and 0.435 for seven-sentence samples. SALAC1, however, exhibits an even stronger correlation in the seven-sentence category, reaching 0.698, which reflects its consistent ability to preserve the meaning across these specific paragraph lengths.

Examining coherence scores relative to the number of sentences, SALAC1 and SALAC3 show similar correlations (0.231 and 0.230, respectively) across all samples, but both increase markedly in the seven-sentence category, with SALAC1 at 0.665 and SALAC3 at 0.470. SALAC2, while the weakest overall (0.163 for all samples and 0.280 for seven sentences), performs better in this category than with others.

Table 6.3: Correlation of coherence scores between human-written paragraphs to generated shuffled sentences paragraphs

	Pearson Correlation					
Scores	**All samples**			**Samples with 7 sentences**		
algorithm	**SALAC1**	**SALAC2**	**SALAC3**	**SALAC1**	**SALAC2**	**SALAC3**
5-point Likert scale	0.050	0.003	0.157	0.409	0.055	0.427
Correlation for A/B grouped scores	0.103	0.078	0.181	0.698	0.135	0.435
Correlation for A/B grouped scores / number of sentences	0.231	0.163	0.230	0.665	0.280	0.470
Minimum coherence score between sentences	0.797	0.032	0.625			
Maximum coherence score between sentences	0.457	0.051	0.507			

Building on the previous analysis of coherence scores, the effect of the highest and lowest coherence scores between sentences in the generated paragraphs is further investigated. This analysis focused on the influence of both the strongest and weakest sentence connections. By examining these maximum and minimum values, more detailed insights are gained into how varying sentence relationships impact the correlation between human-evaluated coherence scores and automatically generated coherence scores. Specifically, the coherence score between sentences highlights a significant contrast: SALAC1 demonstrates the highest correlation in terms of considering only the minimum coherence score in the paragraph (0.797), significantly outperforming SALAC3 (0.625) and SALAC2 (0.032). When looking at the maximum coherence score, SALAC1 and SALAC3 are more comparable (0.457 and 0.507, respectively), while SALAC2 again lags behind at 0.051. This decrease in correlation, when considering the maximum score, can be attributed to the fact that these algorithms primarily focus on sentences linked by the highest coherence score when shuffling the paragraphs' sentences. As a result, the maximum score is almost always present in the generated paragraphs, offering limited variation in coherence. In contrast, the minimum coherence score plays a more significant role in differentiating the overall coherence of the paragraph. It is the fluctuations in the weakest sentence connections that make a more noticeable difference in the correlation

between human-evaluated and algorithm-generated coherence scores, thus providing a better measure of coherence consistency.

In summary, these results suggest that SALAC1 and SALAC3 excel in ensuring a consistently strong correlation by considering only the minimum coherence scores of paraphrased sentences. Additionally, they generally offer a better correlation, particularly with seven-sentence paragraphs. SALAC2 consistently shows weaker performance across most metrics.

6.4.3. Correlation of Machine-Paraphrased and Human-written Coherence Scores.

Assessing the correlation between the coherence scores of generated paragraphs and their corresponding human-written paragraphs is crucial, especially when producing paragraph-level paraphrases based on source human-written content. This comparison provides insight into how well the paraphrasing algorithms preserve the semantics of the source material. To achieve this, two methods are applied, as outlined previously in the automatic evaluation section 6.3.2, to measure the coherence score of the source paragraphs. These scores are then compared to the coherence scores of the generated paragraphs for each algorithm.

As illustrated in Figure 6.7, the Pearson correlation coefficients for SALAC1, SALAC3, and SALAC2 are 0.89, 0.80, and 0.69, respectively. This demonstrates that SALAC1 and SALAC3 have stronger correlations with the human-written coherence scores than SALAC2. The higher correlation for SALAC1 and SALAC3 suggests that these algorithms are more successful in preserving the semantics of the source text. Notably, these results are achieved using the second method, as described in section 6.3.2, which takes into account the cumulative relationships between sentences, proving to be more effective for all algorithms compared to the first method outlined in section 6.3.2. This further emphasises the importance of considering broader sentence relationships when evaluating paragraph semantics.

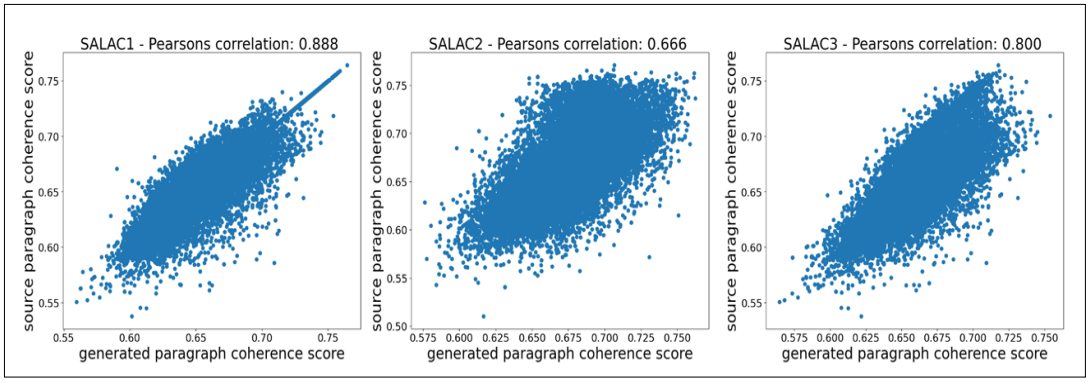

Figure 6.7: Correlation between the generated paragraph (SALACs output) to the human-written paragraph (source).

6.4.4. Mask Applied Method

Utilising the MLM approach, SALAC's output paragraphs are rewritten using three different language models, namely: BERT, RoBERTa, and Longformer, at masking rates of 15%, 20%, and 30%. The main objective of this method is to modify the lexical content of the paragraphs while preserving their original semantic meaning. This allows the generation of paragraph-level paraphrases, where both the structure and vocabulary of the paragraph are changed while the meaning remains preserved. As a result, 27 paraphrased texts are paraphrased at the paragraph-level of each source, exhibiting variations in syntax and lexical choices while maintaining the meaning of the source paragraph (Figure 6.8).

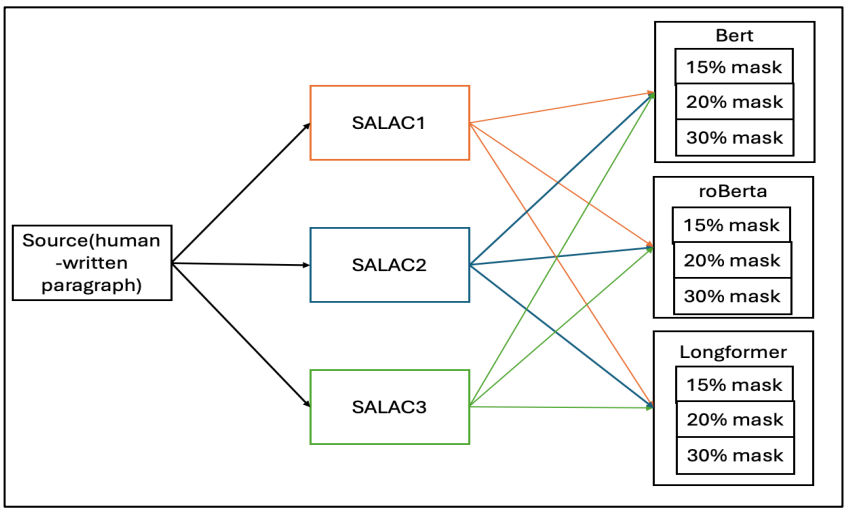

Figure 6.8: Each LLM (BERT, RoBERTa, Longformer) generates three paraphrased paragraphs for each input paragraph

The differing masking rates provide different levels of modification to the text, evaluating each model's capacity to preserve semantic integrity despite substantial changes in syntactic structure. The Pearson correlation coefficients are computed to measure the alignment between the coherence scores of the paraphrased and source paragraphs, as shown in Figures 6.9 – 6.17.

In the case of BERT, the correlation between source and paraphrased paragraphs remained relatively strong at lower MLM rates. Specifically, Pearson correlation values peaked at 0.802 and 0.775 for a 15% masking rate in the SALAC1 and SALAC3 outputs, respectively (Figures 6.9 and 6.11). However, as the masking rate increased to 30%, the correlation diminished significantly, reaching a low of 0.472 in the SALAC2 output (Figure 6.10). This trend indicates that BERT's ability to maintain the coherence and semantic integrity of the source text diminishes as more words are masked.

RoBERTa, in comparison, demonstrated a somewhat weaker correlation overall. At the 15% masking rate, RoBERTa achieved its highest Pearson correlations of 0.715 in the SALAC1 output (Figure 6.12) and 0.547 in the SALAC3 output (Figure 6.14). However, similar to BERT, the performance declined as the masking rate increased, with the correlation dropping to a low of 0.382 in the SALAC2 output at 30% masking (Figure 6.13). This significant decline suggests that RoBERTa struggles more with maintaining semantic coherence as the text is increasingly altered, particularly in cases like the SALAC2 output, where extensive sentence reordering occurs.

Longformer, by contrast, exhibited the strongest performance across all masking rates. At the 15% MLM rate, Longformer achieved the highest Pearson correlations of 0.835 and 0.793 in the SALAC1 and SALAC3 outputs, respectively (Figures 6.15 and 6.17), reflecting a strong preservation of coherence between source and paraphrased paragraphs. Even at higher masking rates, Longformer maintained relatively high correlations, with the lowest correlation being 0.533 in the SALAC2 output at 30% masking (Figure 6.16). These results suggest that Longformer is more robust in handling high masking rates while still effectively preserving the semantic consistency and coherence of the source text.

These findings indicate that LLMs can maintain paragraph semantics more effectively at lower masking rates but become less reliable as the degree of masking increases. In detail, SALAC1 produced constructed paragraphs conveying almost the same source meaning, SALAC3 following it, and SALAC2 remains behind. Thus, implementing each model with a

low level of mask resulted in a small impact on the paragraph semantics (SALAC1 correlation to the source is 0.888 after applying the Longformer with 15% mask probability, becoming 0.835).

Furthermore, the findings underscore that while all models exhibited a decline in performance as the masking rate increased, Longformer demonstrated greater resilience, consistently maintaining higher correlation values than both BERT and RoBERTa. This is especially evident in scenarios that involve more significant textual modifications, such as those seen in the SALAC2 output. The superior performance of Longformer could be attributed to its unique attention mechanism. Unlike BERT and RoBERTa, which employ a standard self-attention mechanism, Longformer utilises a sliding window attention approach that allows it to efficiently capture dependencies over longer text spans. This attention structure likely provides Longformer with an advantage when handling more substantial changes in the paragraphs, as it enables the model to preserve semantic consistency across larger and more complex paraphrasing tasks.

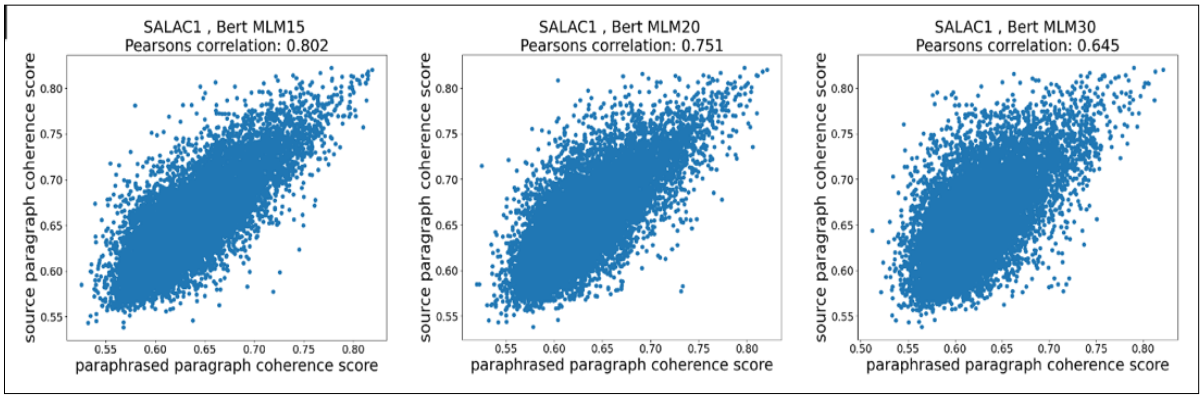

Figure 6.9 Correlation between the paraphrased paragraph generated by BERT and the human-written (Source) paragraph seen in SALAC1 outputs

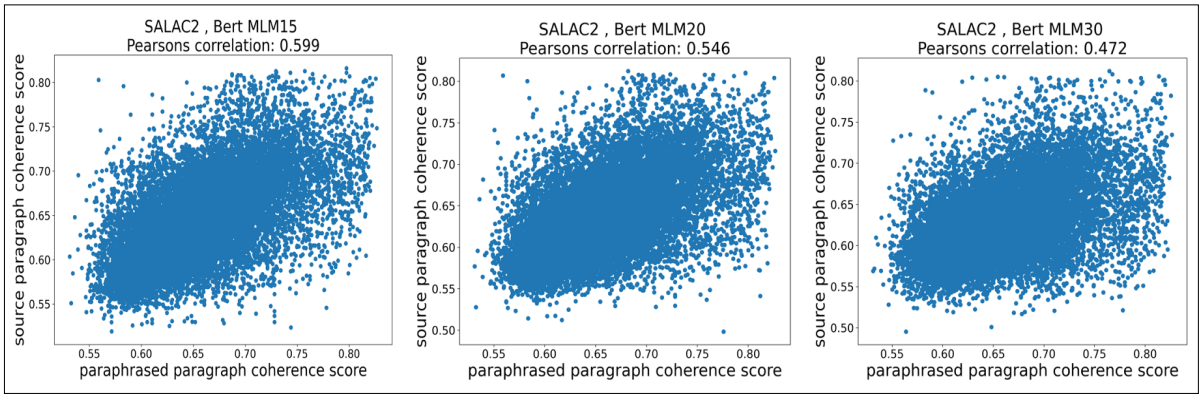

Figure 6.10 Correlation between the paraphrased paragraph generated by BERT and the human-written (Source) paragraph seen in SALAC2 outputs

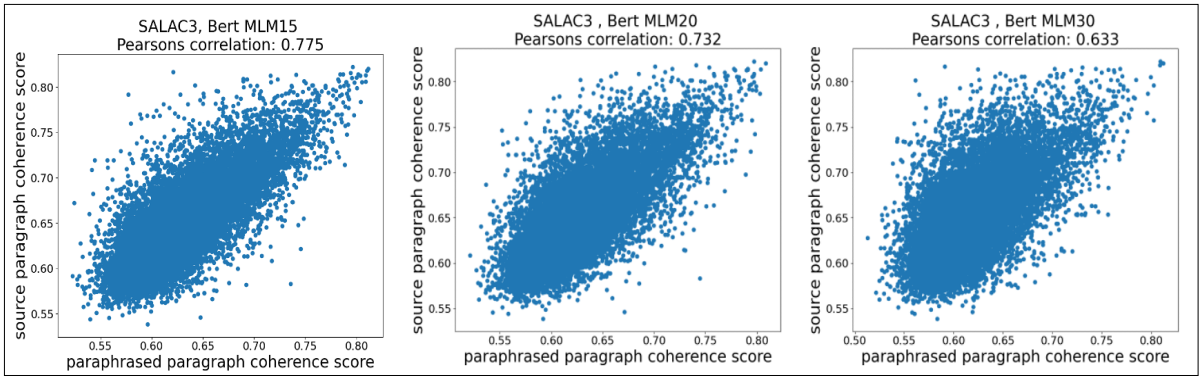

Figure 6.11 Correlation between the paraphrased paragraph generated by BERT and the human-written (Source) paragraph seen in SALAC3 outputs

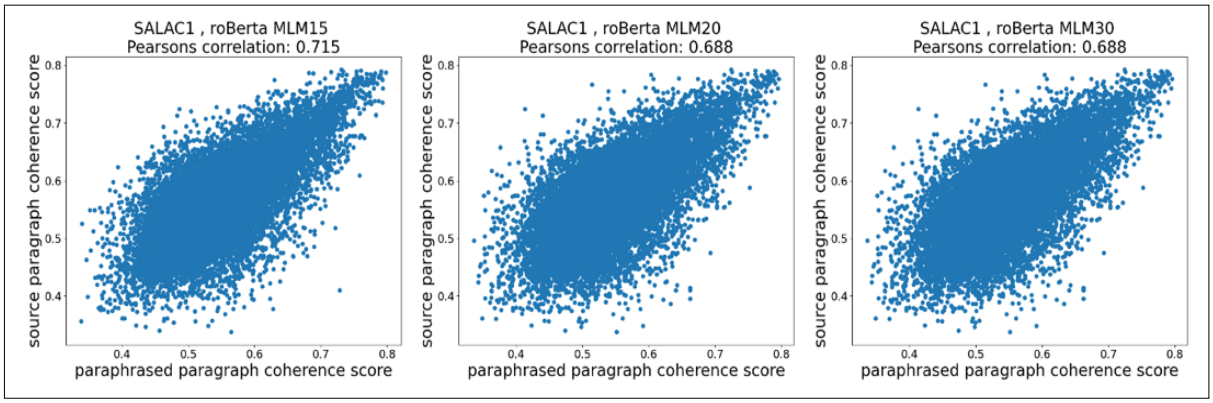

Figure 6.12 Correlation between the paraphrased paragraph generated by RoBERTa and the human-written (Source) paragraph seen in SALAC1 outputs

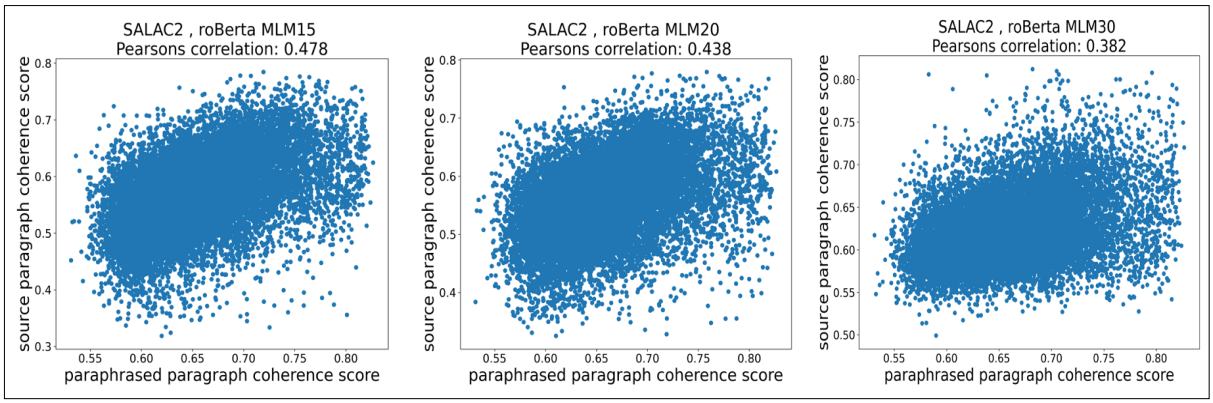

Figure 6.13 Correlation between the paraphrased paragraph generated by RoBERTa and the human-written (Source) paragraph seen in SALAC2 outputs

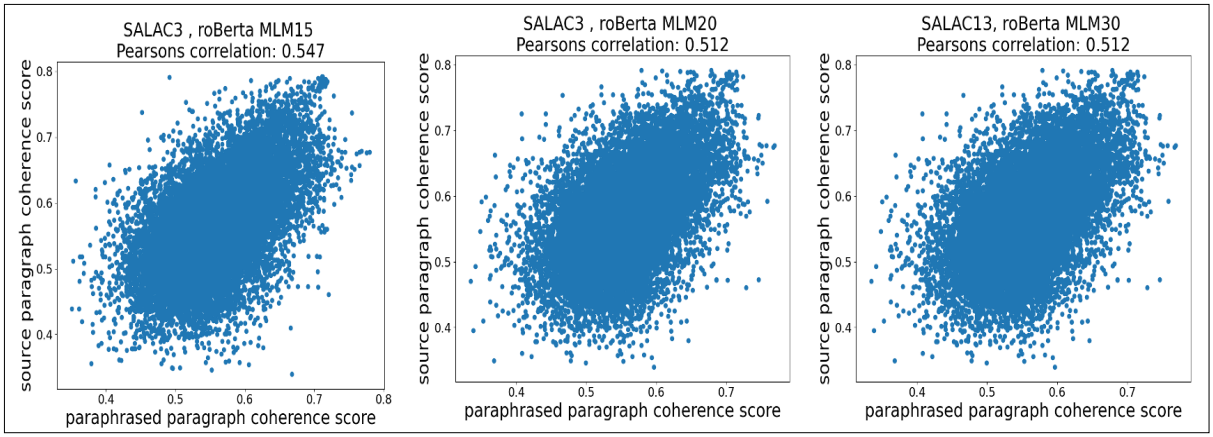

Figure 6.14 Correlation between the paraphrased paragraph generated by RoBERTa and the human-written (Source) paragraph seen in SALAC3 outputs

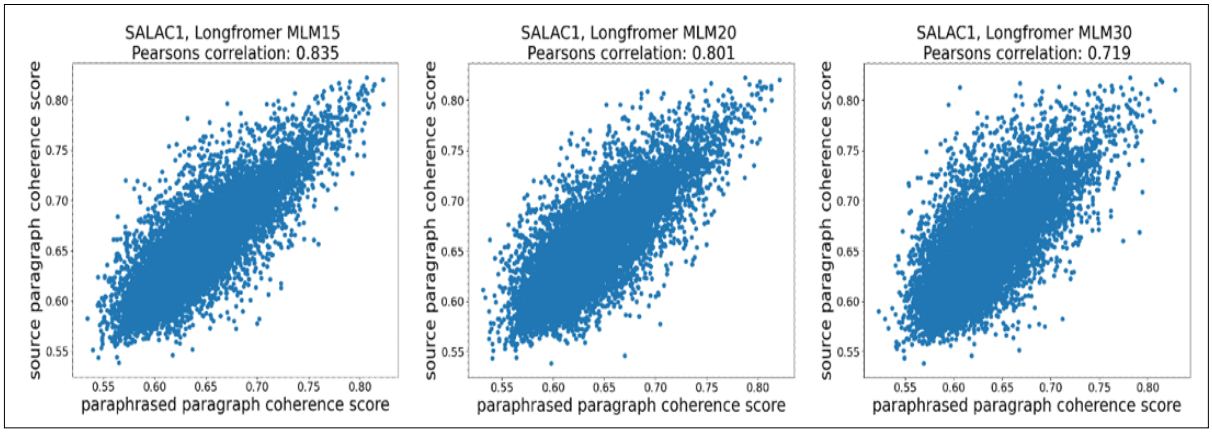

Figure 6.15 Correlation between the paraphrased paragraph generated by Longformer and the human-written (Source) paragraph seen in SALAC1 outputs

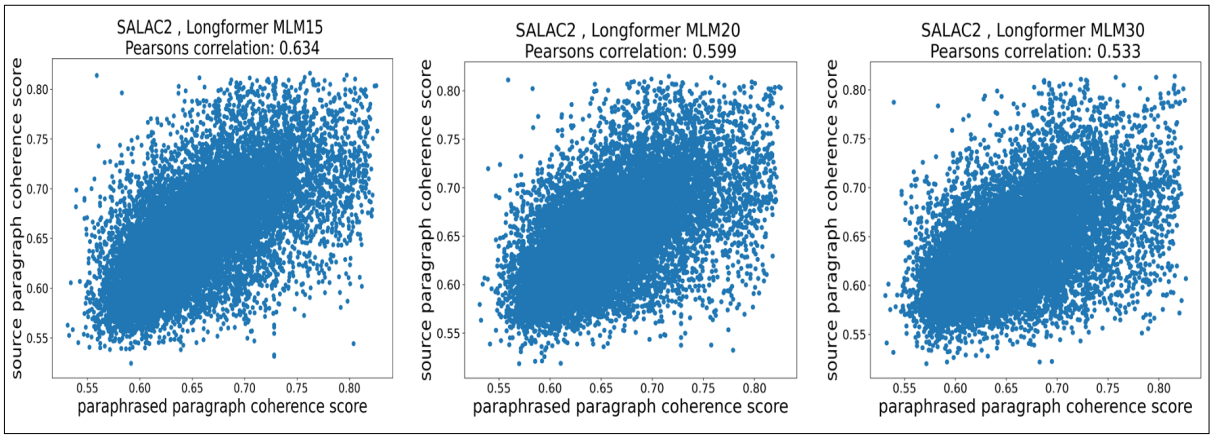

Figure 6.16 Correlation between the paraphrased paragraph generated by Longformer and the human-written (Source) paragraph seen in SALAC2 outputs

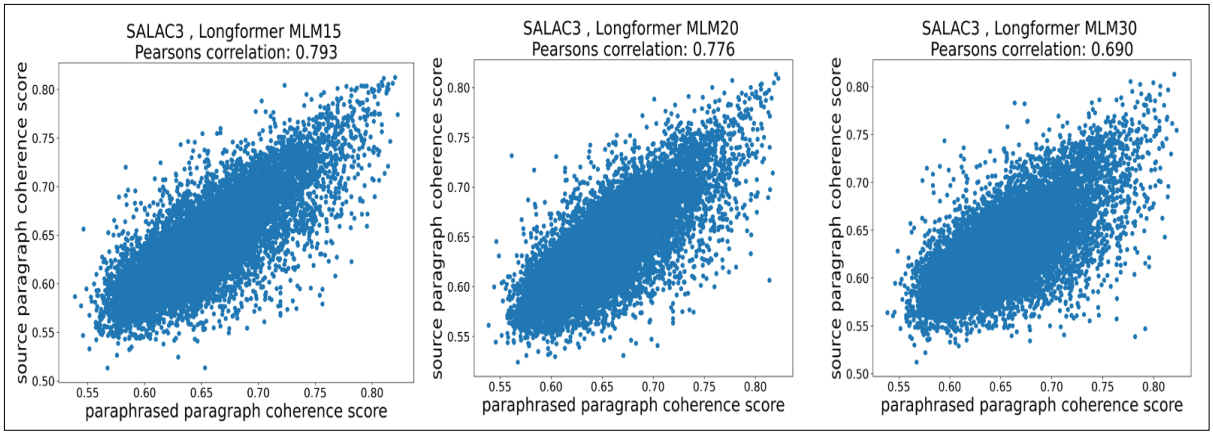

Figure 6.17 Correlation between the paraphrased paragraph generated by Longformer and the human-written (Source) paragraph seen in SALAC3 outputs

6.5. Summary

To facilitate the transition from sentence-level to paragraph-level paraphrasing, three algorithms are designed: SALAC1, SALAC2, and SALAC3. Each algorithm employs a unique approach to reorder the sentences of a paragraph based on the source inter-sentence relations. This results in paragraphs that differ in structure from the source while conveying the same semantics. Subsequently, LLMs are utilised to generate various lexically paraphrased paragraphs, taking into account the intra-sentence relations, maintaining the source meaning of the text. BERT, RoBERTa, and Longformer are employed with different levels of masking to simulate the reality of plagiarism, which entails semantically rewriting the paragraph through a range of lexical and syntactical modifications. Thus, the ALECS-SS dataset consists of 27 paraphrased versions of each source paragraph.

This chapter also addressed three of the thesis's research questions. The evaluation study demonstrated that the SALAC algorithms effectively restructure paragraphs without compromising their semantic integrity. Meanwhile, the masked approach, which paraphrases a predefined percentage of words, modifies the lexical content while preserving the source meaning. This method produced paraphrased content at the paragraph-level that is semantically consistent with the source. Longformer, in particular, excelled in capturing the overall context of paragraphs, generating more accurate paraphrased outputs in terms of maintaining the text coherence and semantics.

Although the SALAC algorithms exhibit a promising capacity for paraphrasing, additional evaluation is required to evaluate their overall efficacy. A key consideration for further investigation is the impact of the text domain on the quality of automatically generated paraphrases. Variations in domain-specific vocabulary, writing style, and contextual nuances may significantly impact the performance of algorithms. Consequently, an in-depth assessment is required to examine how these characteristics influence paraphrase quality across various domains. The next chapter will delve deeper into this issue, analysing the domain-specific problems and their ramifications for the robustness of SALAC generated paraphrases.

CHAPTER 7: MULTI-DOMAIN EVALUATION OF AUTO-PARAPHRASE GENERATION AT PARAGRAPH-LEVEL: INSIGHTS FOR EDUCATION AND PLAGIARISM DETECTION

7.1. Introduction

Paraphrase generation and identification have been extensively explored in NLP, particularly within educational applications such as automated grading, content generation, and PD. Early research primarily focused on sentence-level paraphrasing, utilizing datasets like the MSRP (Dolan & Brockett, 2005) and QQP (Puvvada et al., 2017), which facilitated advancements in ML and neural network-based PI. However, these datasets are limited in their texts which excluded multi-sentence contexts, overlooking critical aspects such as inter-sentence coherence, structural variation, and semantic fidelity. These elements are essential for educational applications and PD. According to Foltýnek et al. (2019) and Astila (2019), plagiarism at the paragraph-level is more prevalent than at the sentence, phrase, or word level, reinforcing the need for more advanced approaches.

The shift to paragraph-level paraphrasis has introduced additional complexities including sentence reordering, merging, and/or splitting while preserving both inter- and intra-sentence relationships. The emergence of Transformer-based models such as BERT (Devlin et al., 2019), RoBERTa (Y. Liu et al., 2019), and Longformer (Beltagy et al., 2020) has enabled the processing of longer text segments, making paragraph-level paraphrasing feasible. These models have demonstrated strong performance in tasks requiring deep semantic comprehension, particularly when coupled with algorithms designed to enhance structural coherence. For example,(Shen et al., 2021) demonstrated the effectiveness of the SOP task implemented in ALBERT (Lan et al., 2020) for structural refinement. Despite these advances, challenges in NLG persist. Research by (Hashimoto et al., 2019) has highlighted the limitations of automated evaluation metrics, which often fail to capture key aspects such as coherence, fluency, and semantic fidelity. Consequently, human evaluation remains indispensable, particularly in paraphrasing tasks, where maintaining semantic accuracy and structural

consistency is crucial. However, human judgment itself can be influenced by the linguistic characteristics of domain-specific texts (Schmidtova et al., 2024).

The impact of domain-specific text complexity on NLP and NLG tasks is well established, with studies showing that linguistic structures, terminology, and readability vary significantly across disciplines (Hashimoto et al., 2019; Schmidtova et al., 2024). Research on readability, domain adaptation, and linguistic diversity suggests that models trained on general-domain corpora often struggle when applied to highly specialised texts (Fei et al., 2023; Alvero et al., 2024). For instance, in machine translation the preservation of domain-specific terminology is critical. (Yasmin et al., 2022) indicated that translation in specialised domains suffer from a lack of parallel in-domain data making it difficult to maintain accuracy and fluency in technical texts. Additionally, translation models trained on general datasets often fail to capture the semantic nuances of specialised terminology leading to misinterpretations in human evaluation (Castilho & Knowles, 2024). Similarly, general-domain summarization models often produce misleading or factually inaccurate summaries when applied to domain-specific content, reducing the reliability of the generated text (Afzal et al., 2023). In paraphrase generation, it is necessary to investigate how domain-specific challenges impact the adaptation of general models across different domains.

In the field of paraphrase generation, a notable contribution to paragraph-level paraphrasis research is the ALECS-SS dataset, introduced in Chapter 6. This dataset incorporates texts from multiple domains ensuring a diverse representation of linguistic structures. By applying SALAC algorithms to manipulate text structure and Transformer-based models to refine lexical choices, Chapter 6 demonstrated that these methods effectively preserve semantic meaning while introducing structural diversity. However, the human evaluation conducted in Chapter 6 is restricted to the Psychology domain leaving unexplored how these techniques would perform across other disciplines. Given the variability in terminology, writing style, and semantic structures across domains, the question of generalizability remains open. This is particularly important in educational settings where diverse nature subject matter requires adaptable paraphrasing techniques that can accommodate a range of academic disciplines.

Despite advancements in paraphrase generation research, a gap remains unexplored in how these methods perform across domains with distinct linguistic complexities. Domains such as Anthropology and Economics, for example, incorporate specialised terminology and longer, more complex sentence structures, which may pose additional challenges for both human

evaluators and automated paraphrasing systems (Bayerl & Paul, 2011). Research in NLG (Clark et al., 2021; Artstein, 2017) underscored the impact of domain-specific characteristics on human evaluation. However, these studies have not specifically investigated paraphrase generation or the relationship between domain readability and paraphrasing quality. Readability metrics such as the Flesch Reading Ease (Kincaid, 1975) and the Gunning Fog Index (Gunning, 1952) are useful for assessing sentence complexity, word structure, and comprehension difficulty. These metrics are particularly relevant for educational applications, where clear and accessible content is essential.

This chapter addresses these gaps by exploring the effectiveness of SALAC algorithms combined with Transformer-based models in generating coherent and semantically accurate paraphrases across multiple domains, considering variations in writing styles, vocabulary, and domain-specific conventions. The ALECS-SS dataset, introduced in Chapter 6, has significantly enhanced the paragraph-level paraphrasing field produced by integrating Transformer-based models with SALAC algorithms. By emphasizing sentence reordering and semantic coherence, these methods have improved the ability to maintain contextual integrity across long text segments. However, their generalizability across diverse domains remains insufficiently examined.

Ensuring reliable paraphrase generation (Chapter 6) and identification (Chapter 8) requires evaluating the applicability of these methods across different disciplines, as academic and educational contexts frequently involve submissions from various subject areas. The ALECS-SS dataset incorporates multi-domain content and employs innovative techniques for sentence reordering and paraphrasing. Despite the promising results, human evaluation of these approaches has so far been limited to a single domain (Chapter 6), raising questions about their performance when applied to texts from different academic disciplines. A more comprehensive understanding of their effectiveness across diverse domains is essential to enhance robustness and adaptability in educational applications.

To bridge this gap, the evaluation of paraphrasing methods is extended in this chapter to multiple domains, including Anthropology, Economics, Sociology, Archaeology, and Management. Additionally, coherence and readability metrics are introduced to assess how linguistic complexity influences annotation reliability and the performance of SALAC algorithms. Accordingly, this chapter addresses the following research questions (RQ):

- **RQ6:** How does the paraphrasing quality of SALAC algorithms vary across multiple domains?
- **RQ7:** Are there domain-specific challenges in paraphrase quality as perceived by human evaluators?

This study advances the development of a reliable and generalizable paraphrasing method at the paragraph-level expanding their applicability to multi-domain NLP tasks. By evaluating the generalizability of SALAC algorithms in a multi-domain context, this chapter provides a broader validation of their effectiveness in paraphrase generation. While previous studies (Chapter 6) primarily focused on a single domain (psychology), this study extends the application of SALAC algorithms across multiple disciplines to assess their adaptability and robustness. The findings indicate that the approach successfully preserves semantic integrity and coherence across domains, although variations in performance arise due to domain-specific linguistic complexities. Through the integration of human assessment, IAA analysis, and readability metrics, this study provides deeper insights into how linguistic characteristics impact paraphrase generation and quality perception. The results further validate the applicability of SALAC algorithms in educational contexts, while also underscoring the necessity for refinements to address domain-specific challenges. Beyond enhancing paraphrase generation systems, this research contributes to the broader educational objective of promoting originality and mitigating plagiarism in academic and professional settings.

7.2. Methodology

To evaluate the effectiveness of SALAC algorithms integrated with Transformer-based models in paragraph-level paraphrasis generation, this study adopts a multi-domain approach. Particular emphasis is placed on applications in education, PD, and paraphrase quality assessment. The methodological framework remains consistent with the original study (Chapter 6), ensuring comparability of findings while extending the scope to incorporate human assessments across diverse domains. By expanding the evaluation beyond a single domain, a more comprehensive understanding of paraphrase generation performance is achieved. Human evaluators are engaged to assess the quality and coherence of generated paraphrases, allowing for insights into domain-specific variations. This approach not only validates the effectiveness of SALAC algorithms but also highlights their adaptability to varying linguistic and contextual characteristics across domains.

7.2.1. Dataset Selection

The foundation of this study is the ALECS-SS dataset, which consists of paragraph-level samples sourced from multiple domains, including Psychology, Anthropology, Economy, Archaeology, Sociology, and Management. These domains exhibit variations in linguistic complexity, structural composition, and lexical diversity, providing a robust basis for evaluating the adaptability of the proposed approach. In this chapter, the terms 'domain' and 'discipline' are used interchangeably. Notably, ALECS-SS comprises text extracted from Wikipedia, adhering to paragraph-length constraints of 50–150 words, with each sample containing 3–7 sentences. Wikipedia articles, collaboratively authored on a wide range of topics, represent a common source of content frequently plagiarised by students (Wahle, Ruas, Foltýnek, et al., 2022). Additionally, Wikipedia serves as an initial reference in instances of academic misconduct (Özşen et al., 2023) making it a highly relevant and practical resource for this research.

While the original study (Chapter 6) conducted evaluations using 100 samples from a single domain (Psychology) per algorithm, the present research extends this scope by incorporating paragraph samples from multiple domains within the dataset. This broader, multi-domain approach is essential for assessing the efficacy of paraphrasing techniques in educational contexts and for improving PD across different disciplines.

7.2.2. Paraphrase Generation

The paraphrase generation process adheres to the methodology outlined in the original study (Section 6.2). To reorder sentences within paragraphs, SALAC algorithms are employed, leveraging coherence scores derived from the SOP feature of the ALBERT model. The SALAC1 and SALAC3 are specifically considered, as they have demonstrated the most effective results, as reported in Section 6.2. These algorithms primarily reconstruct the paragraph by modifying its syntactic structure while preserving its semantic integrity. This approach is particularly crucial in plagiarism, where the challenge lies in altering the structure of a text while retaining its core meaning. It ensures that paraphrased outputs maintain semantics, making them suitable for educational content and robust PD.

7.2.3. Human Evaluation

To assess the quality of paraphrased text across multiple domains, a human evaluation study is conducted. A total of 200 paraphrased paragraphs (100 per algorithm) is randomly selected from five domains, distributed equally across them, following the approach described in Section 6.3.1. The specific topics covered within each domain are outlined in Appendix A. The domains selected for this study—anthropology, economics, archaeology, sociology, and management—were chosen to ensure coverage of a broad spectrum of the social sciences. The choice was motivated by the requirement of aligning with the intended application of ALECS-SS in educational, while also ensuring sufficient textual diversity to test the generalizability of PI methods. While the dataset cannot claim to be exhaustive of all possible domains, the balanced design provides a reasonable and transparent approximation of domain diversity within the social sciences. The sample size is determined according to the median sample size commonly used in NLG evaluations (van der Lee et al., 2021). Each paraphrased sample is scored using a 5-point Likert scale (detailed in Chapter 6), defined 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 evaluators are university students with advanced proficiency in English. Given that the dataset comprises Wikipedia text intended for a general audience, domain-specific expertise is not required for the evaluation as explained in section 7.3.1. Additionally, the ALBERT model employed in this study is not fine-tuned for domain-specific text, aligning with the objective of assessing paraphrasing techniques in their generalised form. This ensures that the findings remain relevant to applications that do not rely on domain-specific tuning, such as PD systems in educational settings.

7.3. Result and Discussion

To ensure a reliable evaluation of semantic similarity, IAA is assessed using the kappa coefficient, which is widely regarded as the standard measure for evaluating annotation reliability (Gehrmann et al., 2023). The categorisation of IAA values follows the framework

proposed by (Landis & Koch, 1977), as referenced in Section 7.3.1. Notably, (Amidei et al., 2018) suggested that an acceptable range for IAA falls between 0.3 and 0.5, with higher values being indicative of greater reliability. Additionally, Van Der Lee et al. (2019) reported that human evaluation IAA scores typically hover around 0.61, while (Amidei et al., 2018; Artstein, 2017) highlighted the challenges posed by linguistic complexity, which often results in lower IAA scores for tasks such as semantic similarity evaluations examined in this chapter.

For the Psychology domain (detailed in Section 7.3.1), 100 samples per algorithm are evaluated by six participants yielding IAA scores of 0.32. While this value is relatively low, it aligns with the inherent complexity of the task. However, a significant increase in IAA is observed when Likert scale ratings are consolidated into binary categories (Group A: 1 and 2, Group B: 3, 4, and 5), resulting in an IAA score of 0.81 (Table 7.1). Extending the evaluation to include 200 samples from five additional disciplines (Anthropology, Economics, Archaeology, Sociology, and Management) similarly demonstrated improved IAA scores when adopting binary grouping, with values ranging from 0.56 (Moderate) to 0.72 (Substantial). However, an exception is noted in the Anthropology domain, where lower agreements are likely influenced by linguistic complexity and text length.

Table 7.1 The IAA results

Domain	5-likert scale	Categorised score (A, B)
Psychology	0.32	0.81
Anthropology	0.07	0.13
Management	0.06	0.56
Sociology	0.17	0.67
Economics	0.21	0.65
Archaeology	0.18	0.72

The considerable increase in agreement following binary conversion is attributed to the simplification of evaluation criteria. Given that most evaluators assigned scores within the 3–5 range, grouping these ratings reduced variability and led to a higher level of consensus. This methodological adjustment is consistent with the subjective nature of human comprehension, where individuals may perceive semantic nuances differently. The observed trend aligns with the results presented in Table 7.4 (Section 7.3.3). Nevertheless, the Anthropology domain remained an outlier, displaying lower agreement levels. This discrepancy may suggest that the generated texts in this discipline lack sufficient variation to elicit consistent evaluations

(Celikyilmaz et al., 2021). A more detailed analysis of potential influencing factors will be conducted in subsequent sections to further examine this anomaly.

7.3.1. Statistical Analysis of Annotator Reliability

A statistical test for measuring the Anthropology evaluator's reliability is conducted by implementing the Kruskal-Wallis test which is suitable for data with ordinal scales. Ordinal data refers to a type of categorical data where the categories have a logical order or ranking, but the differences between the categories are not necessarily equal or meaningful. In this study, the scores (5 to 1) represent a ranked order of similarity, where 5 means "almost identical" and 1 means "extremely different". These scores indicate levels of similarity, but the intervals between the scores are not necessarily equal (i.e., the difference between 5 and 4 may not be the same as between 3 and 2).

The Kruskal-Wallis is used to determine whether there are significant differences between the medians of two or more evaluators. In this case, the p-value for the Anthropology domain is 0.028. Since the p-value is less than the common threshold of 0.05, this indicates a statistically significant difference between the evaluators of the Anthropology domain samples. In other words, the test suggests that at least one participant has a median value that is significantly different from the others. However, while the test indicates that differences exist, it does not specify which specific groups differ, requiring post-hoc testing (e.g., Dunn's test) to identify the exact source of the differences. Dunn's test performs pairwise comparisons between groups using ranks and corrects for multiple comparisons. This test was readjusted the alpha value taking into consideration the number of pairwise comparisons, which is 3 in this case (Equation 7.1):

$$\alpha = 0.05/3 = 0.0167 \tag{7.1}$$

The result is shown as a matrix (Figure 7.1) which indicates no significant difference between the evaluators' ratings. The significant result from the Kruskal-Wallis H-test, followed by the lack of significant findings in Dunn's post-hoc test, could be indicative of a false positive. It occurs when an initial test shows significance, but subsequent testing fails to confirm it, suggesting that the significant result is due to random chance rather than a true underlying effect (Bland & Altman, 1995).

```
      1         2          3
1 1.000000  0.95389   0.024001
2 0.953890  1.00000   0.294750
3 0.024001  0.29475   1.000000
```

Figure 7.1 Dunn's post-hoc test results

7.3.2. Text Characteristics and Readability Metrics

An analysis of text characteristics is conducted to examine their influence on annotation consistency across different domains. Variations in text length and keyword density are considered to understand their impact on evaluation reliability. The findings, presented in Table 7.2 indicate that the average text length ranged between 610 and 731 characters and between 96 and 107 words. Among the domains, Anthropology exhibited the longest texts which is likely a contributing factor to its lower IAA scores. Additionally, keyword density is assessed by calculating the number of unique words present in the source texts of each domain. Anthropology is identified as the most complex domain, containing 940 unique words, followed by Sociology (912), Management (910), Economics (884), then Archaeology (869). It is noteworthy that Psychology has the highest number of unique words due to differences in sample size, with 200 source texts compared to 40 in other domains.

Table 7.2. Text length and Keyword in each domain.

Domain	Average text length by characters	Average text length of words	Keywords
Psychology	667	100	2756
Anthropology	731	107	940
Management	671	99	910
Sociology	658	100	912
Economics	640	99	884
Archaeology	610	96	869

This finding motivated an assessment of the readability of the evaluated texts across domains. Readability refers to the ease with which a text can be read and understood, depending on its linguistic characteristics (Richards et al.,1992).To quantify readability, several commonly used readability metrics are employed (Lenzner, 2014; Crossley et al., 2011; Themistocleous, 2024; Inojosa et al., 2023), including the Flesch Reading Ease (FRE), Flesch-Kincaid Grade (FKG) level, Gunning Fog Index (GFI), Simple Measure of Gobbledygook

(SMOG) Index, Coleman-Liau Index (CLI), and Automated Readability Index (ARI). Each metric assesses text complexity based on linguistic features such as sentence length, word length, and syllable count, providing insights into the relative difficulty of understanding texts across different domains.

FRE measures the general readability of text which ranges from 0 to 100 where the lower score indicates harder to read (Kincaid, 1975), see Equation 7.2. According to Themistocleous (2024), FRE can be used in education and content creation. FKG is closely related to the FRE, this metric expresses the result as an educational grade level (Flesch, 1979), where higher scores signify greater text complexity, Equation 7.3.

$$FRE = 206.835 - 1.015 \times \left(\frac{Word\ Count}{Sentenc\ Count}\right) - 84.6 \times \left(\frac{Syllable\ Count}{Word\ Count}\right) \quad (7.2)$$

$$FKG = 0.39 \left(\frac{Word\ Count}{Sentenc\ Count}\right) + 11.8 \times \left(\frac{Syllable\ Count}{Word\ Count}\right) - 15.59 \quad (7.3)$$

Furthermore, the Gunning Fog index is calculated which takes into consideration the percentage of complex words (words with three or more syllables) and the average sentence length (Gunning, 1952). A text that receives a higher grade is more difficult to understand, Equation 7.4. While the SMOG assesses readability by analysing polysyllabic words, providing accurate results for texts suited to academic and professional contexts (McLaughlin, 1969), Equation 7.5.

$$GFI = 0.4 \left[\left(\frac{Word\ Count}{Sentenc\ Count}\right) + 100 \times \left(\frac{Complex\ Words}{Word\ Count}\right)\right] \quad (7.4)$$

$$SMOG = 1.043 \sqrt{Ploysyllabic\ Words\ count \times \frac{30}{Sentence\ Count}} + 3.1291 \quad (7.5)$$

Additionally, the CLI use characters per word and sentence length instead of syllables, facilitating computational simplicity (Coleman & Liau, 1975), Equation 7.8. Finally, the ARI assesses readability by analysing characters per word and words per sentence, yielding a grade-level score (Smith & Senter, 1967), Equation 7.9. These metrics collectively offer a thorough evaluation of text comprehensibility for specific audiences or the public.

$$L = \left(\frac{Character\ Count}{Word\ Count}\right) \times 100 \quad (7.6)$$

$$S = \left(\frac{Sentece\ Count}{Word\ Count}\right) \times 100 \quad (7.7)$$

$$CLI = 0.0588 \times L - 0.296 \times S - 15.8 \quad (7.8)$$

$$ARI = 4.71\left(\frac{Character\ Count}{Word\ Count}\right) + 0.5\left(\frac{Word\ Count}{Sentece\ Count}\right) - 21.4 \quad (7.9)$$

Table 6.3 shows the results of implementing the readability metrics on the text samples involved in the human evolution from each considered discipline. For FSE, lower scores signify greater difficulty, whereas for FKG, GFI, SMOG, CLI, and ARI, higher scores indicate texts that require a higher grade level for comprehension. Firstly, the notable closeness in measures between Psychology-1 and Psychology-2 in Table 7.3 is attributable to their common domain, as both samples originate from psychology-related texts. Metrics show equal scores such as in FSE or exhibit minor variations of 0.3 or less (e.g., CLI scores of 14.5 versus 14.4), indicating the consistent linguistic structures, vocabulary, and writing styles characteristic of the same domain. These results underscore the efficacy of these metrics in identifying and quantifying nuanced distinctions, while also demonstrating their capacity to differentiate texts more effectively across diverse domains characterised by significant linguistic variance.

Regarding other domains in Table 7.3, Anthropology exhibits the lowest FSE score (30.3), indicating that it is the most challenging to understand based on this criterion. In contrast, Archaeology achieves the greatest FSE score (48.9), signifying that it is the most comprehensible domain within the given categories. The discipline of Economics shows a notably high FSE score of 44.6, indicating its relative ease of readability. Psychology, Sociology, and Management have moderate FSE scores between 35.9 and 41.2, indicating texts of intermediate readability within the reported domains. The FKG results further underscore these insights. Archaeology receives the lowest FKG score (11.3), indicating that the material requires comprehension at an approximate 11th-grade reading level. Inversely, Anthropology demonstrates the highest FKG score (14.4), signifying that readers require a comprehension level of 14 or above. Psychology, Sociology, and Management demonstrate mean scores of roughly 13, signifying intermediate difficulty.

A similar pattern appears in the GFI, with Anthropology achieving the highest score (15.7), thus confirming its status as the most complex domain. On the contrary, Archaeology yields the lowest GFI score (12.9), indicating its relative simplicity. The SMOG scores support this trend, identifying Anthropology as the most difficult (15.7) and Archaeology as the least difficult (13.0). Both indices underscore the linguistic complexity of texts in Anthropology relative to other disciplines. The CLI and ARI further verify these conclusions. In the CLI, Archaeology receives the lowest score (12.8), whilst Anthropology has the highest grade (15). Simultaneously, ARI categorises Archaeology as the least complex domain (13.6) and

Anthropology as the most complex (16.5). The consistent results across multiple criteria underscore the increased complexity of texts in Anthropology, which may have affected the evaluators' level of agreement, as illustrated in Table 7.1.

Table 7.3. The readability scores for each domain are presented (**Bold** = Most Complex, underlined = Easiest), with lower scores indicating easier readability and higher scores indicating greater difficulty, except for the FSE, where the reverse is true.

Readability Metric	Psychology 1	Psychology 2	Anthropology	Management	Sociology	Economics	Archaeology
FSE	36.7	36.7	**30.3**	35.9	41.2	44.6	48.9
FKG	12.9	13.1	**14.4**	13.3	12.7	12.2	11.3
GFI	14.7	14.8	**15.7**	15.1	14.4	14.5	12.9
SMOG	14.8	15.1	**15.7**	15.2	14.3	14.2	13.0
CLI	14.5	14.4	15.0	**15.1**	14.2	13.5	12.8
ARI	14.9	15.2	**16.5**	15.9	15.3	14.7	13.6

The analysis identifies distinct patterns in readability IAA across different domains. Anthropology texts typically have higher difficulty levels as indicated by their readability scores classifying them among the most complicated materials in this study. This observation corresponds with the comparatively low IAA of 0.13 (as seen in Table 7.1) implying that the complexity of texts in this domain may contribute to increased variability in human interpretation. This observation aligns with prior research indicating that more complex domains pose greater challenges in maintaining consistent evaluations (Biber et al., 1998; Bayerl & Paul, 2011). Conversely, Archaeology showing lower readability scores in the above table leads to characterised by a higher IAA of 0.72, signifying that its texts are more comprehensible and yield consistent interpretations among evaluators. Other fields, including Management, Economics and Sociology, have a moderate level of complexity regarding readability, resulting in intermediate IAA scores of 0.56, 0.65 and 0.67, respectively. This suggests that texts in these fields strike a balance between linguistic complexity and interpretability.

In summary, Tables 7.1 and 7.3 demonstrate an inverse relationship between text readability and semantic similarity assessment, a consistently observed trend across all domains examined in this study. Even though earlier studies in psychology have found conflicting results (Chapter 6), the general trend is still strong across the domains discussed in this investigation.

7.3.3. SALAC Algorithms' Performance Across Domains

To expand upon the observations made in Chapter 6, where text is extracted from a single domain, the analysis in this study is extended to five distinct domains. The primary objective is to evaluate the effectiveness of the proposed SALAC algorithms across multiple domains. Similar to the previous approach, 300 scores are assigned by evaluators for each algorithm, ensuring consistency in evaluation. The distribution of human-assigned scores across domains is observed to reveal notable patterns regarding the performance of SALAC1 and SALAC3 (Table 6.4).

In Psychology, the majority of ratings are concentrated in scores 4 and 5, which indicate a high degree of similarity to the source text. SALAC1 received 24% at score 4 (very similar) and 39% at score 5 (almost identical), while SALAC3 achieved 23% and 40%, respectively. This distribution indicates that a strong perception of semantic similarity is maintained in this domain. A similar pattern is observed in Archaeology, where 30% of ratings for SALAC1 are assigned to score 4, while SALAC3 received a substantial 41% at score 4, confirming that SALAC1 and SALAC3 alter paragraph structure while preserving semantic meaning. In Economics, 48% of ratings at score 5 are assigned to both SALAC1 and SALAC3, indicating a strong consensus on paraphrase quality within this field. Given that a score of 5 represents an almost identical meaning, these results suggest that the paraphrased texts remain highly faithful to the source. Similar trends are identified in Sociology and Management, where a greater concentration of ratings is observed at scores 3 (similar with major changes in meaning), 4, and 5. In contrast, Anthropology displays a broader distribution of ratings, with a significant portion of responses falling into score 2 (30% for both SALAC1 and SALAC3), which indicates dissimilarity with significant changes in meaning. This variability is attributed to the complexity of the language used in this domain, as discussed in a previous section.

A key observation is the consistent trend across all domains, where both SALAC1 and SALAC3 receive a higher proportion of ratings in Group B (scores 3, 4, and 5) compared to Group A (scores 1 and 2). In Psychology, 90% of the ratings for SALAC1 fall within Group B, while SALAC3 receives 84%. Archaeology displays a similar pattern, with 89% of SALAC1's ratings and 83% of SALAC2's ratings categorised in Group B. Even in Anthropology, which exhibits the most variability, the proportion of ratings in Group B remains higher than in Group A, at 62% for SALAC1 and 63% for SALAC2. This trend reinforces the

overall effectiveness of the SALAC algorithms in maintaining semantic similarity across different domains. These results emphasise that, despite variations in domain complexity, both algorithms perform effectively across all domains. The consistently higher proportion of ratings in Group B suggests that human evaluators generally perceive the paraphrases as semantically similar, with relatively few cases classified as dissimilar (Group A). The high percentages of scores 4 and 5 further confirm that the majority of paraphrased paragraphs retain the core meaning of the source text. This further supports the adaptability of the SALAC algorithms in paraphrase generation across diverse academic disciplines while also highlighting the importance of domain-specific factors that may influence the IAA of evaluating paraphrased text.

Table 7.4. Distribution of human-assigned similarity scores for SALAC1 and SALAC3 across domains. Scores range from 1 (extreme dissimilarity) to 5 (almost identical). Group A (1–2) represents low similarity, while Group B (3–5) represents semantically consistent paraphrases.

	Score	**1**	**2**	**3**	**4**	**5**	**A**	**B**
Psychology	**SALAC1**	1%	9%	27%	24%	39%	10%	90%
	SALAC3	3%	13%	21%	23%	40%	16%	84%
Anthropology	**SALAC1**	8%	30%	22%	17%	23%	38%	62%
	SALAC3	7%	30%	25%	20%	18%	37%	63 %
Economics	**SALAC1**	14%	8%	15%	15%	48%	22%	78%
	SALAC3	8%	7%	15%	22%	48%	15%	85%
Archaeology	**SALAC1**	3%	8%	32%	30%	27%	11%	89%
	SALAC3	5%	12%	23%	41%	19%	17%	83%
Sociology	**SALAC1**	3%	9 %	20%	33%	35%	12%	88%
	SALAC3	5%	15%	20%	30%	30%	20%	80%
Management	**SALAC1**	12%	8%	28%	27%	25%	20%	80%
	SALAC3	8%	17%	22%	33%	20%	25%	75%
All Domain	**SALAC1**	7%	12%	24%	24%	33%	19%	81%
	SALAC3	6%	16%	21%	28%	29%	22%	78%

While there are instances where human evaluators perceive the paraphrased paragraphs as dissimilar in meaning to the source for both algorithms, the majority of scoring aligns with the confirmation of semantic similarity. To highlight the differences in efficiency between

SALAC1 and SALAC3, two columns are inserted to the right of Table 7.4, categorising the scores based on their definition into two groups (A = 1,2 and B = 3,4,5). The results indicate that the performance of both algorithms remains close across all domains, with the greatest variation observed at 7% in Economics and Sociology, followed by 6% in Psychology, 5% in Archaeology and Management, and the lowest variation in Anthropology at 2%. However, when considering all domains collectively, the overall variation remains low, represented by 2%. These findings further confirm the effectiveness of the SALAC algorithm in producing paraphrased paragraphs that restructure content while preserving semantic meaning across various domains.

7.3.4. Correlation Between Human and Automated Coherence Scores Across Domains

The correlation between the coherence score assigned by human evaluators and the automatically generated coherence score is investigated across two collections, as presented in Table 7.5. The Pearson correlation coefficient is employed to quantify the strength and direction of the relationship between these two variables, producing values ranging between -1 and 1. The results of both algorithms are compared by treating the samples retrieved from the Psychology collection as an independent category, while the remaining samples from the previously mentioned domains constitute a separate collection. This approach aims to examine whether domain diversity affects the correlation between human ratings and automatic coherence scores. The number of samples in both collections is equal, with each comprising 200 instances.

The findings indicate that the number of sentences in a paragraph influences the correlation in both collections. This effect is demonstrated by normalising the coherence score of a paragraph by its sentence count. In the Psychology collection, the correlation for SALAC1 and SALAC3 increases from 0.05 and 0.16 to 0.23, respectively. Similarly, in the multi-domain collection, SALAC1 and SALAC3 show an increase from 0.14 and 0.06 to 0.15 and 0.16, respectively. The strongest correlation is observed when considering the lowest coherence scores between sentences in both collections. This finding aligns with the primary objective of SALAC1 and SALAC3, which prioritise reordering paragraph sentences based on the highest coherence scores, leading to minimising the occurrence of low-coherence sentence pairings.

Table 7.5. The Pearson correlation between the coherence score assigned by humans and the automatically generated coherence score for one domain (Psychology) and Multi-domain collections. The best correlation is highlighted in bold font.

	Pearson correlation			
Collection	**Psychology**		**Multi-domains**	
Algorithm	**SALAC1**	**SALAC3**	**SALAC1**	**SALAC3**
5-point Likert scale	0.05	0.16	0.14	0.06
Coherence score/ number of sentences	0.23	0.23	0.15	0.16
Minimum coherence score between sentences	**0.80**	**0.63**	**0.51**	**0.67**
Maximum coherence score between sentences	0.46	0.51	0.22	0.20

These results confirm that the presence of even a single low-coherence sentence pair significantly influences the paraphrased paragraph semantics, which is observed by human evaluators' ratings. The correlation between human-assigned and automatically generated coherence scores is evident across both collections, highlighting a consistent pattern in the performance of the algorithms. This consistency suggests that the proposed methods remain effective despite variations in text domains. The observed discrepancies between the two collections may be attributed to the linguistic characteristics of each domain, which likely influenced the evaluation process, as discussed in previous sections.

7.4. Summary

This chapter provides a comprehensive evaluation of SALAC algorithms and Transformer-based models for paraphrasing across multiple domains, addressing the limitations of prior work that focused exclusively on psychology (Chapter 6). By incorporating texts from Anthropology, Economics, Archaeology, Sociology, and Management, the research highlights the generalizability of these methods while uncovering domain-specific challenges. The research questions are addressed through findings that demonstrate the effectiveness and adaptability of SALAC algorithms across multiple domains, despite challenges introduced by domain-specific linguistic complexity. The results indicate that SALAC algorithms exhibit effective performance in preserving the semantic meaning of paraphrased text while altering its structure across all domains, demonstrating consistent performance across domains. However, certain domains, such as Anthropology, present unique challenges due to their higher

lexical density and linguistic complexity, which are reflected in lower AA scores and higher readability metrics.

The evaluation methodology developed in this study integrates human annotation consistency, statistical analysis, and readability metrics, providing a robust framework for assessing paraphrasing techniques in multi-domain contexts. The use of IAA and binary scoring systems proved essential for ensuring reliable evaluations, especially in linguistically complex domains. Readability analysis further demonstrated the significant influence of text characteristics on annotator agreement and coherence evaluations, offering new insights into the interaction between linguistic complexity and paraphrasing performance.

Despite the strong performance of SALAC algorithms, the findings emphasise the need for domain-specific optimisations to handle specialised texts effectively. While the algorithms performed well overall, the linguistic complexity inherent in fields such as Anthropology impacted both human evaluations and algorithmic outcomes. These results suggest that further refinement of paraphrasing methods is necessary to address the challenges posed by highly specialised domains.

Future research can build on this work by exploring the potential of fine-tuning LLMs in domain-specific text to complement SALAC algorithms, particularly for domains with unique linguistic requirements. By advancing the evaluation of paraphrase generation techniques in multi-domain contexts, this study contributes to the development of more robust and adaptable solutions for real-world applications.

To advance the progress of the work accomplished in this chapter, it is essential to evaluate the effectiveness of DL models in distinguishing between machine-paraphrased text and human-written text at the paragraph-level, particularly given these models' ability to handle longer texts. This evaluation and comparative analysis will be explored in the following chapter.

CHAPTER 8: A COMPARATIVE STUDY ON IDENTIFYING HUMAN-WRITTEN VS. MACHINE-PARAPHRASED AT PARAGRAPH-LEVEL

8.1. Introduction

PI as a component of PD, has been significantly expanded with the advancements in LLMs, which are capable of processing lengthy texts and paraphrasing sentences, paragraphs, and even long documents efficiently (J. Zhou & Bhat, 2021). Consequently, PI no longer focuses solely on comparing pairs of texts; it now includes distinguishing between human-written content and machine-paraphrased text.

In this chapter, state-of-the-art pre-trained models are employed as detection algorithms for identifying human-written and auto-paraphrased content at the paragraph-level. These paragraphs are paraphrased using SALAC algorithms and MLMs, (Wahle, Ruas, Kirstein, et al., 2022) demonstrating that LLMs can rewrite text in ways that are difficult for humans to recognise as machine-paraphrased text. Thus, two research questions are investigated in this chapter which are

- ***RQ8:*** *How effectively can autoencoding models discriminate between the source (human-written) and machine-paraphrased text generated by the paragraph-level method, without requiring a direct comparison between the two?*
- ***RQ9:*** *How effectively can state-of-the-art autoregressive models discriminate between the source (human-written) and machine-paraphrased text generated by the paragraph-level method, without requiring a direct comparison between the two?*

Although few studies have been conducted on identifying machine-paraphrased text (Wahle et al., 2021; Becker et al., 2023), these experiments focus on sentence-level

paraphrases. In this chapter, the efficiency of pre-trained models in distinguishing human-written and auto-paraphrased at paragraph-level is investigated.

8.2. State-of-the-Art Prior Research

Recent research on detecting machine-generated text has gained considerable attention caused by advancements in artificial intelligence, particularly the development of large pre-trained language models, as discussed in Chapter 3. Similarly, the concentration on detecting machine-paraphrased text has been raised, although the majority of existing studies have been focused on sentence-level paraphrasis, even when addressing paragraph-length texts.

This section discusses prior works that align with the study in this chapter. Early investigations focused on detecting paraphrased text generated by online tools like SpinBot, which paraphrased Wikipedia paragraphs. Then, six word-embedding models are used to extract features from the text, and these features are input into five ML classifiers (Foltýnek et al., 2020). To expand on this work, research papers from arXiv and theses by English language learners are extracted and paraphrased using tools such as SpinnerChief-DF and SpinnerChief-IF. The key difference between these tools is the percentage of content's words paraphrased _ SpinnerChief-DF altered 12.58% of the text, while SpinnerChief-IF changed 19.37%. Then, text-matching software and eight LLMs are employed to detect paraphrasing text (Wahle, Ruas, Foltýnek, et al., 2022).

Following these studies, Wahle et al. developed a dataset in which LLMs are implemented to paraphrase each sentence within a paragraph, with a 15% mask probability. Three LLMs are then evaluated for their ability to detect sentence-level machine-paraphrased text (Wahle et al., 2021). Additionally, human performance is compared with the LLMs efficiency in distinguishing between human-written and machine-paraphrased text that was generated using the SpinnerChief tool (50%), BERT (15%), and GPT. They concluded that humans had difficulty identifying auto-paraphrased text of the source. In particular, the ACC of detecting paraphrases generated by autoencoding models varied between 48.28 % and 84.48% (Wahle, Ruas, Kirstein, et al., 2022)

This chapter focuses on evaluating the effectiveness of autoencoding and autoregressive LLMs in detecting paragraph-level paraphrases, setting it apart from previously discussed works that primarily concentrate on sentence-level paraphrasis. However, the findings of these

earlier studies are taken into consideration, particularly those that emphasised the ability of LLMs such as BERT, RoBERTa, and Longformer to generate paraphrased text that is challenging to differentiate from human-written content. Furthermore, the use of a low masking probability to paraphrase a smaller percentage of words is taken into account, as previous studies have demonstrated that text paraphrased by SpinnerChief-IF, which alters approximately 19% of the words, is more easily distinguishable than text paraphrased by SpinnerChief-DF, 12.58%.

8.3. Methodology

8.3.1. Dataset

In this chapter, the task of discriminating between human-written and auto-paraphrased text at paragraph-level is addressed, which begins with considering the ALECS-SS dataset. The ALECS-SS dataset is developed to address the lack of available paragraph-level paraphrases datasets, and then categorised into nine subsets (see section 8.4.1). It is primarily designed to assess the effectiveness of pre-trained models in detecting different levels of machine-generated paraphrased paragraphs. To facilitate this, sentences within the paragraphs are rearranged, then 15%, 20%, or 30% of the tokens are paraphrased. These percentages are specifically chosen based on research indicating that most online paraphrasing tools paraphrase approximately 15%-19 % of the input text (Wahle, Ruas, Foltýnek, et al., 2022). Additionally, the dataset is enhanced by incorporating paraphrases at 20% and 30%, further increasing its diversity by representing the same content in varied textual formats. This expanded ALECS-SS, serves as an excellent resource for training LLMs to comprehend and capture diverse language patterns and variations.

8.3.2. Classification Algorithms

In the evaluation of PI, several state-of-the-art pre-trained models are examined, utilising their default hyperparameter settings. These models include BERT, RoBERTa, Longformer, GPT-3, GPT-3.5, and GPT-4. The models previously employed for paraphrasing are considered as detection models, following the findings of Zellers et al. (2019) and Wahle et al. (2021), who observed that the most effective model for detecting automatically generated text is often the same model used to generate it. While their research focused on detecting fake news and

sentence-level paraphrasing, this study is extended to paragraph-level paraphrasing. Furthermore, state-of-the-art generative models are analysed to investigate their classification capabilities and compare their performance to the models previously used for paraphrasing.

When employing encoding models such as BERT, RoBERTa, and Longformer for classification tasks, modifications to their architecture are often necessary to transition from language encoding to classification. These models, initially designed to generate token embeddings and contextual representations, output hidden states for each token in an input sequence. For classification, a common approach is to extract the representation of the [CLS] token, which serves as a summary of the input sequence. This [CLS] representation is then passed through a fully connected (dense) layer, acting as a classification head that maps the representation to the target output classes (Figure 8.1). Additionally, a dropout layer may be introduced to mitigate overfitting and improve generalisation. The final layer typically applies a softmax activation function for multi-class classification or a sigmoid activation for binary classification, converting the dense layer's outputs into probability distributions for class prediction. Alternatively, instead of relying solely on the [CLS] token, an average (mean pooling) of all token embeddings can be computed across the sequence, allowing the model to capture a more comprehensive representation that incorporates information from all tokens, potentially improving classification performance.

The implementation of GPT models for classification tasks requires careful prompt engineering to ensure the task requirements are met. As outlined in the OpenAI documentation, the structure and clarity of the prompt significantly impact the model's performance. Thus, constructing a well-formulated prompt is crucial for effectively utilising GPT models in various tasks. Thus, numerous studies have been conducted to investigate the influence of prompt engineering on the results of NLP downstream tasks (as discussed in Chapter 3). However, the majority of these studies have focused primarily on auto-generated content or paraphrased text at the sentence-level. In contrast, this study moves beyond sentence-level paraphrasis, concentrating on distinguishing between human-written and machine-paraphrased text at the paragraph-level. This shift enables a more comprehensive exploration of how GPT models process complex text structures and variations, offering new insights into their capacity to identify nuanced paraphrase patterns across larger textual segments.

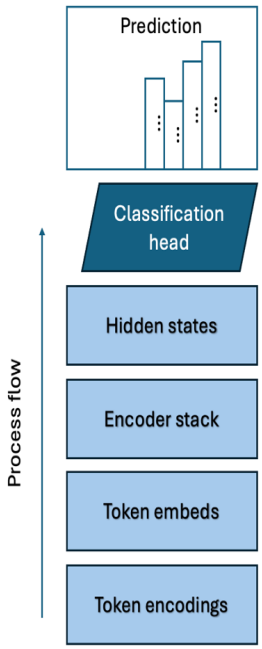

Figure 8.1The architecture employed for sequence classification utilizes an encoder-based transformer.

In the field of classification, there are three primary strategies used to guide a model's response: zero-shot, few-shot, and chain prompting. Zero-shot prompting involves assigning tasks, such as concept or class recognition, that the model has not encountered during training. In this approach, the model makes predictions based on auxiliary information, such as descriptions, relationships between known and unknown classes, or semantic embeddings, without the use of labelled samples (Kojima et al., 2022). This method proves particularly useful for scenarios where labelled data is scarce or unavailable.

In contrast, few-shot prompting aims to improve the model's efficiency by reducing the need for extensive labelled data during training. By leveraging the model's prior knowledge or applying meta-learning techniques, the model can quickly adapt to new tasks or classes with only a minimal amount of labelled data (Reynolds & McDonell, 2021). This approach facilitates quicker and more resource-efficient training, making it especially beneficial in domains where data is limited.

Furthermore, chain prompting involves breaking down a complex task into a sequence of smaller prompts. Each prompt addresses a specific subtask, and the output from one prompt serves as the input for the next, creating a step-by-step process. This structured approach

ensures that the model generates focused and coherent responses, making it particularly effective for multi-step problems requiring logical progression (Shao et al., 2023).

In Section 8.4.2, the results of applying both zero-shot and few-shot approaches to the ALECS-SS dataset are presented. The findings offer valuable insights into the performance of the models in detecting various levels of paraphrasing, highlighting their effectiveness in this classification task.

8.4. Results and Discussion

8.4.1. Efficiency of Autoencoder LLMs in Classification

To answer RQ8, the performance of three autoencoding classification models in NLP is evaluated: BERT, RoBERTa, and Longformer. This evaluation is carried out using the ALECS-SS subsets, in which each main category is paraphrased by an LLM with varying levels of MLM set at 15%, 20%, and 30%. As a result, the ALECS-SS dataset is composed of Human-written paragraphs and nine paraphrased subsets as in Table 8.1.

Table 8.1 ALECS-SS dataset subsets used for paraphrase identification (PI). Paragraphs are paraphrased using BERT, RoBERTa, and Longformer with masked language modelling (MLM) at 15%, 20%, and 30%, representing the proportion of text paraphrased.

Subset	Model	MLM Level
1	BERT	15%
2		20%
3		30%
4	RoBERTa	15%
5		20%
6		30%
7	Longformer	15%
8		20%
9		30%

In addition to considering paragraphs, each sentence is paraphrased independently to evaluate the performance of LLMs in detecting machine-paraphrased sentences. This approach is particularly relevant as most prior research has primarily concentrated on assessing the performance of LLMs on sentence-level paraphrased text within the two datasets previously discussed, namely MSRP and QQP.

Table 7.2 provides a comprehensive comparison of the three MLM, namely: BERT, RoBERTa, and Longformer, evaluated across the different subsets of the ALECS-SS dataset. These subsets are identified based on the model used for paraphrasing (ALECS-SS(BERT), ALECS-SS(RoBERTa), and ALECS-SS(Longformer)) and the varying levels of mask probabilities: MLM 15, MLM 20, and MLM 30. This table presents the F1-micro scores for each model across the subsets, along with their average (AVE) performance across all subsets within each category. The F1-micro score aggregates the counts of true positives, false positives, and false negatives across all classes before calculating precision and recall, ensuring that equal weight is given to each instance regardless of its class affiliation. This framework facilitates an overall perspective on model performance instead of concentrating on specific classes. The results underscore the differences in efficiency among the models, revealing significant variations in their classification capabilities.

In the first section of Table 8.2, which focuses on paragraphs paraphrased through the use of BERT, it is observed that RoBERTa achieves the highest overall performance, with an average score of 93.2. This result indicates that RoBERTa surpasses both BERT, which attained an average of 92.5, and Longformer, which scored 91.3, in terms of effectiveness. Upon further analysis of the individual subsets, RoBERTa's performance in recognising human-written and paraphrased paragraphs at MLM 15, MLM 20, and MLM 30 is reflected in its F1 micro scores of 87.0, 95.4, and 97.2, respectively. RoBERTa's superior performance can be largely attributed to its advanced architecture, which represents an enhanced evolution of BERT to handle long text, providing it with a competitive advantage in this context. BERT's performance also remains notably strong, especially in the MLM 20 and MLM 30 subsets, although it lags slightly behind RoBERTa by a small margin. It could be suggested that this is because BERT is utilised to generate the paraphrased text within these particular subsets. Although Longformer achieved remarkable results, its performance is lower than that of BERT and RoBERTa.

In the analysis of the ALECS-SS(RoBERTa) subset, it is observed that RoBERTa attains the leading average score of 97.5, closely followed by Longformer with 97.3, while BERT lags significantly with a lower score of 95.2. RoBERTa has outstanding performance across the subsets, achieving remarkable scores of 96.2, 97.9, and 98.4 in identifying segments paraphrased at MLM 15, MLM 20, and MLM 30 levels, respectively. In the MLM 15 subset, Longformer exhibits performance that is comparable to that of RoBERTa. However, it lags

slightly behind in the MLM 20 and MLM 30 subsets. Compared to RoBERTa and Longformer, BERT performs adequately but falls short in the MLM 15 subset due to its lower fraction of paraphrased tokens. These findings indicate that both RoBERTa and Longformer demonstrate strong capabilities in processing lengthy documents, effectively distinguishing between human-written and machine-paraphrased paragraphs by identifying subtle lexical and syntactic variations.

In the ALECS-SS (Longformer) category, Longformer achieves the highest performance, with an average score of 94.5, exceeding RoBERTa's 92.3 and BERT's 89.6. In detail, Longformer achieves F1-micro scores of 90.0, 95.4, and 98.3 in the MLM 15, MLM 20, and MLM 30 sets, respectively, demonstrating its robust capacity to handle lengthy contexts and extensive sequences, particularly when paraphrasing is generated by Longformer itself. In a similar vein, RoBERTa has solid performance, particularly in the MLM 20 and MLM 30 subgroups, although it does not achieve the highest scores obtained by Longformer. In contrast, BERT particularly struggles to detect paraphrased paragraphs with a low percentage of paraphrased tokens.

In summary, the results indicate that RoBERTa consistently outperforms in distinguishing between paraphrased and human-written paragraphs across two ALECS-SS categories that are generated using BERT and RoBERTa. This confirms the effectiveness of RoBERTa, as mentioned in other work (Wahle et al., 2021). Furthermore, as noted by Zellers et al., the effectiveness of a classifier is significantly influenced by the language model utilised for generating the paraphrased text (Zellers et al., 2019). Similarly, Longformer demonstrates exceptional capability in managing long contexts, surpassing other models in detecting paraphrased paragraphs generated by Longformer itself and nearly matching RoBERTa's performance on datasets paraphrased using the RoBERTa model. This is attributed to the fact that Longformer is primarily built on the RoBERTa checkpoint, with additional training on longer documents and the use of different attention mechanisms. For BERT, it lags behind both RoBERTa and Longformer in performance; nonetheless, it achieves comparable results in identifying paragraphs paraphrased with BERT.

Table 8.2 F1-score of implementing detection algorithms on the different subsets of ALECS-SS

Paraphrasing model	ALECS-SS (BERT)			
Dataset /classifier	**MLM 15**	**MLM 20**	**MLM 30**	**AVE.**
BERT	86.2	94.4	97.1	92.5
RoBERTa	87.0	95.4	97.2	**93.2**
Longformer	84.2	93.4	96.4	91.3
Paraphrasing model	**ALECS-SS (RoBERTa)**			
Dataset /classifier	**MLM 15**	**MLM 20**	**MLM 30**	**AVE.**
BERT	91.5	96.3	97.8	95.2
RoBERTa	96.2	97.9	98.4	**97.5**
Longformer	96.2	97.7	98.2	97.3
Paraphrasing model	**ALECS-SS (Longformer)**			
Dataset /classifier	**MLM 15**	**MLM 20**	**MLM 30**	**AVE.**
BERT	83.3	89.4	96.2	89.6
RoBERTa	87.1	93.1	96.9	92.3
Longformer	90.0	95.4	98.3	**94.5**

A more detailed analysis is conducted to assess the performance across all categories, with particular attention to subsets that had been paraphrased by other models (see Table 8.3). In general, RoBERTa and Longformer show comparable performance, highlighting their similar effectiveness in processing longer texts. However, Longformer performed better than the others in identifying both human-written and machine-paraphrased text produced by other models (ALECS-SS (BERT) and ALECS-SS (RoBERTa)), with RoBERTa following closely in categories (ALECS-SS (BERT) and ALECS-SS (Longformer)) and BERT ranking last in performance on (ALECS-SS (RoBERTa) and ALECS-SS (Longformer) subsets.

Table 8.3 Presents the results, including the **AVE** values from Table 8.2. **AVE*** refers to the average of all results obtained by the classifier. **AVE**** represents the average of results achieved by the classifier when applied to ALECS-SS subsets that have been paraphrased using other models.

Classifier	ALECS-SS (BERT)	ALECS-SS (RoBERTa)	ALECS-SS (Longformer)	AVE*	AVE**
BERT	92.5	95.2	89.6	92.43	92.4
RoBERTa	**93.2**	**97.5**	92.3	94.33	92.75
Longformer	91.3	97.3	**94.5**	94.36	94.3

To align with previous research, a comparison is made between the results of the current study and earlier studies that utilised a dataset sourced from Wikipedia and paraphrased using

MLM15 with Longformer (Wahle et al., 2021). The primary methodological difference lies in their approach of paraphrasing each sentence independently, focusing solely on intra-sentence relations, which led to lexical differences between the paraphrased paragraphs and the source text. In contrast, the ALECS-SS dataset features paraphrased paragraphs that differ both lexically and syntactically from the source text, as it considers both intra-sentence and inter-sentence relations during the paraphrasing process. In this comparison, both datasets are paraphrased using Longformer with a 15% mask probability. Table 8.4 shows other work results to this study, focusing on the performance of BERT, RoBERTa, and Longformer in their ability to differentiate between human-written text and text paraphrased through LLMs. Performance metrics, assessed using the F1-micro score, clearly indicate that the present study outperformed previous research, with all classifiers showing improved results. In particular, the BERT classifier in this study recorded an impressive F1-micro score of 83.4, significantly surpassing the 69.4 scores reported in prior studies. Likewise, both the RoBERTa and Longformer classifiers demonstrated superior performance, achieving F1-micro scores of 88.3 and 90.1, respectively, which exceed the 82.1 and 86.0 results reported in earlier research. These improvements emphasise the effectiveness of LLMs in discriminating between machine-paraphrased paragraphs and human-written paragraphs without relying on information from the source text. Although changes in paragraph structure can pose challenges for classifiers that primarily rely on lexical variation, the modifications applied at both the syntactic and lexical levels, while preserving the overall meaning, unexpectedly enhanced the performance of LLMs to detect paraphrased paragraphs. Additionally, this finding is particularly relevant to real-world plagiarism practices, where plagiarists often preserve the overall paragraph structure rather than merely rewording individual phrases or sentences (Foltýnek et al., 2019).

Table 8.4 F1-score comparison on the PI task: this study (ALECS-SS, paragraph-level) vs. Wahle et al. (2021, sentence-level).

Classifier	Current study	Wahle et al. 2021
BERT	83.4	69.4
RoBERTa	88.3	82.1
Longformer	90.1	86.0

The performance of LLMs on sentence-level paraphrasing and sentence length is subjected to further investigation (Table 7.5). To carry out this analysis, sentences within the paragraphs of the ALECS-SS dataset are separated to emulate the sentence-level paraphrasis seen in the

MSRP and QQP datasets (outlined in Chapter 3). Subsequently, a comparative analysis is performed to evaluate the performance of classifiers, namely BERT, RoBERTa, and Longformer, across various datasets. The variations in classifiers' performance are underscored by the F1-micro score percentages, which are used as the primary performance metric to illustrate how effectively these models functioned when applied to distinct text datasets.

Table 8.5 Results on paraphrase identification (PI) for sentence-level paraphrases datasets (F1-score). *Data from (Devlin et al. 2019), **Data from (Y. Liu et al. 2019)

Classifier/ Dataset	ALECS-SS	MSRP	QQP
BERT	65.1	89.3*	72.1*
RoBERTa	65.6	91.2**	73.0**
Longformer	67.5	-	-

The result in Table 8.5 shows that a notable decline in performance scores is observed across all three models when evaluated on the sentence-level paraphrases of ALECS-SS dataset, attributed to the difficulty posed by the need to identify machine-paraphrased paragraphs without access to the source text for comparison. By contrast, in the experiments carried out using the MSRP and QQP datasets, the classifiers are simultaneously provided with both the source and the paraphrased text, facilitating the identification of paraphrases and yielding improved performance. The additional observation revealed that LLMs exhibited superior performance with paragraph-length texts in comparison to sentence-length texts. This is attributed to the fact that longer texts offer more comprehensive contextual and semantic information, thus enhancing the models' effectiveness in PI. These findings further corroborate the results presented in the experiments discussed in Chapter 5.

8.4.2. Efficiency of Generative Pre-trained Transformer Models in Classification

The state-of-the-art models introduced by OpenAI are examined, with particular attention given to ChatGPT, which is based on the architectures of GPT-3.5 and GPT-4, in addition to three models that have been specifically fine-tuned on the ALECS-SS dataset. OpenAI recommends evaluating ChatGPT's inherent performance before engaging in fine-tuning, as it is anticipated to deliver superior results in certain tasks without the need for additional adjustments. Nevertheless, the focus of this experimental evaluation is placed on assessing

GPT models' classification capabilities, rather than examining their potential for text generation. As noted by (Anderson et al., 2023; Weber-Wulff et al., 2023; Mitchell et al., 2023), identifying automatically generated text is generally less challenging than detecting machine-paraphrased content. Therefore, the goal of this study is to assess the effectiveness of these advanced models in performing classification tasks and to evaluate their performance in comparison to models that have been fine-tuned specifically for the ALECS-SS dataset.

The findings presented in Tables 8.6 and 8.7 reveal that ChatGPT's performance lacks consistency and proves to be less effective, especially when applied to classification tasks related to PI. Despite OpenAI's expectations, ChatGPT did not deliver consistent results in this area. Conversely, the models fine-tuned on the ALECS-SS dataset achieved significantly high F1-micro scores using only a few hundred samples, demonstrating their robustness and efficiency in executing classification tasks effectively.

Specifically, Table 8.6 illustrates the outcomes of using ChatGPT as a classifier model to differentiate between human-written and machine-paraphrased paragraphs. The results in Table 8.5 provide a comparison of the performance of various ChatGPT models on targeted prompts, evaluated using the F1-micro score. ChatGPT-3.5 exhibits varying performance based on the prompt engineering method. When the prompt is generated by ChatGPT, the F1-micro score is 64.2, while it increases to 72.0 for the prompt engineered by the author. Both prompts instruct the model to return a value of 0 for human-written paragraphs and 1 for machine-paraphrased paragraphs. However, the second prompt, created by the author, provides more detailed instructions regarding the task compared to the prompt generated by ChatGPT: Additionally, ChatGPT-3.5-turbo, across its two versions (0613 and 0125), records F1-micro scores ranging from 40 to 60 for both the “Zero Shot” and “Few Shot” prompts. Notably, ChatGPT-4 lags with an F1-micro score of 18.5 for “Zero Shot” and 41.4 for “Few Shot.”. These findings support OpenAI's suggestion to prioritise the use of GPT-3.5 over GPT-4.

Table 8.6. Zero-shot and Few-shot results for the paraphrase identification (PI) task at paragraph-level

Model	Prompt	F1-score

ChatGPT3.5	Generated by ChatGPT	64.2
	Created by the author	72.0
ChatGPT3.5-turbo-0613	Zero Shot	40.2
	Few Shot	60.3
ChatGPT3.5-turbo-0125	Zero Shot	53.1
	Few Shot	53.4
ChatGPT4	Zero Shot	18.5
	Few Shot	41.4

The results presented in Table 8.7 demonstrate that the fine-tuned GPT models surpass the performance achieved when these models are implemented without tuning. The F1-micro score increases dramatically to reach 93.7 and 96.0 by considering only 800 samples for training the models. While OpenAI recommends beginning with 100 samples for initial training, our findings indicate that utilising 800 samples significantly improves the model's ability to distinguish between machine-paraphrased and human-written paragraphs. It is argued that using more than 100 samples is essential for fine-tuning the models, particularly considering the complexity of this task, which requires generating predictions without the advantage of directly comparing two texts. This approach guarantees a more robust adaptation to the task, enhancing the model's performance and the overall outcomes. Notably, the results of the fine-tuned GPT models outperform those achieved by the autoencoder LLMs employed during the paraphrasing stage. The fine-tuning and evaluation of the GPT-3.5-turbo models are conducted using 800 randomly selected samples from the ALECS-SS dataset, while the autoencoder LLMs results (Table 7.2) are derived by utilising the entire dataset.

The results for the GPT models are obtained by considering the most challenging text, where only 15% of the tokens had been paraphrased. In line with OpenAI's recommendation, the GPT-3.5-turbo versions are considered, as GPT-4 currently unavailable for fine-tuning by individuals. The ability of the Davinci model is further investigated, as it is intended to serve as a replacement for earlier GPT models specifically trained for classification tasks. However, a weak result of 48.9 is presented. Based on the results obtained, fine-tuned GPT-3.5-turbo models are advocated over autoencoder LLMs for detecting plagiarism and identifying paraphrased paragraphs. However, it must be acknowledged that the fine-tuning and implementation of GPT models entail considerable expenses.

Table 8.7. Fine-tuned GPTs results for the paraphrase identification (PI) task on ALECS-SS

Model	F1-score
gpt-3.5-turbo-1106	96.0
gpt-3.5-turbo-0125	93.7
Davinci 002	48.9

8.5. Summary

This study has conducted an extensive investigation into PI utilising several advanced models, specifically focusing on the differentiation between human-authored and machine-generated paraphrased content at the paragraph-level. The ALECS-SS dataset, which provides a diverse range of paraphrased paragraphs both lexically and syntactically, is utilised to analyse the performance of various LLMs in terms of PI. A comparative assessment is conducted between auto-encoding models, such as BERT, RoBERTa, and Longformer, and autoregressive models, including GPT-3.5 and GPT-4.

Significantly, fine-tuned versions of GPT-3.5-turbo outperformed the auto-encoding LLMs in the task of identifying paraphrased paragraphs, even when required to make predictions without direct access to source texts for comparison. This is especially evident when fine-tuning is performed with small datasets (up to 800 samples), as opposed to the limited sample (100 samples) fine-tuning suggested by OpenAI.

The results highlighted several significant findings: Initially, fine-tuning GPT-3.5-turbo substantially enhanced classification performance, attaining F1-micro scores of 93.7 and 96.0, which greatly exceeded the results of prior studies utilising MLMs. The job of PI is notably difficult when a minimal fraction of tokens is paraphrased, as evidenced by experiments using 15% paraphrased text of the ALECS-SS dataset. Despite this, the optimised GPT models exhibited strong performance, confirming their proficiency in managing the task even in challenging conditions.

CHAPTER 9: CONCLUSION

Given the ease of access to textual content online, the advancements in LLMs and online paraphrasing tools, which can produce sophisticated paraphrases, this thesis presents novel methods and analyses for detecting paraphrased text at the paragraph-level in Chapters 4,5,6 and 7. This focus is particularly relevant, as paragraph-level paraphrasing is more commonly employed by individuals attempting to commit plagiarism (Foltýnek et al., 2019).

9.1. Revisiting RQs

This thesis investigated paraphrase generation and identification at the paragraph-level, addressing the overarching question of how text length and paraphrasis type influence the efficiency of ML and DL approaches.

To respond to RQ1 and RQ2, the initial approach to detect paragraph-level paraphrases content is introduced in Chapter 5, where the significance of this detection is thoroughly examined. By considering text length and the available datasets, experiments employ both ML and DL methods. In these experiments, handcrafted features that are demonstrated to yield robust results in prior research (see Chapter 3) are extracted from each subset of the datasets. The samples are categorised according to text length and paraphrasing level. The experiments in this thesis focus on English text, specifically targeting text with an average paragraph length of 50 to 150 words. The findings indicate that short text segments do not provide sufficient information for comparing the semantic equality of two segments, resulting in low performance for both ML and DL methods within this category. Conversely, longer text segments often contain excessive semantic information, which can confuse the ML and DL models. Mid-length texts (paragraphs) offered the most reliable basis for PI, with handcrafted features such as TF-IDF and n-gram overlap proving effective. SBERT achieved strong results on short and mid-length inputs but struggled on longer texts, demonstrating the practical trade-offs between approaches.

Building on this, Chapter 6 introduced the ALECS-SS dataset and the SALAC algorithms to answer RQ3-RQ5. The ALECS-SS dataset has been artificially established with the assistance of LLMs. This innovative approach enables the generation of diverse and

contextually rich paraphrased texts, which addresses some of the limitations associated with traditional corpus construction methods. By leveraging the capabilities of LLMs and SALAC algorithms, the dataset is designed to provide a wide range of paragraph-level paraphrases that maintain semantic equivalence while varying in linguistic structure and lexicalisation. This not only enhances the dataset's robustness but also facilitates more comprehensive analyses and experiments in the field of PI. In addition, three innovative SALAC algorithms are created to reconstruct paragraphs while preserving their semantic integrity. Following this reconstruction, a masking technique is utilised to modify the text lexically while maintaining its source meaning. These methods take into account both inter-sentence and intra-sentence relationships within the source text. The combination of algorithms and techniques generates paraphrased content that maintains the meaning of the source text, effectively preserving the semantic relationships and contextual subtleties. The research seeks to improve the quality and diversity of generated paraphrases by the implementation of these methodologies, hence facilitating more effective PI and analysis. Subsequently, both human and automatic evaluation procedures are conducted on the generated paraphrased paragraphs showing that sentence reordering combined with lexical masking could generate coherent paraphrases while preserving meaning. Human evaluation confirmed the validity of these algorithms, while coherence scores provided further evidence that automatic metrics can approximate, though not replace, human judgments.

To address RQ6 and RQ7, Chapter 7 extended the analysis to multiple domains, showing that although the SALAC algorithms generalise well across disciplines, paraphrase quality is influenced by domain complexity. The study also revealed an inverse relationship between text complexity and IAA: more complex texts produced greater variability in human judgments, whereas clearer texts with shorter words, simpler structures, and concise sentences led to higher agreement. The consistency between automatic and human evaluation further confirmed the reliability of the algorithms across domains, highlighting both their strengths and the need for sensitivity to domain-specific challenges. Finally, to respond to RQ8 and RQ9, Chapter 8 focuses on a classifier tasked with distinguishing between human-written and machine-paraphrased text. This shift in focus is motivated by the advancements in LLMs which have demonstrated the ability to generate human-like text. The growing sophistication of these models necessitated the exploration of classification techniques capable of distinguishing between human-written and machine-paraphrased paragraphs, as this has become a critical area of study in response to the evolving capabilities of LLMs and the rapid growth of generative

AI. In this classification task, the classifier is fed one paragraph at a time, generating a score of 0 if the text is written by a human or 1 if it is a machine-paraphrased version. Notably, this task is more challenging, as the algorithm must decide without prior information. Additionally, both sets of paragraphs are originally written by humans, while the paraphrased versions alter a specific percentage of the text tokens and the text structure. In this thesis, three LLMs: BERT, RoBERTa, and Longformer are used to paraphrase 15%, 20%, and 30% of the SALAC outputs. The results of these experiments are analysed in depth, leading to several significant insights. Firstly, the finding confirms the impact of text length on the efficiency of DL classification models. Moreover, it highlights the effect of the masking probability percentage applied during the paraphrasing stage on the results of the experiments. Furthermore, the findings underscore the efficiency and differences between the implemented autoencoding and autoregressive LLMs. Generally, this chapter demonstrated that both autoencoding models (BERT, RoBERTa, Longformer) and autoregressive models (GPTs) are capable of distinguishing human-written from machine-paraphrased text. Together, these findings highlight the robustness of paragraph-level PI compared to sentence-level approaches and show its practical relevance to PD.

9.2. Limitations

The study is subject to several limitations that shape the scope of its contributions. The first concerns the domain restriction of the ALECS-SS dataset, which was constructed primarily from social science texts such as psychology, sociology, and economics. Thus, this focus limits the generalisability of the findings to disciplines with different writing conventions, such as the sciences. A further limitation relates to dataset size and sample variety. Although ALECS-SS is both novel and considerably larger than existing sentence- and paragraph-level resources, it is composed solely of machine-paraphrased paragraphs and does not yet include real cases of academic plagiarism. Human evaluation also presents challenges: while the study engaged Durham University students from a range of academic backgrounds with strong English proficiency, the inclusion of domain experts could provide deeper insight into discipline-specific demands. In addition, each sample was assessed by only three participants, and increasing this number to five or more would likely yield more reliable and robust results. Finally, computational resources further restricted the scope of the work, as transformer models, particularly GPT-based systems, were only partially explored due to the high cost of training and fine-tuning at scale.

9.3. Future Work

Future research should address these limitations in order to strengthen and extend the findings. A clear priority is the expansion of the ALECS-SS dataset to incorporate additional domains beyond the social sciences and real-world paraphrase plagiarism cases, particularly technical and scientific writing where paraphrasing practices and stylistic conventions differ. Extending ALECS-SS to multilingual contexts would also enhance the applicability of PI methods to global academic settings. Complementing this, human evaluation would benefit from broader participation, both in scale and in domain specialized, to mitigate bias and produce more reliable and representative judgments.

On the algorithmic side, combining structural methods such as those embodied in SALAC with the expressive power of fine-tuned generative models offers a promising avenue for generating paraphrases that are both natural and meaning-preserving. At the same time, the rapid evolution of generative AI presents both challenges and opportunities. The increasing fluency of models such as GPT-4 and GPT-5 will make plagiarism harder to detect, yet these same models could be harnessed as evaluators, detectors, or data generators. Beyond the technical challenges, there is scope to embed paragraph-level PI methods more directly into educational integrity systems and publishing workflows. Such integration would not only aid detection but also serve a pedagogical role, supporting students in developing sound paraphrasing practices by offering feedback on their work.

BIBLIOGRAPHY

Abd-Elaal, E.-S., Gamage, S. H. P. W., & Mills, J. E. (2022). Assisting academics to identify computer generated writing. *European Journal of Engineering Education*, *47*(5), 725–745. https://doi.org/10.1080/03043797.2022.2046709

Afzal, A., Vladika, J., Braun, D., & Matthes, F. (2023). Challenges in Domain-Specific Abstractive Summarization and How to Overcome Them: *Proceedings of the 15th International Conference on Agents and Artificial Intelligence*, 682–689. https://doi.org/10.5220/0011744500003393

Agarwal, B. (2018). A deep network model for paraphrase detection in short text messages. *Information Processing and Management*, 16.

Al Saqaabi, A., Akrida, E., Cristea, Alexandra. I., & Stewart, C. (2022). A Paraphrase Identification Approach in Paragraph Length Texts. *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, 358–367. https://doi.org/10.1109/ICDMW58026.2022.00055

Alamleh, H., AlQahtani, A. A. S., & ElSaid, A. (2023). Distinguishing Human-Written and ChatGPT-Generated Text Using Machine Learning. *2023 Systems and Information Engineering Design Symposium (SIEDS)*, 154–158. https://doi.org/10.1109/SIEDS58326.2023.10137767

Alsallal, M., Iqbal, R., Amin, S., & James, A. (2013). Intrinsic Plagiarism Detection Using Latent Semantic Indexing and Stylometry. *2013 Sixth International Conference on Developments in eSystems Engineering*, 145–150. https://doi.org/10.1109/DeSE.2013.34

AlSallal, M., Iqbal, R., Palade, V., Amin, S., & Chang, V. (2019). An integrated approach for intrinsic plagiarism detection. *Future Generation Computer Systems*, *96*, 700–712. https://doi.org/10.1016/j.future.2017.11.023

Alvero, A. J., Lee, J., Regla-Vargas, A., Kizilcec, R. F., Joachims, T., & Antonio, A. L. (2024). Large language models, social demography, and hegemony: Comparing authorship in human and synthetic text. *Journal of Big Data*, *11*(1), 138. https://doi.org/10.1186/s40537-024-00986-7

Alzahrani, S., & Salim, N. (2010). Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection. *Braschler and Harman*, *1176*, 1–8.

Alzahrani, S., Salim, N., Abraham, A., & Palade, V. (2011). iPlag: Intelligent Plagiarism Reasoner in scientific publications. *2011 World Congress on Information and Communication Technologies*, 1–6. https://doi.org/10.1109/WICT.2011.6141191

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

Amin, M. M., Cambria, E., & Schuller, B. W. (2023). Can ChatGPT's Responses Boost Traditional Natural Language Processing? *IEEE Intelligent Systems*, *38*(5), 5–11. https://doi.org/10.1109/MIS.2023.3305861

Anderson, N., Belavy, D. L., Perle, S. M., Hendricks, S., Hespanhol, L., Verhagen, E., & Memon, A. R. (2023). AI did not write this manuscript, or did it? Can we trick the AI text detector into generated texts? The potential future of ChatGPT and AI in Sports & Exercise Medicine manuscript generation. *BMJ Open Sport & Exercise Medicine*, *9*(1), e001568. https://doi.org/10.1136/bmjsem-2023-001568

Arase, Y., & Tsujii, J. (2021). Transfer fine-tuning of BERT with phrasal paraphrases. *Computer Speech & Language*, *66*, 101164. https://doi.org/10.1016/j.csl.2020.101164

Artstein, R. (2017). Inter-annotator Agreement. In N. Ide & J. Pustejovsky (Eds.), *Handbook of Linguistic Annotation* (pp. 297–313). Springer Netherlands. https://doi.org/10.1007/978-94-024-0881-2_11

Asghari, H., Fatemi, O., Mohtaj, S., & Faili, H. (2021). A crowdsourcing approach to construct mono-lingual plagiarism detection corpus. *International Journal on Digital Libraries*, *22*(1), 49–61. https://doi.org/10.1007/s00799-020-00294-4

Astila, I. (2019). Students Awareness of Plagiarism in Paraphrasing English Text. *Getsempena English Education Journal*, *6*(2), 258–266. https://doi.org/10.46244/geej.v6i2.882

Bach, N. X., Minh, N. L., & Shimazu, A. (2014). Exploiting discourse information to identify paraphrases. *Expert Systems with Applications*, *41*(6), 2832–2841. https://doi.org/10.1016/j.eswa.2013.10.018

Bär, D., Zesch, T., & Gurevych, I. (2012). *Text Reuse Detection Using a Composition of Text Similarity Measures*. 18.

Barrón-Cedeño & Vila, M. (2013). Plagiarism Meets Paraphrasing: Insights for the Next Generation in Automatic Plagiarism Detection. *Computational Linguistics*, *39*(4), 32.

Bayerl, P. S., & Paul, K. I. (2011). What Determines Inter-Coder Agreement in Manual Annotations? A Meta-Analytic Investigation. *Computational Linguistics*, *37*(4), 699–725. https://doi.org/10.1162/COLI_a_00074

Becker, J., Wahle, J. P., Ruas, T., & Gipp, B. (2023). *Paraphrase Detection: Human vs. Machine Content* (No. arXiv:2303.13989). arXiv. http://arxiv.org/abs/2303.13989

Beltagy, I., Peters, M. E., & Cohan, A. (2020). *Longformer: The Long-Document Transformer* (No. arXiv:2004.05150). arXiv. http://arxiv.org/abs/2004.05150

Belz, A., & Reiter, E. (2006). Comparing automatic and human evaluation of NLG systems. *In Proc. 11th Conf. European Chapter of the Association for Computational Linguistics* (pp. 313–320).

Ben Aouicha, M., Hadj Taieb, M. A., & Ben Hamadou, A. (2018). SISR: System for integrating semantic relatedness and similarity measures. *Soft Computing*, *22*(6), 1855–1879. https://doi.org/10.1007/s00500-016-2438-x

Bengio, Y., Ducharme, R., & Vincent, P. (2000). A Neural Probabilistic Language Model. *Advances in Neural Information Processing Systems*, *13*.

Bensal, E. R. (2013). Plagiarism: Shall We Turn to Turnitin? *Computer-Assisted Language Learning-Electronic Journal*, *14(2)*, 2–22.

Bensalem, I., Rosso, P., & Chikhi, S. (2014). Intrinsic Plagiarism Detection using N-gram Classes. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1459–1464. https://doi.org/10.3115/v1/D14-1153

Bhagat, R., & Hovy, E. (2013). What Is a Paraphrase? *Computational Linguistics*, *39*(3), 463–472. https://doi.org/10.1162/COLI_a_00166

Biber, D., Conrad, S., & Reppen, R. (1998). *Corpus Linguistics: Investigating Language Structure and Use* (1st ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511804489

Birch, A., Blunsom, P., & Osborne, M. (2009). A quantitative analysis of reordering phenomena. *Proceedings of the Fourth Workshop on Statistical Machine Translation - StatMT '09*, 197. https://doi.org/10.3115/1626431.1626471

Bland, J. M., & Altman, D. G. (1995). Statistics notes: Multiple significance tests: the Bonferroni method. *BMJ*, *310*(6973), 170–170. https://doi.org/10.1136/bmj.310.6973.170

Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2017) 'Enriching word vectors with subword information', Transactions of the association for computational linguistics, 5, pp. 135-146.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., & Henighan, T. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Burrows, S., Potthast, M., & Stein, B. (2013). Paraphrase acquisition via crowdsourcing and machine learning. *ACM Transactions on Intelligent Systems and Technology*, *4*(3), 1–21. https://doi.org/10.1145/2483669.2483676

Castilho, S., & Knowles, R. (2024). A survey of context in neural machine translation and its evaluation. *Natural Language Processing*, 1–31. https://doi.org/10.1017/nlp.2024.7

Celikyilmaz, A., Clark, E., & Gao, J. (2021). *Evaluation of Text Generation: A Survey* (No. arXiv:2006.14799). arXiv. https://doi.org/10.48550/arXiv.2006.14799

Chae, D.-K., Ha, J., Kim, S.-W., Kang, B., & Im, E. G. (2013). Software plagiarism detection: A graph-based approach. *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management - CIKM '13*, 1577–1580. https://doi.org/10.1145/2505515.2507848

Chali, Y., & Egonmwan, E. (2024). Transfer-Learning based on Extract, Paraphrase and Compress Models for Neural Abstractive Multi-Document Summarization. *In Proceedings of the 17th International Natural Language Generation Conference*, 213--221. https://aclanthology.org/2024.inlg-main.17

Chang, C.-Y., Lee, S.-J., Wu, C.-H., Liu, C.-F., & Liu, C.-K. (2021). Using word semantic concepts for plagiarism detection in text documents. *Information Retrieval Journal*. https://doi.org/10.1007/s10791-021-09394-4

Cheng, N., Chandramouli, R., & Subbalakshmi, K. P. (2011). Author gender identification from text. *Digital Investigation*, *8*(1), 78–88. https://doi.org/10.1016/j.diin.2011.04.002

Chi, X., Xiang, Y., & Shen, R. (2020). Paraphrase Detection with Dependency Embedding. *2020 4th International Conference on Computer Science and Artificial Intelligence*, 213–218. https://doi.org/10.1145/3445815.3445850

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches* (No. arXiv:1409.1259). arXiv. http://arxiv.org/abs/1409.1259

Chowdhury, G. G. (2005). Natural language processing. *Annual Review of Information Science and Technology*, *37*(1), 51–89. https://doi.org/10.1002/aris.1440370103

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling* (No. arXiv:1412.3555). arXiv. http://arxiv.org/abs/1412.3555

Church, K. W. (2017). Word2Vec. *Natural Language Engineering*, *23*(1), 155–162. https://doi.org/10.1017/S1351324916000334

Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S., & Smith, N. A. (2021). *All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text* (No. arXiv:2107.00061). arXiv. https://doi.org/10.48550/arXiv.2107.00061

Clough, P., & Stevenson, M. (2011). Developing a corpus of plagiarised short answers. *Language Resources and Evaluation*, *45*(1), 5–24. https://doi.org/10.1007/s10579-009-9112-1

Cohn, T., Callison-Burch, C., & Lapata, M. (2008). Constructing Corpora for the Development and Evaluation of Paraphrase Systems. *Computational Linguistics*, *34*(4), 597–614. https://doi.org/10.1162/coli.08-003-R1-07-044

Coleman, M., & Liau, T. L. (1975). A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, *60*(2), 283–284. https://doi.org/10.1037/h0076540

Cordeiro, J., Dias, G., & Brazdil, P. (2007). A Metric for Paraphrase Detection. *2007 International Multi-Conference on Computing in the Global Information Technology (ICCGI'07)*, 7–7. https://doi.org/10.1109/ICCGI.2007.4

Crossley, S. A., Allen, D. B., & McNamara, D. S. (2011). Text readability and intuitive simplification: A comparison of readability formulas. *Reading in a Foreign Language*, *23*(1), 84-101.

Dagan, I., Karov, Y., & Roth, D. (1997). *Mistake-Driven Learning in Text Categorization* (No. arXiv:cmp-lg/9706006). arXiv. http://arxiv.org/abs/cmp-lg/9706006

Das, D., & Smith, N. A. (2009). Paraphrase identification as probabilistic quasi-synchronous recognition. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - ACL-IJCNLP '09*, *1*, 468. https://doi.org/10.3115/1687878.1687944

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. *1*, 4171–4186. http://arxiv.org/abs/1810.04805

Dirac, G. A. (1952). Some Theorems on Abstract Graphs. *Proceedings of the London Mathematical Society,* s3-2(1), 69–81. https://doi.org/10.1112/plms/s3-2.1.69

Dolan, B., Quirk, C., & Brockett, C. (2004). *Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources*. 7.

Dolan, W. B., & Brockett, C. (2005). Automatically Constructing a Corpus of Sentential Paraphrases. *In Third International Workshop on Paraphrasing*.

Ehsan, N., & Shakery, A. (2016). Candidate document retrieval for cross-lingual plagiarism detection using two-level proximity information. *Information Processing & Management*, *52*(6), 1004–1017. https://doi.org/10.1016/j.ipm.2016.04.006

Ehsan, N., Shakery, A., & Tompa, F. W. (2019). Cross-lingual text alignment for fine-grained plagiarism detection. *Journal of Information Science*, *45*(4), 443–459. https://doi.org/10.1177/0165551518787696

Eisa, T. A. E., Salim, N., & Alzahrani, S. (2015). Existing plagiarism detection techniques: A systematic mapping of the scholarly literature. *Online Information Review*, *39*(3), 383–400. https://doi.org/10.1108/OIR-12-2014-0315

Elkhatat, A. M. (2023). Evaluating the authenticity of ChatGPT responses: A study on text-matching capabilities. *International Journal for Educational Integrity*, *19*(1), 15. https://doi.org/10.1007/s40979-023-00137-0

Elkhatat, A. M., Elsaid, K., & Almeer, S. (2021). Some students plagiarism tricks, and tips for effective check. *International Journal for Educational Integrity*, *17*(1), 15. https://doi.org/10.1007/s40979-021-00082-w

Elkhatat, A. M., Elsaid, K., & Almeer, S. (2023). Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text. *International Journal for Educational Integrity*, *19*(1), 17. https://doi.org/10.1007/s40979-023-00140-5

Elsner, M. & Charniak, E. (2011, June). *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).

Ezzikouri, H., Erritali, M., & Oukessou, M. (2017). Fuzzy-Semantic Similarity for Automatic Multilingual Plagiarism Detection. *International Journal of Advanced Computer Science and Applications*, *8*(9). https://doi.org/10.14569/IJACSA.2017.080912

Fei, Z., Shen, X., Zhu, D., Zhou, F., Han, Z., Zhang, S., Chen, K., Shen, Z., & Ge, J. (2023). *LawBench: Benchmarking Legal Knowledge of Large Language Models* (No. arXiv:2309.16289). arXiv. https://doi.org/10.48550/arXiv.2309.16289

Fernando, S., & Stevenson, M. (2008). A Semantic Similarity Approach to Paraphrase Detection. *In Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*, 45–52.

Ferreira, R., Cavalcanti, G. D. C., Freitas, F., Lins, R. D., Simske, S. J., & Riss, M. (2018). Combining sentence similarities measures to identify paraphrases. *Computer Speech & Language*, *47*, 59–73. https://doi.org/10.1016/j.csl.2017.07.002

Flesch, R. F. (1979). *How to Write in Plain English: A Book for Lawyers and Consumers*. Harper.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res., 3(Mar), 1289-1305.

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

Foltýnek, T., Meuschke, N., & Gipp, B. (2020). Academic Plagiarism Detection: A Systematic Literature Review. *ACM Computing Surveys*, *52*(6), 1–42. https://doi.org/10.1145/3345317

Ganitkevitch, J., Durme, B. V., & Callison-Burch, C. (2013). *PPDB: The Paraphrase Database*.

Gehrmann, S., Clark, E., & Sellam, T. (2023). Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text. *Journal of Artificial Intelligence Research*, *77*, 103–166. https://doi.org/10.1613/jair.1.13715

Gunning, R. (1952). *The Technique of Clear Writing*. McGraw-Hill.

Guozhu, D., & Liu, H. (2018). *Feature Engineering for Machine Learning and Data Analytics* (1st ed.). CRC Press. https://doi.org/10.1201/9781315181080

Gupta, A., Agarwal, A., Singh, P., & Rai, P. (2018). A Deep Generative Framework for Paraphrase Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). https://doi.org/10.1609/aaai.v32i1.11956

Gupta, D., Vani, K., & Singh, C. K. (2014). Using Natural Language Processing techniques and fuzzy-semantic similarity for automatic external plagiarism detection. *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2694–2699. https://doi.org/10.1109/ICACCI.2014.6968314

Hagen, M., Potthast, M., & Stein, B. (2015). Source Retrieval for Plagiarism Detection from Large Web Corpora: Recent Approaches. *Sematic Scholar 1391*.

Hämäläinen, M., & Alnajjar, K. (2021). *The Great Misalignment Problem in Human Evaluation of NLP Methods* (No. arXiv:2104.05361). arXiv. http://arxiv.org/abs/2104.05361

Hany, M., & Gomaa, W. H. (2022). A Hybrid Approach to Paraphrase Detection Based on Text Similarities and Machine Learning Classifiers. *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, 343–348. https://doi.org/10.1109/MIUCC55081.2022.9781678

Hashimoto, T. B., Zhang, H., & Liang, P. (2019). *Unifying Human and Statistical Evaluation for Natural Language Generation* (No. arXiv:1904.02792). arXiv. https://doi.org/10.48550/arXiv.1904.02792

He, Y., Wang, Z., Zhang, Y., Huang, R., & Caverlee, J. (2020). 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),*. http://arxiv.org/abs/2010.03725

Hochreiter, S., & Jürgen, S. (1997). *Long short-term memory*. 1735–1780.

Hu, J. E., Rudinger, R., Post, M., & Van Durme, B. (2019). *ParaBank: Monolingual Bitext Generation and Sentential Paraphrasing via Lexically-constrained Neural Machine Translation* (No. arXiv:1901.03644). arXiv. http://arxiv.org/abs/1901.03644

Huang, L., Ma, D., Li, S., Zhang, X., & WANG, H. (2019). *Text Level Graph Neural Network for Text Classification* (No. arXiv:1910.02356). arXiv. http://arxiv.org/abs/1910.02356

Hunt, E., Janamsetty, R., Kinares, C., Koh, C., Sanchez, A., Zhan, F., Ozdemir, M., Waseem, S., Yolcu, O., Dahal, B., Zhan, J., Gewali, L., & Oh, P. (2019). Machine Learning Models for Paraphrase Identification and its Applications on Plagiarism Detection. *2019 IEEE International Conference on Big Knowledge (ICBK)*, 97–104. https://doi.org/10.1109/ICBK.2019.00021

Incitti, F., Urli, F., & Snidaro, L. (2023). Beyond word embeddings: A survey. *Information Fusion*, *89*, 418–436. https://doi.org/10.1016/j.inffus.2022.08.024

Inojosa, H., Gilbert, S., Kather, J. N., Proschmann, U., Akgün, K., & Ziemssen, T. (2023). Can ChatGPT explain it? Use of artificial intelligence in multiple sclerosis communication. *Neurological Research and Practice*, *5*(1), 48. https://doi.org/10.1186/s42466-023-00270-8

Ji, Y., & Eisenstein, J. (2013). *Discriminative Improvements to Distributional Sentence Similarity*. 6.

João, C., Gaël, D., & Pavel, B. (2007). New Functions for Unsupervised Asymmetrical Paraphrase Detection. *Journal of Software*, *2*(4), 12–23. https://doi.org/10.4304/jsw.2.4.12-23

Kanerva, J., Ginter, F., Chang, L.-H., Rastas, I., Skantsi, V., Kilpeläinen, J., Kupari, H.-M., Saarni, J., Sevón, M., & Tarkka, O. (2021). *Finnish Paraphrase Corpus* (No. arXiv:2103.13103). arXiv. http://arxiv.org/abs/2103.13103

Karen, F., & Park, J. (2002). Improvements for scalable and accurate plagiarism detection in digital documents. *Proceedings of the 8th International Conference on Parallel and Distributed Systems*.

Kartelj, A., Mladenović, M., & Vujičić Stanković, S. (2025). Comparison of algorithms for the recognition of ChatGPT paraphrased texts. *Journal of Big Data*, *12*(1), 28. https://doi.org/10.1186/s40537-025-01082-0

Kenter, T., & de Rijke, M. (2015). Short Text Similarity with Word Embeddings. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1411–1420. https://doi.org/10.1145/2806416.2806475

Kim, R., Kotsenko, A., Andreev, A., Bazanova, A., Aladin, D., Todua, D., Marushchenko, A., & Varlamov, O. (2024). Evaluation of BERT and ChatGPT models in inference, paraphrase and similarity tasks. *E3S Web of Conferences*, *515*, 03016. https://doi.org/10.1051/e3sconf/202451503016

Kincaid, J. P. (1975). *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel*.

Ko, B., & Choi, H. (2020). Twice fine-tuning deep neural networks for paraphrase identification. *Electronics Letters*, *56*(9), 444–447. https://doi.org/10.1049/el.2019.4183

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners. *Advances in Neural Information Processing Systems*, *35*, 22199–22213.

Korde, V. (2012). Text Classification and Classifiers:A Survey. *International Journal of Artificial Intelligence & Applications*, *3*(2), 85–99. https://doi.org/10.5121/ijaia.2012.3208

Korenius, T., Laurikkala, J., Järvelin, K., & Juhola, M. (2004). Stemming and lemmatization in the clustering of finnish text documents. *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, 625–633. https://doi.org/10.1145/1031171.1031285

Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, & Brown. (2019). Text Classification Algorithms: A Survey. *Information*, *10*(4), 150. https://doi.org/10.3390/info10040150

Krishna, K., Song, Y., Karpinska, M., Wieting, J., & Iyyer, M. (2023). *Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense* (No. arXiv:2303.13408). arXiv. http://arxiv.org/abs/2303.13408

Krisztián, M., Finkel, R., Zaslavsky, A., Hodász, G., & Pataki, M. (2002). Comparison of overlap detection techniques. *N Computational Science—ICCS 2002: International Conference Amsterdam, The Netherlands*, 51–60.

Kurt Pehlivanoğlu, M., Gobosho, R. T., Syakura, M. A., Shanmuganathan, V., & de-la-Fuente-Valentín, L. (2024). Comparative analysis of paraphrasing performance of ChatGPT, GPT-3, and T5 language models using a new ChatGPT generated dataset: ParaGPT. *Expert Systems*, *41*(11), e13699. https://doi.org/10.1111/exsy.13699

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations* (No. arXiv:1909.11942). arXiv. http://arxiv.org/abs/1909.11942

Lancaster, T., & Culwin, F. (2004). Using freely available tools to produce a partially automated plagiarism detection process. *Beyond the Comfort Zone: Proceedings of the 21st ASCILITE Conference*, 520–529.

Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, *33*(1), 159. https://doi.org/10.2307/2529310

Larock, M.H., Tressler, J.C., & Lewis, C.E.(1980). *Mastering effective English*. Copp Clark Pitman, Mississauga.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Lenzner, T. (2014). Are Readability Formulas Valid Tools for Assessing Survey Question Difficulty? *Sociological Methods & Research*, *43*(4), 677–698. https://doi.org/10.1177/0049124113513436

Li, B., Liu, T., Wang, B., & Wang, L. (2021). Enhancing Deep Paraphrase Identification via Leveraging Word Alignment Information. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7843–7847. https://doi.org/10.1109/ICASSP39728.2021.9414944

Li, H., & Xu, J. (2014). Semantic Matching in Search. *Foundations and Trends® in Information Retrieval*, *7*(5), 343–469. https://doi.org/10.1561/1500000035

Lin, Z., Cai, Y., & Wan, X. (2021). *Towards Document-Level Paraphrase Generation with Sentence Rewriting and Reordering* (No. arXiv:2109.07095). arXiv. http://arxiv.org/abs/2109.07095

Liu, X., You, X., Zhang, X., Wu, J., & Lv, P. (2020). Tensor Graph Convolutional Networks for Text Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 8409–8416. https://doi.org/10.1609/aaai.v34i05.6359

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach* (No. arXiv:1907.11692). arXiv. http://arxiv.org/abs/1907.11692

Liu, Z., Lin, Y., & Sun, M. (Eds.). (2023). *Representation Learning for Natural Language Processing*. Springer Nature Singapore. https://doi.org/10.1007/978-981-99-1600-9

Manzoor, M. F., Farooq, M. S., Haseeb, M., Farooq, U., Khalid, S., & Abid, A. (2023). Exploring the Landscape of Intrinsic Plagiarism Detection: Benchmarks, Techniques, Evolution, and Challenges. *IEEE Access*, *11*, 140519–140545. https://doi.org/10.1109/ACCESS.2023.3338855

Maurer, H. A., Kappe, F., & Zaka, B. (2006). Plagiarism—A Survey. *J. Univers. Comput. Sci*, *12.8 (2006): 1050-1084.*

McLaughlin, H. M. (1969). *SMOG Grading – A New Readability Formula n the journal of reading*. 12(8), 639–646

Mihalcea, R., Chklovski, T., & Kilgarriff, A. (2004). The SENSEVAL–3 English Lexical Sample Task. *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 25–28.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*, *26*.

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. *In International Conference on Machine Learning*, 24950–24962.

Mohebbi, M., & Talebpour, A. (2016). *Texts Semantic Similarity Detection Based Graph Approach*. *Int. Arab J. Inf. Technol.,* 13(2), 246–251.

Momtaz, M., Bijari, K., Salehi, M., & Veisi, H. (2016). *Graph-based Approach to Text Alignment for Plagiarism Detection in Persian Documents*. 4.

Muangprathub, J., Kajornkasirat, S., & Wanichsombat, A. (2021). Document Plagiarism Detection Using a New Concept Similarity in Formal Concept Analysis. *Journal of Applied Mathematics*, *2021*, 1–10. https://doi.org/10.1155/2021/6662984

Muneer, I., Shehzadi, A., Ashraf, M. A., & Nawab, R. M. A. (2025). Has machine paraphrasing skills approached humans? Detecting automatically and manually generated paraphrased cases. *Big Data Research*, *39*, 100507. https://doi.org/10.1016/j.bdr.2025.100507

Nguyen, H. T., Duong, P. H., & Cambria, E. (2019). Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowledge-Based Systems*, *182*, 104842. https://doi.org/10.1016/j.knosys.2019.07.013

Oberreuter, G., & Velásquez, J. D. (2013). Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style. *Expert Systems with Applications*, *40*(9), 3756–3763. https://doi.org/10.1016/j.eswa.2012.12.082

Opitz, J., & Frank, A. (2022). SBERT Studies Meaning Representations: Decomposing Sentence Embeddings into Explainable Semantic Features. *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, 625–637. https://aclanthology.org/2022.aacl-main.48

Osman, A. H., Salim, N., Binwahlan, M. S., Twaha, S., Kumar, Y. J., & Abuobieda, A. (2012). Plagiarism detection scheme based on Semantic Role Labeling. *2012 International Conference on Information Retrieval & Knowledge Management*, 30–33. https://doi.org/10.1109/InfRKM.2012.6204978

Özşen, T., Saka, İ., Çelik, Ö., Razı, S., Akkan, S. Ç., & Dlabolova, D. H. (2023). Testing of support tools to detect plagiarism in academic Japanese texts. *Education and Information Technologies*, *28*(10), 13287–13321. https://doi.org/10.1007/s10639-023-11718-4

Palivela, H. (2021). Optimization of paraphrase generation and identification using language models in natural language processing. *International Journal of Information Management Data Insights*, *1*(2), 100025. https://doi.org/10.1016/j.jjimei.2021.100025

Pagliardini, M., Gupta, P., & Jaggi, M. (2018). Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 1 (Long Papers), 528–540. https://doi.org/10.18653/v1/N18-1049

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318.

Patil, R., Boit, S., Gudivada, V., & Nandigam, J. (2023). A Survey of Text Representation and Embedding Techniques in NLP. *IEEE Access*, *11*, 36120–36146. https://doi.org/10.1109/ACCESS.2023.3266377

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Perkins, M., Roe, J., Postma, D., McGaughran, J., & Hickerson, D. (2023). Detection of GPT-4 Generated Text in Higher Education: Combining Academic Judgement and Software to Identify Generative AI Tool Misuse. *Journal of Academic Ethics*. https://doi.org/10.1007/s10805-023-09492-6

Polydouri, A., Vathi, E., Siolas, G., & Stafylopatis, A. (2020). An efficient classification approach in imbalanced datasets for intrinsic plagiarism detection. *Evolving Systems*, *11*(3), 503–515. https://doi.org/10.1007/s12530-018-9232-1

Potter, W. J., & Levine-Donnerstein, D. (1999). Rethinking validity and reliability in content analysis. *J. Appl. Commun. Res*., 27, 258–284.

Potthast, M., Barrón-Cedeño, A., Eiselt, A., Stein, B., & Rosso, P. (2010). Overview of the 2nd International Competition on Plagiarism Detection. *CEUR Workshop Proceedings*, *1176*, 1–14.

Potthast, M., Eiselt, A., Barrón-Cedeño, A., Stein, B., & Rosso, P. (2011). Overview of the 3rd International Competition on Plagiarism Detection. *The CLEF 2011 Evaluation Labs*, *1177*.

Potthast, M., Stein, B., Barrón-Cedeño, A., & Rosso, P. (2010). An Evaluation Framework for Plagiarism Detection. *In Coling 2010: Posters (Pp. 997-1005)*.

Prentice, F. M., & Kinden, C. E. (2018). Paraphrasing tools, language translation tools and plagiarism: An exploratory study. *International Journal for Educational Integrity*, *14*(1), 11. https://doi.org/10.1007/s40979-018-0036-7

Puvvada, N., Revanuru, K., & Teja, S. (2017). *Quora question pairs dataset. Unpublished manuscript*

Qiu, D. (2022). Document-level paraphrase generation base on attention enhanced graph LSTM. *Applied Intelligence, 1-13*, 13.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 1–67.

Razaq, A., Halim, Z., Ur Rahman, A., & Sikandar, K. (2024). Identification of paraphrased text in research articles through improved embeddings and fine-tuned BERT model. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-024-18359-w

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* (No. arXiv:1908.10084). arXiv. http://arxiv.org/abs/1908.10084

Reynolds, L., & McDonell, K. (2021). Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–7. https://doi.org/10.1145/3411763.3451760

Roostaee, M., Fakhrahmad, S. M., & Sadreddini, M. H. (2020). Cross-language text alignment: A proposed two-level matching scheme for plagiarism detection. *Expert Systems with Applications*, *160*, 113718. https://doi.org/10.1016/j.eswa.2020.113718

Roostaee, M., Sadreddini, M. H., & Fakhrahmad, S. M. (2020). An effective approach to candidate retrieval for cross-language plagiarism detection: A fusion of conceptual and keyword-based schemes. *Information Processing & Management*, *57*(2), 102150. https://doi.org/10.1016/j.ipm.2019.102150

Ruiz, M. E., & Padmini, S. (1997). Automatic text categorization using neural networks. *Advances in Classification Research Online*, 58–68.

Saini, A., Sri, M. R., & Thakur, M. (2021). Intrinsic Plagiarism Detection System Using Stylometric Features and DBSCAN. *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 13–18. https://doi.org/10.1109/ICCCIS51004.2021.9397187

Sánchez-Vega, F., Villatoro-Tello, E., Montes-y-Gómez, M., Rosso, P., Stamatatos, E., & Villaseñor-Pineda, L. (2019). Paraphrase plagiarism identification with character-level features. *Pattern Analysis and Applications*, *22*(2), 669–681. https://doi.org/10.1007/s10044-017-0674-z

Sánchez-Vega, F., Villatoro-Tello, E., Montes-y-Gómez, M., Villaseñor-Pineda, L., & Rosso, P. (2013). Determining and characterizing the reused text for plagiarism detection. *Expert Systems with Applications*, *40*(5), 1804–1813. https://doi.org/10.1016/j.eswa.2012.09.021

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). *DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter* (No. arXiv:1910.01108). arXiv. http://arxiv.org/abs/1910.01108

Saputro, W. F., Djamal, E. C., & Ilyas, R. (2019). Paraphrase Identification Between Two Sentence Using Support Vector Machine. *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, 406–411. https://doi.org/10.1109/ICEEI47359.2019.8988874

Schmidtova, P., Mahamood, S., Balloccu, S., Dusek, O., Gatt, A., Gkatzia, D., Howcroft, D. M., Platek, O., & Sivaprasad, A. (2024). *Automatic Metrics in Natural Language Generation: A survey of Current Evaluation Practices*. 557--583. https://aclanthology.org/2024.inlg-main.44/

Servan-Schreiber, D., Cleeremans, A., & McClelland, J. L. (1988). Learning Sequential Structure in Simple Recurrent Networks. *Advances in Neural Information Processing Systems*, *1*.

Shao, Z., Gong, Y., Shen, Y., Huang, M., Duan, N., & Chen, W. (2023). Synthetic Prompting: Generating Chain-of-Thought Demonstrations for Large Language Models. *In International Conference on Machine Learning*, 30706–30775.

Shelke, S., Savant, S., & Joshi, M. (2024). Towards Building Efficient Sentence BERT Models using Layer Pruning. arXiv. https://arxiv.org/abs/2409.14168

Shen, A., Mistica, M., Salehi, B., Li, H., Baldwin, T., & Qi, J. (2021). Evaluating Document Coherence Modeling. *Transactions of the Association for Computational Linguistics*, *9*, 621–640. https://doi.org/10.1162/tacl_a_00388

Shoyukhi, M., Vossen, P. H., Ahmadi, A. H., Kafipour, R., & Beattie, K. A. (2023). Developing a comprehensive plagiarism assessment rubric. *Education and Information Technologies*, *28*(5), 5893–5919. https://doi.org/10.1007/s10639-022-11365-1

Shree, V., & Jayita, S. (2023). A Study on Paraphrase Corpus Detection Using Various ML Models. *2023 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, 43–48. https://doi.org/10.1109/DISCOVER58830.2023.10316665

Smith, E. A., & Senter, R. J. (1967). *Automated Readability Index* (Vol. 66). Aerospace Medical Research Laboratories, Aerospace Medical Division, Air Force Systems Command.

Soricut, R., & Marcu, D. (2006). Discourse generation using utility-trained coherence models. *Proceedings of the COLING/ACL on Main Conference Poster Sessions* -, 803–810. https://doi.org/10.3115/1273073.1273176

Suleiman, D., Awajan, A., & Al-Madi, N. (2017). Deep Learning Based Technique for Plagiarism Detection in Arabic Texts. *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, 216–222. https://doi.org/10.1109/ICTCS.2017.42

Themistocleous, C. (2024). Open Brain AI and language assessment. *Frontiers in Human Neuroscience*, *18*, 1421435. https://doi.org/10.3389/fnhum.2024.1421435

Thompson, B., & Post, M. (2020). *Automatic Machine Translation Evaluation in Many Languages via Zero-Shot Paraphrasing* (No. arXiv:2004.14564). arXiv. http://arxiv.org/abs/2004.14564

Ul-Qayyum, Z., & Wasif, A. (2017). Paraphrase Identification using Semantic Heuristic Features. *Research Journal of Applied Sciences, Engineering and Technology*, *14*(9), 324–333. https://doi.org/10.19026/rjaset.14.5072

van der Lee, C., Gatt, A., van Miltenburg, E., & Krahmer, E. (2021). Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, *67*, 101151. https://doi.org/10.1016/j.csl.2020.101151

Van Enschot, R., Spooren, W., van den Bosch, A., Burgers, C., Degand, L., Evers-Vermeul, J., Kunneman, F., Liebrecht, C., Linders, Y., & Maes, A. (2017). *Taming our wild data: On intercoder reliability in discourse research.*

Vani, K., & Gupta, D. (2017). Identifying document-level text plagiarism: A two-phase approach. *J. Eng. Sci. Technol.*, 12(12), 3226–3250Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, *30*.

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

Vila, M., Martí, M. A., & Rodríguez, H. (2014). Is This a Paraphrase? What Kind? Paraphrase Boundaries and Typology. *Open Journal of Modern Linguistics*, *04*(01), 205–218. https://doi.org/10.4236/ojml.2014.41016

Vrbanec, T., & Meštrović, A. (2020). Corpus-Based Paraphrase Detection Experiments and Review. *Information*, *11*(5), 241. https://doi.org/10.3390/info11050241

Vrbanec, T., & Meštrović, A. (2021). Relevance of Similarity Measures Usage for Paraphrase Detection: *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 129–138. https://doi.org/10.5220/0010649800003064

Vrbanec, T., & Meštrović, A. (2023). Comparison study of unsupervised paraphrase detection: Deep learning—The key for semantic similarity detection. *Expert Systems*, *40*(9), e13386. https://doi.org/10.1111/exsy.13386

Vrublevskyi, V., & Marchenko, O. (2020). Paraphrase Identification Using Dependency Tree and Word Embeddings. *2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT)*, 372–375. https://doi.org/10.1109/ATIT50783.2020.9349338

Wahle, J. P., Ruas, T., Foltýnek, T., Meuschke, N., & Gipp, B. (2022). *Identifying Machine-Paraphrased Plagiarism*. *13192*, 393–413. https://doi.org/10.1007/978-3-030-96957-8_34

Wahle, J. P., Ruas, T., Kirstein, F., & Gipp, B. (2022). How Large Language Models are Transforming Machine-Paraphrased Plagiarism. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 952--963). Association for Computational Linguistics. https://www.authorea.com/users/580669/articles/621672-how-large-language-models-are-transforming-machine-paraphrased-plagiarism?commit=2072221b9db370504c848bb0953cf7a0189853dc

Wahle, J. P., Ruas, T., Meuschke, N., & Gipp, B. (2021). Are Neural Language Models Good Plagiarists? A Benchmark for Neural Paraphrase Detection. *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 226–229. https://doi.org/10.1109/JCDL52503.2021.00065

Wan, S., Dras, M., Dale, R., & Paris, C. (2006). Using Dependency-Based Features to Take the "Para-farce" out of Paraphrase. *In Proceedings of the Australasian Language Technology Workshop*, 8.

Wang, W., Bi, B., Yan, M., Wu, C., Bao, Z., Xia, J., Peng, L., & Si, L. (2019). *StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding* (No. arXiv:1908.04577). arXiv. http://arxiv.org/abs/1908.04577

Wang, X., Li, C., Zheng, Z., & Xu, B. (2018). Paraphrase Recognition via Combination of Neural Classifier and Keywords. *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN.2018.8489222

Weber-Wulff, D., Anohina-Naumeca, A., Bjelobaba, S., Foltýnek, T., Guerrero-Dib, J., Popoola, O., Šigut, P., & Waddington, L. (2023). Testing of detection tools for AI-generated text. *International Journal for Educational Integrity*, *19*(1), 26. https://doi.org/10.1007/s40979-023-00146-z

White, D. R., & Joy, M. S. (2004). Sentence-based natural language plagiarism detection. *Journal on Educational Resources in Computing*, *4*(4), 2. https://doi.org/10.1145/1086339.1086341

Wu, T. (2022, June). Deep neural network in text classification. *In Proc. 2nd Int. Conf. Artificial Intelligence, Big Data and Algorithms (CAIBDA 2022)* (pp. 1–5).

Xu, S., Shen, X., Fukumoto, F., Li, J., Suzuki, Y., & Nishizaki, H. (2020). Paraphrase Identification with Lexical, Syntactic and Sentential Encodings. *Applied Sciences*, *10*(12), 4144. https://doi.org/10.3390/app10124144

Yang, R., Zhang, J., Gao, X., Ji, F., & Chen, H. (2019). Simple and Effective Text Matching with Richer Alignment Features. *arXiv:1908.00300 [Cs]*. http://arxiv.org/abs/1908.00300

Yang, Y., Yih, W., & Meek, C. (2015). WikiQA: A Challenge Dataset for Open-Domain Question Answering. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2013–2018. https://doi.org/10.18653/v1/D15-1237

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Advances in Neural Information Processing Systems*.

Yasmin, M., Haque, R., Kelleher, J., & Way, A. (2022). Domain-specific text generation for machine translation. *In Proc. 15th Biennial Conf. Association for Machine Translation in the Americas (AMTA 2022),* Vol. 1: Research Track (pp. 14–30).

Yin, W., & Schütze, H. (2015). Convolutional Neural Network for Paraphrase Identification. *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 901–911. https://doi.org/10.3115/v1/N15-1091

Yin, W., Schütze, H., Xiang, B., & Zhou, B. (2016). ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *Transactions of the Association for Computational Linguistics*, *4*, 259–272. https://doi.org/10.1162/tacl_a_00097

Yin To, X., Schalk, D., & Abenmacher, M., (2020). Modern Approaches in Natural Learning Processing (p 3 -8).

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending Against Neural Fake News. *Advances in Neural Information Processing Systems*, *32*.

Zhang, T., Lee, B., & Zhu, Q. (2019). Semantic measure of plagiarism using a hierarchical graph model. *Scientometrics*, *121*(1), 209–239. https://doi.org/10.1007/s11192-019-03204-x

Zhang, X., Rong, W., Liu, J., Tian, C., & Xiong, Z. (2017). Convolution neural network based syntactic and semantic aware paraphrase identification. *2017 International Joint Conference on Neural Networks (IJCNN)*, 2158–2163. https://doi.org/10.1109/IJCNN.2017.7966116

Zhou, C., Qiu, C., & Acuna, D. E. (2022). *Paraphrase Identification with Deep Learning: A Review of Datasets and Methods* (No. arXiv:2212.06933). arXiv. http://arxiv.org/abs/2212.06933

Zhou, J., & Bhat, S. (2021). Paraphrase Generation: A Survey of the State of the Art. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 5075–5086. https://doi.org/10.18653/v1/2021.emnlp-main.414

APPENDIX A

Details of section 6.3.1: Human Evaluation

The evaluation was conducted by six postgraduate researchers at Durham University, where English is the primary language of instruction. All evaluators were highly proficient in written English, with two being postdoctoral researchers in computer science and the remaining four in the final years of their PhD programmes: two in English, one in computer science, and one in Statistic. The evaluators were between 28 and 33 years old, comprising three men and three women. Prior to the evaluation, all participants provided informed consent and were given a clear explanation of the task requirements, along with illustrative examples to serve as training and ensure consistency in their judgments. Although not domain experts in psychology, their advanced level of education, disciplinary training, and language proficiency were deemed sufficient for assessing semantic similarity and coherence in paraphrased paragraphs. The text covers cognitive theory, clinical disorders, developmental processes, social/psychosocial factors.

Details of section 7.2.3: Human Evaluation

The evaluation was carried out by twelve undergraduate students in the final year of their studies at Durham University, where English is the primary language of instruction. The participants represented a range of disciplinary backgrounds, including four from computer science, five from statistics, and three from English. The group comprised seven men and five women. Before beginning the evaluation, all participants provided informed consent and received a clear explanation of the task requirements, together with illustrative examples to serve as training and to promote consistency in their judgments. The samples spanned a range of social science domains: anthropology focused on human cultural evolution and anthropological theory; economics addressed banking, finance, and regional economic development; archaeology examined material culture and labour in past societies; sociology explored language, social identity, and social movements; and management considered labour relations and organizational practices.