UNIVERSITY OF SHEFFIELD

DOCTORAL THESIS

Monolingual Plagiarism Detection and Paraphrase Type Identification

Author:
Faisal Alvi

Supervisors:
Dr. Mark Stevenson
Dr. Paul Clough

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Natural Language Processing Research Group
Department of Computer Science

August 3, 2020

Declaration of Authorship

I, Faisal Alvi, declare that this thesis titled, "Monolingual Plagiarism Detection and Paraphrase Type Identification" and the work presented in it are my own. I confirm that:

- This work was done mainly while in candidature for a research degree at the University of Sheffield.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at the University of Sheffield or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Acknowledgements

All praise is to Almighty God by whose grace good deeds are completed.

I would like to express my sincere thanks to my supervisor Dr. Mark Stevenson and co-supervisor Dr. Paul Clough for all the support, encouragement and guidance provided towards the completion of this PhD. Their feedback and advice has provided me with valuable insight and growth not only during the course of the PhD, but also towards a career in research.

I would also like to thank the PhD Panel members, Dr. Siobhan North and Dr. Mark Hepple for the fruitful panel meetings and the helpful comments throughout the course of this work.

Finally, thanks is due to my wife Mahjabeen and children Sarah and Haseeb for their love, support and encouragement during the course of this PhD.

UNIVERSITY OF SHEFFIELD

Abstract

Faculty of Engineering
Department of Computer Science

Doctor of Philosophy

Monolingual Plagiarism Detection and Paraphrase Type Identification

by Faisal Alvi

Over the past couple of decades research in plagiarism detection has gained immense importance due to the widespread availability of information and the ease with which it can be copied. In this PhD thesis, we address research in plagiarism detection from two perspectives: monolingual plagiarism detection and paraphrase type identification.

From a plagiarism detection perspective, we propose a cascade of approaches for detecting plagiarism in various obfuscation types in the PAN corpora. In particular, our methods for detecting summary obfuscation produce a plagdet score of 0.698, which is much higher as compared to the current best performing methods. We then extend the range of obfuscation types by creating a corpus of additional plagiarism types using retold versions of short stories. Obfuscation types such as homoglyph substitution, which score an unusually low score using current approaches are explored further and two approaches are proposed to address this type of technical disguise.

From a paraphrase type identification viewpoint, we propose methods to identify same polarity substitutions, which are the most frequent paraphrase type used in plagiarism. We propose a phased approach based on matching contexts and utilizing word embedding similarity scores to identify both contextual and non-contextual same polarity substitutions. Results of our approach give F_1 scores of 0.584 for the P4P corpus and 0.565 for the MSRP-A corpus, which are much higher than methods currently in use for paraphrase plagiarism detection. From a research perspective, identification of same polarity substitutions is a novel problem in plagiarism detection research.

Research from both these perspectives has the potential to enhance the current generation of plagiarism detection systems. This can be achieved by not only incorporating improved detection methods for various obfuscation types, but also by introducing paraphrase type identification in plagiarism detection systems.

The following publications were produced during the course of this thesis: (Alvi, Stevenson, and Clough, 2014), (Alvi, Stevenson, and Clough, 2015), (Alvi, Stevenson, and Clough, 2017)

List of Figures

1.1 Structure of the Thesis Chapters 5

2.1 Taxonomy of the well-known Forms of Text Reuse (Potthast, 2011) . . 13
2.2 Forms of Plagiarism (Maurer, Kappe, and Zaka, 2006) 16
2.3 Taxonomy of Plagiarism Types (Alzahrani, Salim, and Abraham, 2012) 17
2.4 White Box Diagram for (a) Extrinsic Plagiarism Detection, and (a) Intrinsic Plagiarism Detection (Alzahrani, Salim, and Abraham, 2012) . . 21
2.5 Classification of Lexical (non-NLP) Plagiarism Detection Approaches . 23
2.6 Smith Waterman Matrix Used for Alignment of Passages (Glinos, 2014) 25
2.7 The Construction of a Fingerprint F_d of a Document d. (Potthast, 2011) 28
2.8 Examples of Word n-grams and Skip n-grams (Guthrie et al., 2006) . . 29
2.9 Text Preprocessing Techniques for Plagiarism Detection (Ceska and Fox, 2009) 35
2.10 Stopword n-grams for Plagiarism Detection (Stamatatos, 2011) 37
2.11 (Stanford) Dependency Relations and Tree for a Sentence 41
2.12 Synset Comparison Using Jaccard Coefficient in WordNet 44
2.13 Example of Similarity Detection using Fuzzy Semantic Methods (Alzahrani and Salim, 2010) 46
2.14 A Description of Synonym Substitution (Bhagat and Hovy, 2013) . . . 50
2.15 A Description of Same Polarity Substitution (Barrón-Cedeño et al., 2013) 51
2.16 A List of Paraphrase Types with their Frequency of Occurrence (Barrón-Cedeño et al., 2013) 52
2.17 An Example of a Paraphrase Cast with Matching Left and Right contexts (Grigonyte et al., 2010) 54
2.18 Word Embedding Vector Space corresponding to the Analogy "*man* is to *king* as *woman* is to *queen*" 55
2.19 Part of ConceptNet Numberbatch Word Vector Matrix 56

3.1 CORPUSVIEWER: Passage Alignment in the PAN-2013 Training Corpus showing the No-obfuscation Plagiarism Type 61
3.2 Block Diagram of the Cascade Framework 63

3.3 Placement of String Tiles (size 3 or higher) within Various Obfuscation Types . . . 65
3.4 Criteria for Selecting Obfuscation Type in the Cascade of Approaches . 67
3.5 Classes of Matching Pairs . . . 68
3.6 A Comparison of Plagdet Scores for (a) No Obfuscation and (b) Random Obfuscation . . . 73
3.7 A Comparison of Plagdet Scores for (c) Translation Obfuscation and (d) Summary Obfuscation . . . 74

4.1 Ferret Trigram Similarity of Suspicious Document 00014 with Source Document 00011 . . . 81
4.2 Example of Textual Similarity in Story Retellings . . . 82
4.3 Plagdet Scores for (a) Story Retelling, (b) Synonym Replacement, and Character Substitution (c) (low) and (d) (high) . . . 88
4.4 Precision and Recall Scores for (a) Story Retelling, (b) Synonym Replacement, and Character Substitution (c) (low) and (d) (high) . . . 89

5.1 Visually Confusable characters for 'p' from the Unicode List of Confusables . . . 94
5.2 View of Source and Plagiarised Text Obfuscated using Homoglyphs . . 95
5.3 Block Diagram for Plagiarism Detection using the List of Confusables 96
5.4 Homoglyph Matrix with sim_h values . . . 97
5.5 Precision, Recall and Plagdet Scores for the complete dataset using Normalized Hamming Distance Similarity . . . 98
5.6 Precision, Recall and Plagdet Scores for each obfuscation type for various values of sim_h . . . 100

6.1 Block Diagram showing Paraphrase Type Identification added to Extrinsic Plagiarism Detection (Textual Alignment) Module . . . 106
6.2 Paraphrase Types in Source and Paraphrased Snippets (P4P Corpus, 17312–313) . . . 107
6.3 Block Diagram of the Three Stage System to detect Same Polarity Substitutions . . . 110
6.4 (a) Constituent Sentence Pairs in Snippets 16488-89, (b) Matrix of Pairwise Sentence Similarity Values, (c) Bipartite Graph showing Maximal Matching . . . 112
6.5 Scoring (Similarity) Matrix Constructed Using the Smith Waterman Algorithm for Snippet Pair 17732-33 in the P4P Corpus . . . 114
6.6 Possible Mismatch of Identical Text Fragments . . . 114
6.7 Word Similarity Matrix for Snippet Pair 17614-15 in the P4P Corpus . . 115

6.8 Word Similarity Matrix (with similarity values) for the Snippet Pair 16952-53 116
6.9 Annotation Guidelines showing Bifurcation for the RTE-2006 Dataset (Brockett, 2007) 117
6.10 Bifurcation of Same Polarity Substitution based on Word Embedding Similarity Score (Snippet Pair 17530-31, P4P Corpus) 117

List of Tables

2.1 A Mapping Between Paraphrase Types from Two Works 51

3.1 Strategies for Merging Various Cases 70
3.2 Plagdet Scores for Selected Approaches on the PAN-2014 Test Corpus for each obfuscation type 71

4.1 Statistics of Passage Sizes in the Corpus (characters) (*Sizes of no plagiarism are none since plagiarised passage sizes are zero) 79
4.2 Example of Synonym Substitution 84
4.3 Letters ‘a’ and ‘e’ with their Cyrillic Equivalents 84

5.1 Plagdet Scores by Replacing Homoglyphs 98
5.2 Plagdet Scores using the Normalized Hamming Distance for $sim_h = 0.45$ 99

6.1 Precision, Recall and F_1 scores for the P4P corpus using various approaches 120
6.2 Precision, Recall and F_1 scores for the MSRP-A corpus using various approaches 120

Contents

Declaration of Authorship iii

Abstract vii

1 Introduction 1
- 1.1 Brief Overview of Topics . . . 1
- 1.2 Background . . . 3
- 1.3 Structure and Topics of the Thesis . . . 4
- 1.4 Research Objectives . . . 7
- 1.5 Main findings of the Research . . . 8
- 1.6 Research Contributions (Publications) . . . 8

2 Literature Review 11
- 2.1 Introduction . . . 11
- 2.2 Text Reuse and Plagiarism . . . 11
  - 2.2.1 Text Reuse and Plagiarism . . . 12
  - 2.2.2 Plagiarism Prevalence and Attitudes . . . 13
  - 2.2.3 Forms of Plagiarism . . . 15
  - 2.2.4 Plagiarism Types: Different Views . . . 18
  - 2.2.5 Section Summary . . . 19
- 2.3 Plagiarism Detection . . . 19
  - 2.3.1 Extrinsic Plagiarism Detection . . . 20
  - 2.3.2 Intrinsic Plagiarism Detection . . . 20
  - 2.3.3 Monolingual and Cross-Lingual Plagiarism Detection . . . 21
- 2.4 Lexical Approaches for Plagiarism Detection . . . 22
  - 2.4.1 Exhaustive String Search . . . 22
  - 2.4.2 String and Sequence Alignment Based Methods . . . 22
  - 2.4.3 Vector Space Model . . . 26
  - 2.4.4 Fingerprinting . . . 27
  - 2.4.5 N-Gram Overlap . . . 29
  - 2.4.6 Other non-NLP Methods . . . 31
  - 2.4.7 Comparison of Various Methods . . . 32
  - 2.4.8 Section Summary . . . 32

2.5 Natural Language Processing Techniques for Plagiarism Detection . . 33
2.5.1 Natural Language Processing 33
2.5.2 Text Preprocessing 34
2.5.3 Stopword *n*-grams and Stopword Removal 36
2.5.4 Stemming and Lemmatization 37
2.5.5 Other Preprocessing Operations 39
2.5.6 POS Tagging 39
2.5.7 Syntactic Methods 40
2.5.8 Syntactic Parsing for Plagiarism Detection 40
2.5.9 Semantic Methods 42
2.5.10 Synonym Recognition Using WordNet 43
2.5.11 Word Generalization 45
2.5.12 Fuzzy Semantic Methods 46
2.5.13 Section Summary 47
2.6 Paraphrase Typologies 47
2.6.1 Overview of Textual Transformations 48
2.6.2 Earlier Works 48
2.6.3 What is a paraphrase? 49
2.6.4 Plagiarism Meets Paraphrasing 50
2.6.5 A Mapping Between the Two Paraphrase-Type Lists 51
2.6.6 The P4P (Plagiarism for Paraphrase) Corpus 52
2.7 Methodologies for Paraphrase Type Identification 53
2.7.1 Distributional Hypothesis 53
2.7.2 Automatic Synonym Detection 53
2.7.3 Vector Representations of Words (Word Embeddings) 55
Pretrained Word Embeddings 56
2.8 Conclusion 57

3 Cascade of Approaches for Plagiarism Detection **59**
3.1 Introduction 60
3.2 Overview of the Proposed Approach 61
3.2.1 Brief Description of the Framework 62
3.2.2 Seeding 62
3.2.3 Extension 62
3.3 Detailed Description of Approaches 66
3.3.1 No Obfuscation 66
3.3.2 Summary Obfuscation 66
3.3.3 Random and Translation Obfuscation 68
3.3.4 Strategy for Merging 69

3.3.5 Filtering . . . 69
3.4 Results and Discussion . . . 71
3.5 Conclusion . . . 75

4 Generation of Additional Obfuscation Types **77**
4.1 Introduction . . . 78
4.2 Corpus Construction . . . 79
4.2.1 No plagiarism . . . 80
4.2.2 Story Retelling . . . 80
4.2.3 Synonym Replacement . . . 83
4.2.4 Character Substitution . . . 84
4.3 Reviews of the Corpus . . . 85
4.4 Results and Discussion . . . 86
4.4.1 Story Retelling . . . 87
4.4.2 Synonym Replacement . . . 87
4.4.3 Character Substitution . . . 87
4.5 Conclusion . . . 90

5 Plagiarism Detection in Texts Obfuscated by Homoglyphs **91**
5.1 Introduction . . . 92
5.2 Resources . . . 93
5.2.1 The Unicode List of Confusable Characters . . . 93
5.2.2 Evaluation Dataset . . . 94
5.3 Approaches . . . 95
5.3.1 Using the List of Unicode Confusables . . . 95
5.3.2 Normalized Hamming Distance between words . . . 96
5.4 Results and Discussion . . . 97
5.4.1 Substitution using the Unicode List of Confusables . . . 97
5.4.2 Normalized Hamming Distance . . . 99
5.5 Conclusions and Future Work . . . 101

6 Same Polarity Substitution Detection **103**
6.1 Introduction . . . 104
6.1.1 Problem Description . . . 106
6.2 Datasets and Measurement Parameters . . . 106
6.2.1 Datasets Used . . . 107
6.2.2 Measurement Parameters . . . 108
6.3 Proposed Approach . . . 109
6.3.1 Preprocessing . . . 111
6.3.2 Detection of same polarity Substitutions . . . 112

Context-Based Same-Polarity Substitutions: 112
Word Embedding Based Same-Polarity Substitutions: 115
6.3.3 Merging Same-Polarity Substitutions 118
6.3.4 Filtering 118
6.4 Results and Discussion 119
6.4.1 Observations 122
6.5 Conclusions and Future Work 122

7 Conclusions and Future Work **123**
7.1 Findings of the Thesis 124
7.1.1 Impact of Research 125
7.2 Future Work 125
7.3 Conclusion 126

References **127**

Chapter 1

Introduction

Plagiarism and its detection have been one of the active areas of research in Natural Language Processing since the past two decades. The field has progressed rapidly from identifying verbatim types of text reuse, to detecting more involved forms of textual modification. Likewise, the incidence of plagiarism has also increased manifold with the availability of digital platforms. With these advances, it is imperative to develop better automated plagiarism detection systems that not only detect more sophisticated forms of obfuscation but also assist the human evaluator in readily identifying cases of plagiarism or text reuse.

Researchers have identified several forms of obfuscation in plagiarism, which include copy-and-paste obfuscation, paraphrase plagiarism, translation plagiarism and disguised plagiarism. From a plagiarism detection perspective, a large variety of plagiarism detection techniques have been proposed in the literature which include the use of several classes of natural language processing methods. These techniques have shown tremendous success in detecting plagiarism for text that has been plagiarised using various types of obfuscation. Despite these advances, plagiarism detection when the text has been paraphrased continues to be an active area of research (Carmona et al., 2018).

In this chapter we provide an introduction to the thesis by starting with a brief overview of the relevant concepts and terms involved. This is followed by a description of the research goals and and an overview of the research contributions. We conclude this chapter by outlining the contents of each chapter, followed by the research outcomes in the form of publications.

1.1 Brief Overview of Topics

In this section we give a brief overview of text reuse and plagiarism, various forms of plagiarism, plagiarism detection methodologies and paraphrase types used in plagiarism. This overview provides the required background for outlining the research objectives in the next section and the research contributions of the thesis in the subsequent sections.

- **Text Reuse** and **Plagiarism** are two terms that are sometimes used interchangeably, however there are some differences between them. Text Reuse is the act of reusing text either in its original or modified form, while plagiarism is reusing text without attribution and may constitute an unacceptable act. The incidence of plagiarism has risen quite rapidly with the availability of digital platforms and the easy availability of copy-and-paste functions. It is in this context that the need for automated plagiarism detection arises significantly. While automated plagiarism detection systems (e.g. Turnitin) help identify reused sections of text, it is up to a human evaluator to decide whether an incidence of text reuse constitutes plagiarism or not. Both text reuse and plagiarism are discussed in more detail in Section 2.2.1, with prevalence of plagiarism and its forms discussed in Sections 2.2.2 and 2.2.3 respectively.

- **Automated Plagiarism Detection** is the process of automatically detecting plagiarism with computational methods both from within the source and the suspicious documents. In **extrinsic plagiarism detection**, an external collection of source documents is available with which a given suspicious document can be compared, to detect sections of plagiarised text. However, in **intrinsic plagiarism detection**, no collection of source documents is available, and therefore methods detecting changes in writing style are used for plagiarism detection. **Cross-lingual plagiarism detection** refers to plagiarism detection when the source and suspicious documents are in different languages. An overview of these broad categories of plagiarism detection is given in Section 2.3.

- **Plagiarism Detection Techniques** can be broadly classified into lexical and language processing based approaches. Lexical or language independent plagiarism detection methods are based on finding patterns of textual similarity between two texts. Such methods do not involve language processing and therefore treat data as symbols rather than as linguistic units. Lexical techniques for plagiarism detection include string and sequence alignment based methods, vector-space model based methods, fingerprinting based methods and n-gram based methods. These methods are discussed in Section 2.4. Natural Language Processing (NLP) based plagiarism detection methods primarily involve language processing. These methods require knowledge of the language and its terms, for example, lemmatization requires the knowledge of the base form and the derived forms of a given word. Text preprocessing, lemmatization, syntactic parsing and semantic methods are just some of the method families included in these types of methods. Some of these methods may also involve mechanisms from lexical method families, e.g. use of n-grams in stopword n-grams. An overview of semantic methods is discussed in Section 2.5.

- **Paraphrase Types** or Textual Transformations are various types of rewrites undertaken when modifying reused text. Among some of the more important paraphrase types (as identified by the researchers) are **synonym substitution**, **sentence restructuring** and **verb modifications**. From a plagiarism perspective, several researchers have identified same polarity substitution (which is a broader form of synonym substitution) as the most frequent paraphrase type in paraphrased, plagiarised text. Section 2.6 gives an overview of paraphrase types, while Section 2.7 provides general concepts useful for the detection of same polarity substitutions.

1.2 Background

Plagiarism and its detection have witnessed a vast amount research over the past several years. This can be observed from several literature surveys, e.g. (Alzahrani, Salim, and Abraham, 2012; Eisa, Salim, and Alzahrani, 2015; Foltýnek, Meuschke, and Gipp, 2019; Kanjirangat and Gupta, 2016; Meuschke and Gipp, 2013) which give an overview of state-of-the-art detection methods and tools employed for plagiarism detection. Based on the current research trends, plagiarism detection research can be broadly classified into three key areas:

1. Plagiarism Detection Methods,
2. Plagiarism Detection Systems, and
3. Plagiarism Detection and Prevention Policies.

Plagiarism detection methods have shown tremendous research progress in the past decade. The PAN Series of Evaluation Labs on plagiarism detection from 2009–2014 (Potthast et al., 2009, 2014) has motivated research on plagiarism detection methods. This is evident from the high plagdet scores achieved for various obfuscation types in PAN-2013 and subsequently PAN-2014 (Potthast et al., 2013, 2014). Several classes of methods are now available identified as lexical (e.g. word and character n-gram based), syntax-based (e.g. POS tagging, morphology) and semantics-based (e.g. WordNet, word embeddings) methods that have shown promising results in detecting several types of obfuscation. However, researchers have observed that a combination of diverse classes of methods as an ensemble (of methods) achieves improved results (Agirre et al., 2016; Franco-Salvador et al., 2016) as compared to the usage of a single class of methods (Foltýnek, Meuschke, and Gipp, 2019).

From a plagiarism detection perspective, the best performing system in the PAN-2014 plagiarism detection (text alignment) task (Sanchez-Perez, Sidorov, and Gelbukh, 2014, 2015) used an adaptive approach, whereby the values of the parameters were adjusted corresponding to each obfuscation type. In this thesis, we further

extend the idea of using a combination of methods by proposing a cascade of plagiarism detection approaches in Chapter 3.

As research in detection approaches showed significant progress towards 2014-15, research in plagiarism detection turned towards expanding the scope of obfuscation types in PAN-2015 (Potthast et al., 2015). Chapter 4 describes our proposed obfuscation types of story retelling, synonym substitution and UTF character (homolgyph) substitution. Plagiarism detection on these proposed obfuscation types showed that existing detection approaches successfully detect plagiarism for most obfuscation types, except in the case of homoglyph substitution where most of the detection approaches reported low plagdet scores.

Homoglyph substitution (Gillam, Marinuzzi, and Ioannou, 2010) is a form of technical disguise wherein Latin script characters are replaced with visually identical characters from some other script. Due to a change in the computer representation of text, homoglyph substitution turns out to be a very effective technique in masking plagiarism. In this thesis, we address methods for detecting plagiarism in texts obfuscated by homoglyph substitution in Chapter 5.

The link between paraphrasing and plagiarism has been well established with paraphrasing being considered as a form of plagiarism. Vila, Martí, Rodríguez, et al. (2014) and Barrón-Cedeño et al. (2013) have proposed a set of paraphrase types emulating textual rewrites that a plagiarist might undertake in order to obfuscate plagiarism. In this sense, developing methods to identify paraphrase types in text that has already been detected as plagiarised, moves research in plagiarism detection a step further as compared to the reporting of plagdet scores between source and suspicious documents. Chapter 6 describes our proposed methods for the detection of same polarity substitutions – the most frequently reported paraphrase type in plagiarised text.

1.3 Structure and Topics of the Thesis

This thesis presents research work on research problems in monolingual plagiarism detection and paraphrase type identification as highlighted briefly in the preceding background section. It can broadly be divided into two parts, described as follows:

Part One: Monolingual Plagiarism Detection: The first part on monolingual plagiarism detection can be considered as a sequence of Chapters consisting of Chapters 3, 4 and 5, which address related research problems in plagiarism detection. Chapter 3 proposes a cascade of detection approaches for plagiarism detection on standard obfuscation types within the PAN datasets. Chapter 4 builds upon these

obfuscation types by introducing additional obfuscation types which test the effectiveness of existing collection of approaches for plagiarism detection. Finally, Chapter 5 addresses plagiarism detection in texts obfuscated by homoglyph substitution, which is an effective form of technical disguise in evading plagiarism detection.

Part Two: Paraphrase Type Identification: The second part of the thesis addresses research on paraphrase type identification which consists of Chapter 6. In particular, the chapter focuses on the detection of same polarity substitutions (the most frequently occurring paraphrase type) in text that has already been identified as plagiarised. We propose an integrated framework of both classical and contemporary research methods for the detection of same polarity substitutions. Research presented within this chapter can prove to be helpful by providing valuable information to a human evaluator in making an informed decision about the actual occcurrence of plagiarism.

A schematic diagram showing the relatedness of the various chapters within the thesis is shown in Figure 1.1.

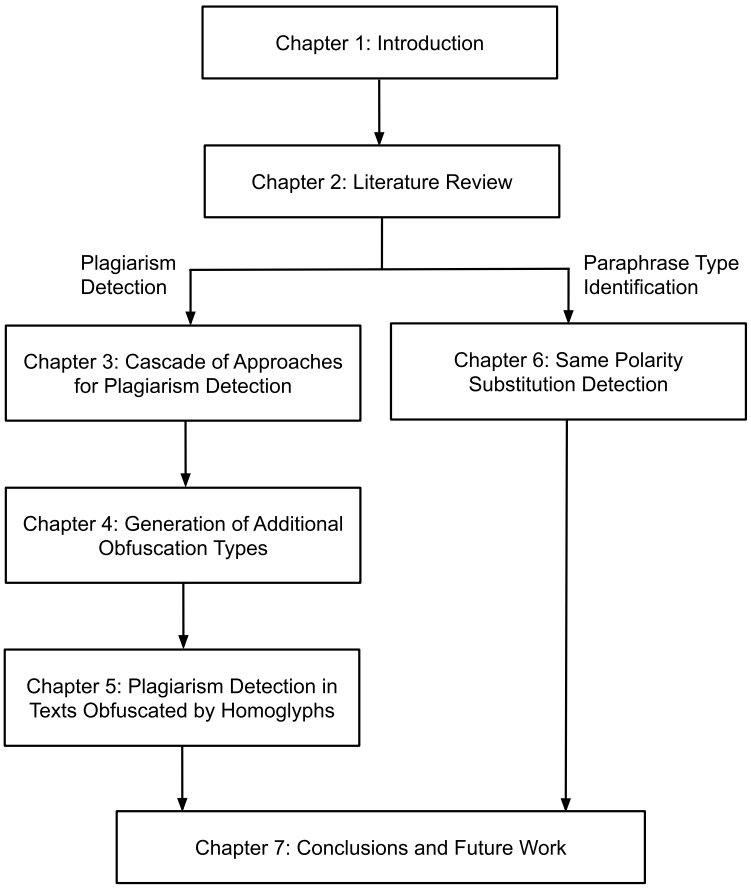


FIGURE 1.1: Structure of the Thesis Chapters

Outline of the Chapters

In this section we provide a detailed description of the content of chapters.

- **Chapter 1**: (Introduction) provides an introduction to the thesis with a description of terms and background. This is followed by the thesis structure and an outline of the chapters. The chapter concludes by presenting the research objectives, main findings of the thesis and research publications produced during the course of this work.

- **Chapter 2**: (Literature Review) provides a comprehensive description of the literature for plagiarism detection that is pertinent to the research work described within the thesis. Topics covered include: plagiarism and text reuse (definition), forms of plagiarism, plagiarism detection techniques, paraphrasing and paraphrase types, distributional hypothesis and word embeddings.

- **Chapter 3**: (Cascade of Detection Approaches), describes a proposed cascade of detection approaches for plagiarism detection, thereby extending the idea of using a multitude of approaches. Our proposed model classifies obfuscation types based upon criteria for seed lengths and seed distances and presents a modular structure whereby a customized approach for detecting plagiarism can be used corresponding to each obfuscation type. Plagdet scores from our model give improved results for both (verbatim) no-obfuscation and summary obfuscation as compared to the current state-of-the-art results for the PAN-2014 dataset.

- **Chapter 4:** (Generation of Additional Obfuscation Types) focuses on creating additional obfuscation types by creating a corpus of simulated cases of plagiarism using retold versions of short stories. The primary motivation for the creation of this dataset is to test the effectiveness of available plagiarism detection approaches on newer obfuscation types. We propose three obfuscation types called story retelling, synonym substitution and UTF-character (homoglyph) substitution. Test results from the best performing PAN approaches show that while plagiarism is successfully detected for story retelling and synonym substitution types, plagdet scores for plagiarism detection for homoglyph substitution are low.

- **Chapter 5:** (Plagiarism Detection in Texts Obfuscated by Homoglyphs) presents two different approaches for detecting plagiarism when text has been subjected to technical disguise in the form of homoglyph obfuscation. We propose two different approaches consisting of: (a) replacement of homoglyphs with ASCII characters using the Unicode list of confusable characters, and (b) using the normalized hamming distance for approximate string matching. The

differences between these two approaches are discussed with appropriate use-case scenarios identified for each of these proposed approaches.

- **Chapter 6**: (Same Polarity Substitution Detection) advances research towards a distinct but interconnected problem of identifying paraphrase types in text that has already been detected as plagiarised. In particular, this chapter focuses on investigating methods to identify the most frequent paraphrase type encountered in plagiarised text, i.e. same polarity substitution. We use the Smith Waterman Algorithm and ConceptNet Numberbatch word embedding similarity measures to detect contextual and non-contextual same polarity substitutions, respectively, using an integrated framework. Results from these approaches show that a combined use of these two approaches gives improved results as compared to the use of either method individually.

- **Chapter 7**: (Conclusions and Future Work) finally concludes this thesis by giving a summary of the research contributions carried out within this thesis. We also identify applications of this research into enhancing the current generation of plagiarism detection systems and propose future work that has the potential to build upon this research.

1.4 Research Objectives

This thesis addresses research in plagiarism and its detection from several perspectives which involve detection strategies, obfuscation types and paraphrasing. The overall research aim is to investigate research questions that contribute towards enhancing the functionality of plagiarism detection systems. Towards this aim, the following research objectives are to be undertaken in this thesis.

1. To develop a modular structure for plagiarism detection that extends the mechanism of using a combination of detection approaches.

2. To develop a plagiarism detection approach for summary obfuscation that utilizes the methodology for automatic summarization.

3. To extend the range of obfuscation types for plagiarism detection and identify types that report a low plagdet score using existing techniques.

4. To propose approaches for identifying plagiarism in text obfuscated by homoglyphs and compare these approaches with respect to their use-case scenarios.

5. To develop novel methods for identifying same polarity substitutions in paraphrased, plagiarised text using contextual matches and word embedding similarity measures.

1.5 Main findings of the Research

Corresponding to the research objectives, the main findings of the research are presented here briefly as follows:

1. The cascade of plagiarism detection approaches has a modular structure, thereby giving the flexibility of integrating several customized approaches corresponding to each obfuscation type (Chapter 3).

2. Plagiarism using summary obfuscation can be detected effectively by using the distance metric between matching string tiles within the source and suspicious documents. (Chapter 3)

3. Obfuscation types corresponding to technical disguise, such as homoglyph substitution can evade plagiarism detection techniques, since these change the underlying computational representation of text (Chapter 4).

4. Plagiarism in texts obfuscated using homoglyph substitution can be effectively detected by character replacement using the Unicode list of confusable characters as well as using approximate string matching using normalized hamming distance (Chapter 5).

5. Both contextual methods (Smith Waterman Algorithm) and non-contextual methods (ConceptNet Numberbatch word embeddings similarity) are effective in detecting contextual and non-contextual same polarity substitutions from paraphrased, plagiarised text (Chapter 6).

6. An integrated, phased framework for detecting same polarity substitutions consisting of contextual and non-contextual detection methods produces improved results, in contrast to use of either class of methods individually (Chapter 6).

1.6 Research Contributions (Publications)

The following research works have been published from this thesis as follows:

1. Alvi, Faisal, Mark Stevenson, and Paul Clough (2014). "Hashing and Merging Heuristics for Text Reuse Detection – Notebook for PAN at CLEF 2014". In: Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014, pp. 939–946

2. Alvi, Faisal, Mark Stevenson, and Paul Clough (2015). "The Short Stories Corpus –Notebook for PAN 2015". In: Working Notes for CLEF 2015 Conference, Toulouse, France, September 08-11, 2015.

3. Alvi, Faisal, Mark R. Stevenson, and Paul Clough (2017). "Plagiarism Detection in Texts Obfuscated with Homoglyphs". In: European Conference on Information Retrieval. Springer, pp. 669–675, 2017.

Chapter 2

Literature Review

2.1 Introduction

Plagiarism and its detection has been a well researched area since the early 1990s. In this chapter we present the background literature relevant to the research work in this thesis. We begin by highlighting several definitions of 'text reuse' and 'plagiarism' and discuss their differences. We also present an overview of various surveys about the prevalence of plagiarism in academia and scientific work. Various forms of plagiarism, such as verbatim copy-and-paste and paraphrase plagiarism, are surveyed next, identified in the literature as plagiarism taxonomies.

From a plagiarism detection perspective, an overview of the two main categories of plagiarism detection, i.e., extrinsic and intrinsic plagiarism detection is presented. This is followed by an extensive review of various types of plagiarism detection techniques including both lexical (non-NLP) and language processing based (NLP) techniques. This is followed by a survey of paraphrase types, which discusses the various text rewrite operations commonly used in paraphrase plagiarism. We conclude this chapter by presenting an overview of the mechanisms useful for the detection of paraphrase types in plagiarised text.

The rest of this chapter is organized as follows. In Section 2.2, the definitions of text reuse and plagiarism are presented, along with the prevalence and forms of plagiarism. In Section 2.3, an overview of basic types of plagiarism detection is given. Sections 2.4 and Section 2.5 present an extensive coverage of lexical and language processing based techniques for plagiarism detection. Finally, Sections 2.6 describe paraphrase typologies, while Section 2.7 presents methods to detect paraphrase types.

2.2 Text Reuse and Plagiarism

This section provides a brief introduction to text reuse and plagiarism. We state the definitions of text reuse and plagiarism and identify the relationship between them. We also present an overview of surveys and experimental work providing

numerical estimates on the percentage of students involved in plagiarism as well as attitudes towards it. Furthermore, we also state the forms of plagiarism in text given by researchers from different viewpoints.

2.2.1 Text Reuse and Plagiarism

Text reuse is the process in which content from existing texts is used to write a new text, either in its exact or modified form. A formal definition of 'text reuse' is stated as follows (Clough, 2001, p. 27):

> *"The process by which literal content from a single source document is reused in the creation of a target document. Content is reused in the same context either word-for-word (verbatim) or paraphrased (rewritten)."*

Another definition of text reuse appears as (Bendersky and Croft, 2009, p. 1)

> *"a wide scope of text transformations, including exact recapitulations, loose restatements of the information from the previous sources, and reports that have little in common except for the subject matter"*.

From a literary perspective, text reuse has been defined as (Romanello, Berra, and Trachsel, 2014)

> *"Text reuse is the meaningful reiteration of text, usually beyond the simple repetition of common language. Such a broad concept can naturally be understood at different levels and studied in a large variety of contexts."*

From these definitions, we can infer that text reuse is the deliberate[1] reuse of text in order to create another text, either in its exact or modified form.

Several forms of text reuse exist, for example journalistic text reuse, collaborative authoring and plagiarism. Among these, plagiarism, which is *an unacceptable form of text reuse* (Clough et al., 2002), is defined in various sources as:

1. *when the work of someone else is reproduced without acknowledging the source, this is known as plagiarism* (Clough, 2000),
2. *taking the ideas, writings, or inventions of another and representing them as your own* (Liddell, 2003),
3. *the passing off of another person's work as if it were one's own, by claiming credit for something that was actually done by someone else* (Maurer, Kappe, and Zaka, 2006),

[1] Cryptomnesia (inadvertent plagiarism) is the unconscious reuse of text (Brown and Murphy, 1989)

4. *the use of ideas and/or words from sources without giving due acknowledgement as imposed by academic*[2] *principles* (Meuschke and Gipp, 2013).

The above research works give definitions of text reuse and plagiarism from various perspectives. From these definitions it can be inferred that plagiarism can be considered as text reuse without due acknowledgment of the source.

Figure 2.1 shows a classification of several types of text reuse. It can be observed that plagiarism is a type of text reuse; furthermore paraphrasing can be considered as a type of plagiarism (Potthast, 2011).

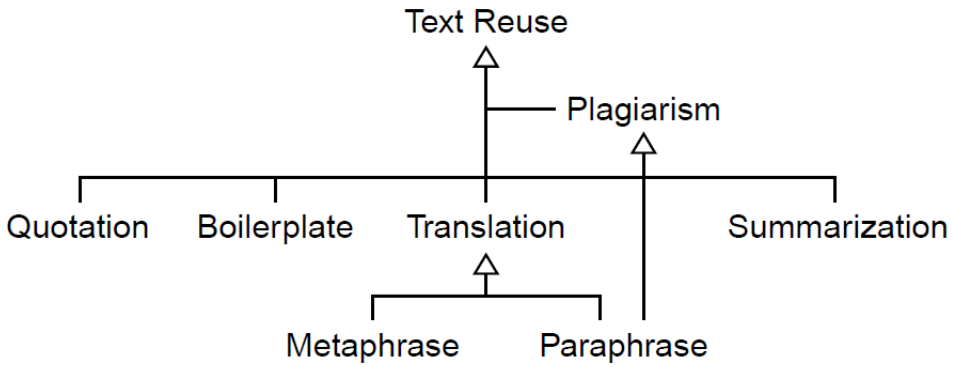


FIGURE 2.1: Taxonomy of the well-known Forms of Text Reuse (Potthast, 2011)

With the availability of large collections of documents on the Internet, text can be copied and reused with just a few keystrokes (Weber-Wulff, 2014, p. 7). The purpose of Automated Plagiarism Detection Systems is to greatly ease the effort in finding instances of text reuse and thereby assist human evaluators in determining whether an instance of text reuse can be considered as a case of plagiarism. In order to achieve this goal, most plagiarism detection techniques as well as plagiarism detection tools provide a similarity score between the source and suspicious text.

2.2.2 Plagiarism Prevalence and Attitudes

Instances of plagiarism have been reported in several areas, for example, academic research, political speeches and advertising (Bailey, 2017). As an example, in a 'massive plagiarism scandal', 200 South Korean Professors were charged for plagiarizing republished content as original books under their names (Iyengar, 2015).

From an academic perspective, surveys and experiments have been also been conducted that not only provide an estimate regarding the percentage of students engaging in plagiarism, but also give an idea of students' and educators' perceptions towards what constitutes plagiarism.

[2]Definition of academic plagiarism

Surveys and Experimental Work

A number of surveys, reports and experiments have been carried out to assess the prevalence, impact and attitudes towards plagiarism within the academic community. Here we give a brief outline of a selection of these works from both academic and scientific communities within the last decade.

- In a survey of 238 doctoral students and 92 faculty members in health sciences by Ewing et al. (2019), surveyed students and faculty reported that there was a significant prevalence of plagiarism among the student population, but it was not being self-reported. Furthermore, awareness of plagiarism was higher in online students than campus-based students, while campus-based students reported a higher incidence of unauthorized collaboration on assignments.

- In an empirical study, Kauffman and Young (2015) found that 79.5% of students engaged in digital plagiarism when copy-and-paste function was made available to them in a digital environment. The researchers remarked that writers attempted to complete their writing tasks with a minimal amount of effort using time saving strategies.

- In another survey, Dias and Bastos (2014) provided an assessment of attitudes towards plagiarism by getting the opinion of 170 teachers and 334 students from seven European countries. According to the assessment, although teachers and students both agreed that plagiarism is on the rise with the ease of copying from the Internet, teachers state the cause as "*students' lack of skills*", while students cite the reason as the "*pressure to get good grades*", among others.

- In a technical report by the Pew Research Center, Parker, Lenhart, and Moore (2011) give a detailed survey of college presidents' opinions in the United States regarding the prevalence of plagiarism. Some of the findings in this report are: (a) 55% of the college presidents think that *plagiarism is on the rise*, and (b) among these presidents, 89% think that "*computers and internet have played a major role in this trend*".

- With regards to plagiarism in scientific literature, a study by Baždarić et al. (2012) found that out of 754 manuscripts submitted to the Croatian Medical Journal in 2009-2010, 11% were found to cases of plagiarism; among these 3% of the cases were of self-plagiarism.

In addition to these, a large number of surveys have been conducted earlier. A comprehensive overview of these surveys appears in (Barrón-Cedeño, 2012). It can be concluded from these surveys that the percentage of students engaging in plagiarism is significant and that plagiarism is on the rise. In response, there is a need

for (a) educating students and researchers towards plagiarism and its consequences, and (b) use of automated plagiarism detection systems to detect instances of plagiarism.

2.2.3 Forms of Plagiarism

Researchers have given several classifications of plagiarism. In this section we review the various types of plagiarism as identified in selected papers in the literature from Martin (1994) to Foltýnek, Meuschke, and Gipp (2019). Plagiarism in text can be either word-for-word (verbatim) using a simple copy-and-paste operation on a computer, or by reusing modified text. Modifications in text can be further categorised based on the extent to which text is changed, as well as the methodology employed for the same.

In an earlier paper, Martin (1994) has given several forms of plagiarism including the following.

1. *Word-for-word plagiarism*: This refers to verbatim copying without acknowledgment.
2. *Paraphrasing*: This refers to changing the source words.
3. *Secondary sources plagiarism*: This refers to material obtained from secondary sources without verifying the content in the original source.
4. *Source-form plagiarism*: This refers to use of an argument form from a source without due acknowledgement.
5. *Idea plagiarism*: This refers to using ideas from another source, without using the actual source words.
6. *Authorship plagiarism*: This refers to "*putting one's name on someone else's work*".

Among these, word-for-word plagiarism and paraphrasing can be considered as direct forms of text reuse. Some of the other forms stated above can be considered as errors in citing the correct source, for example, source-form plagiarism and secondary sources plagiarism. While these forms are not exhaustive, nevertheless they provide an insight into the forms of plagiarism described in the earlier literature.

Plagiarism can also be classified based on the intent of the plagiarist. This approach has been taken by Maurer, Kappe, and Zaka (2006) where they describe the following four broad categories of plagiarism.

1. *Accidental*: This type of plagiarism may be the result of a lack of understanding of institutional policies and/or referencing style.

2. *Unintentional*: One might unconsciously reuse someone else's ideas or expressions without a deliberate intent, e.g., when being influenced by someone's ideas.

3. *Intentional*: This is a deliberate attempt to copy and reproduce someone else's work in its original or modified form, without referring to the original author.

4. *Self plagiarism*: This is the act of using previous work by oneself again without referring to previous work being used.

Apart from these broad categories, Maurer, Kappe, and Zaka (2006) also state various forms of plagiarism, which include copy-and-paste, paraphrasing and idea plagiarism among others. A list of these types is given in Figure 2.2. This list also includes translated plagiarism, which refers to cross-lingual plagiarism discussed in the next section.

- copy-paste: copying word to word textual contents.
- idea plagiarism: using similar concept or opinion which is not common knowledge.
- paraphrasing: changing grammar, similar meaning words, re-ordering sentences in original work. Or restating same contents in different words.
- artistic plagiarism: presenting some one else's work using different media, such as text, images, voice or video.
- code plagiarism: using program code, algorithms, classes, or functions without permission or reference.
- forgotten or expired links to resources: addition of quotations or reference marks but failing to provide information or up-to-date links to sources.
- no proper use of quotation marks: failing to identify exact parts of borrowed contents.
- misinformation of references: adding references to incorrect or non existing original sources.
- translated plagiarism: cross language content translation and use without reference to original work.

FIGURE 2.2: Forms of Plagiarism (Maurer, Kappe, and Zaka, 2006)

In another work, Joy et al. (2009) provide a taxonomy of plagiarism aimed towards computer science students which consists of 6 categories subdivided into 23 sub-categories. For the category of plagiarism and copying, they have produced subcategories of copying, paraphrasing and self-plagiarism. This faceted taxonomy is useful for educating students on what constitutes plagiarism.

In a later work, Alzahrani, Salim, and Abraham (2012) give a taxonomy of plagiarism types broadly classified into two basic forms: literal plagiarism and intelligent plagiarism. Literal Plagiarism consists of exact copy, near copy and modified copy; it is largely composed of an exact replica of the original document with minor changes at most. Intelligent plagiarism consists of text manipulation, translation and idea adoption with further sub-categories as shown in Figure 2.3.

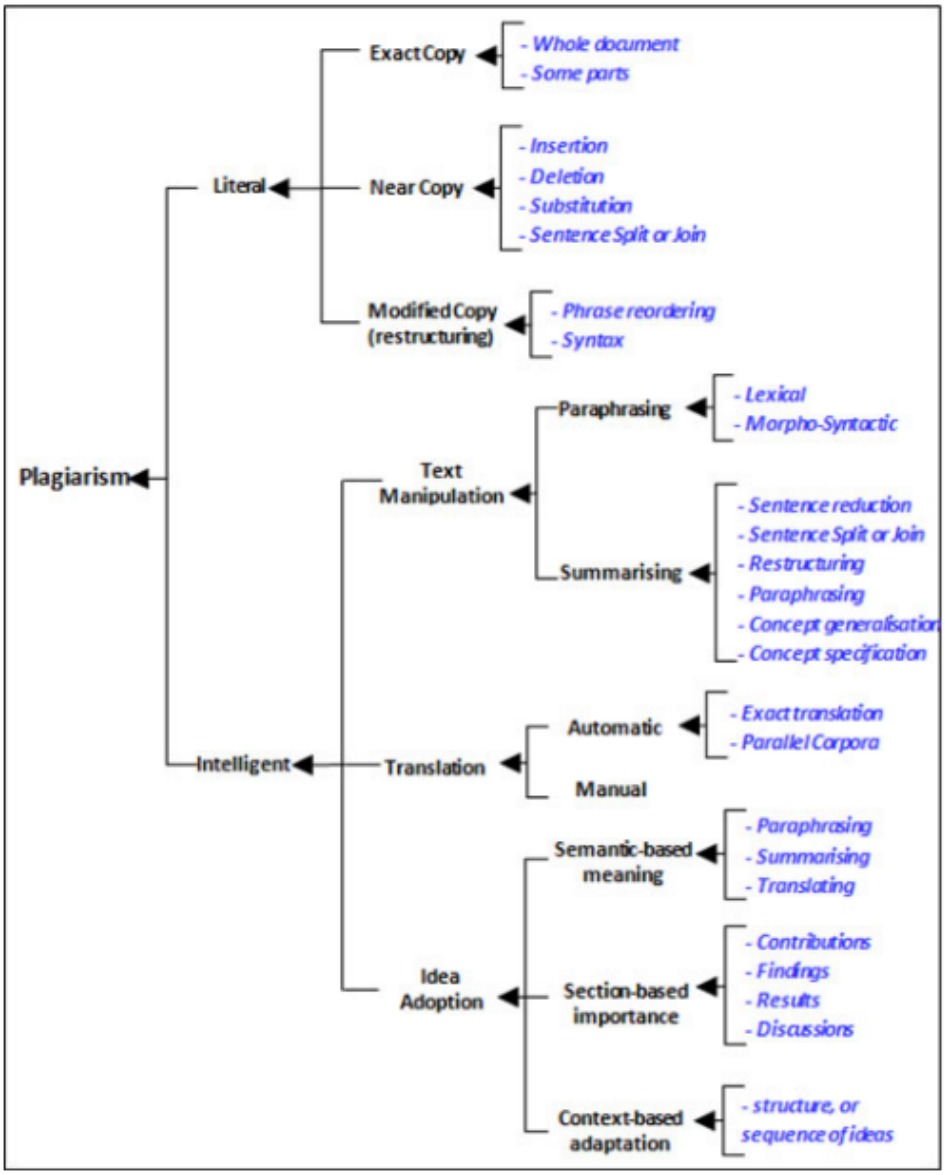


FIGURE 2.3: Taxonomy of Plagiarism Types (Alzahrani, Salim, and Abraham, 2012)

This taxonomy is an extension of an earlier taxonomy presented in Zu Eissen and Stein (2006). Alzahrani, Salim, and Abraham (2012) provide several examples from the Corpus of Plagiarised Short Answers (Clough and Stevenson, 2011) to support their classification. Figure 2.3 gives an illustrated version of this taxonomy. It can be seen that some of the deeper types of modifications like paraphrasing, summarization and translation come under the heading of intelligent plagiarism.

In another research work Meuschke and Gipp (2013) have also given four basic forms of academic plagiarism, namely, *literal*, *disguised*, *translated* and *self-plagiarism*. This typology was updated in recently in their work (Foltýnek, Meuschke, and Gipp, 2019) with the following forms of plagiarism.

1. *Characters Preserving Plagiarism*: This consists of literal (copy-and-paste) plagiarism with possible modifications.
2. *Syntax Preserving Plagiarism*: This is further sub-divided into technical disguise and synonym substitution.
3. *Semantics Preserving Plagiarism*: This consists of translation plagiarism and paraphrase plagiarism.
4. *Idea Preserving Plagiarism*: This consists of structural plagiarism and the use of ideas and concepts only.
5. *Ghostwriting* This describes utilizing the services of another person or entity to produce original text.

Technical Disguise: In the above classification a particular mention must be made of technical disguise (Heather, 2010), which is exploiting the weaknesses of plagiarism detection software to evade detection. For example, the insertion of look-alike UTF characters (also known as 'homoglyphs') in lieu of standard ASCII characters is an example of technical disguise. To a human, texts with technical disguise might appear exactly the same, but for a computer these appear as completely different collections of characters.

2.2.4 Plagiarism Types: Different Views

From the above classifications we observe that researchers have classified forms of plagiarism from a multitude of different perspectives. These are summarised as follows.

Method used: Most identified forms of plagiarism represent this perspective. For example, copy-and-paste, paraphrasing, summarization and translation (Potthast et al., 2013, 2014) from source texts, all refer to the method used by the plagiarist. Broadly speaking, this could be stated as comprising literal and intelligent plagiarism (Alzahrani, Salim, and Abraham, 2012).

Source used: This viewpoint is represented by self-plagiarism (Meuschke and Gipp, 2013) and translated plagiarism, where the emphasis is on the type of the source document used. For example, self-plagiarism refers to the source of text used and not the method adopted. Similarly, translation of source texts can also be considered to be a part of this category since translation implies that the source text was in a different language (except for back translation).

Intent of the Plagiarist: This primarily describes the intent of the plagiarist (Maurer, Kappe, and Zaka, 2006), for example whether the plagiarism was accidental (a result of not knowing correct citation rules), unintentional (for example as a result of links to sources being expired) or intentional (as in copy-paste).

2.2.5 Section Summary

Text reuse is the re-application of text to produce new text, while plagiarism is a certain kind of unacceptable text reuse. Surveys and experimental work from educational institutions show that, with the widespread availability of digital content, plagiarism has increased manifold. Several forms of plagiarism have been identified in the text based on the methodology used, the source of texts and the intent of the plagiarist.

In the next section, we discuss plagiarism detection and state the associated terminology and the types of plagiarism detection. We also focus on widely used methods and techniques for monolingual extrinsic plagiarism detection.

2.3 Plagiarism Detection

With the widespread prevalence of plagiarism as outlined in the previous section, rises the need for plagiarism detection too. However, given the large amount of information available on the Web, manual detection of plagiarism is infeasible (Clough, 2000). In this respect automated plagiarism detection tools aid in finding textual similarity between pairs of documents, thereby helping in finding cases of plagiarism. This point is clearly outlined in (Barrón-Cedeño, 2012, p. 2) as follows.

> *Determining whether a text fragment has been plagiarised, or even reused, is a decision that concerns to human judge. Automatic systems are aimed at assisting such an expert to uncover a potential case and, if possible, to take an informed decision. Claiming that a person is culpable of plagiarism is the responsibility of the expert, not of a computer program.*

Therefore plagiarism detection tools should actually be considered as text reuse detection tools which find instances of textual similarity within a pair of documents. The final decision on whether a reported similarity score represents a genuine case of plagiarism rests with a human evaluator (McKeever, 2006). This is because, "*it [the decision] requires a careful consideration of these annotated matches by a person to determine which, if any, constitute plagiarism*" (Mphahlele and McKenna, 2019, p. 1084). In this respect automated plagiarism detection tools greatly assist the human expert by speeding up the process of finding text fragments that may have been reused.

In addition to detecting cases of text reuse, automated plagiarism detection systems also discourage people from being tempted to plagiarise (Pupovac, Bilic-Zulle, and Petrovecki, 2008).

Traditionally plagiarism detection has been divided into two types: extrinsic or external plagiarism detection and intrinsic plagiarism detection. We state each of them as follows.

2.3.1 Extrinsic Plagiarism Detection

Extrinsic plagiarism detection (Potthast et al., 2009) refers to detection of plagiarism in a given suspicious (query) document from an external collection of source documents. This kind of plagiarism detection might be necessary when, for example, a given document is suspected of being copied from a possible collection of source documents.

The mechanism for detecting plagiarism through extrinsic plagiarism detection is stated as follows.

1. In the first step, a small number of candidate documents is retrieved from a large collection for similarity computation with the suspicious document. This is done by representing the document collection using some kind of heuristic model, e.g., the Vector Space Model. Using such a representation, similarity between the query (suspicious) document and the collection of source documents is computed. The resulting collection of source documents having a similarity measure above or equal to a given threshold is called the *candidate collection* of documents.

2. Secondly, exhaustive pairwise document analysis is carried out to find similar components of text between the suspicious document and the candidate collection of source documents. Passages of texts that are found to be similar are marked as having been plagiarised or copied.

3. Finally some knowledge-based post processing is done to merge the detected similar components into larger sections. Usually small matches or false positives are eliminated at this stage and finally, the suspected plagiarised portions of text are reported.

Figure 2.4 (a) shows the white box design for extrinsic plagiarism detection (Alzahrani, Salim, and Abraham, 2012).

2.3.2 Intrinsic Plagiarism Detection

Intrinsic plagiarism detection (Stein, Lipka, and Prettenhofer, 2011) happens when no collection of source documents is available to compare against a given suspicious document. A possible scenario when a document may be considered to be plagiarised is when it has inconsistencies in writing style, but the investigating authority might not have any source collection of documents to compare the suspicious document against with.

Intrinsic plagiarism detection relies on stylometric features and abrupt changes in style in order to find plagiarised sections and proceeds as follows.

1. Initially, the given document is divided into various segments. The segment size depends on the document under consideration and could be as small as a few sentences to as large as complete sections of a document.

2. Secondly, stylometric extraction is carried out on these segments for all the segments. This includes extracting stylometric features of the document, for example text statistics, syntactic features, and semantic features.

3. Finally, stylometric analysis is carried out to find abrupt changes or differences in features. Sections of the document that have relatively different stylometry compared to the rest of the document are marked as plagiarised sections and reported.

Figure 2.4 (b) shows a white box design for intrinsic plagiarism detection.

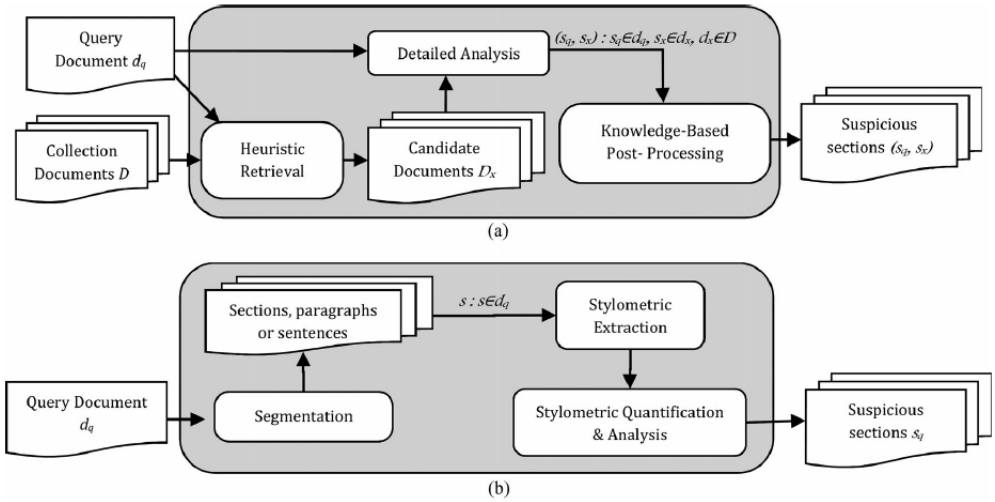


FIGURE 2.4: White Box Diagram for (a) Extrinsic Plagiarism Detection, and (a) Intrinsic Plagiarism Detection (Alzahrani, Salim, and Abraham, 2012)

2.3.3 Monolingual and Cross-Lingual Plagiarism Detection

Extrinsic plagiarism detection can further be classified based on the languages used in the suspicious and candidate documents. **Monolingual plagiarism detection** refers to plagiarism detection when the source and suspicious documents are in the same language. A large body of research in monolingual extrinsic plagiarism detection is in English (Alzahrani, Salim, and Abraham, 2012). **Cross-lingual plagiarism detection** (Ceska, Toman, and Jezek, 2008) refers to plagiarism detection when the languages in the suspicious and the candidate documents are different (e.g., English and Spanish), thereby implying translated plagiarism. This happens when source text has been taken from another language and plagiarised by translating into the language of the suspicious document.

2.4 Lexical Approaches for Plagiarism Detection

In this section we provide an overview of the methods used for plagiarism detection that require little or no language processing (except for some preprocessing to extract language units such as characters or words). This class of methods treats language units (such as characters or words) as symbols, hence the term *lexical detection methods* (Foltýnek, Meuschke, and Gipp, 2019) can be used to describe these methods. These methods can also be considered as *non-NLP* methods in line with the classification used by Chong (2013).

Figure 2.5 provides a classification of the methods surveyed in this section. It is important to state that each of the following methods actually refers to a family of approaches since there are many variants and modified versions for each method.

2.4.1 Exhaustive String Search

For plagiarism detection, the most basic method is exhaustive string search, i.e., given a collection of source documents and a suspicious document, we find all instances of strings that occur in both source and suspicious documents. In the worst case this could take $O(mn)$ time where n and m are the sizes of the source and suspicious documents respectively (Cormen et al., 2009). Several methods can be used to speed up this search such as searching for entire words or sentences, use of trie based methods as well as hashing, but this becomes rapidly inefficient for larger document collections.

2.4.2 String and Sequence Alignment Based Methods

String comparison and sequence alignment rely on aligning matching parts of two strings (Navarro, 2001). DNA Sequence Alignment Algorithms for bioinformatics have also been used for plagiarism detection (Irving, 2004). Here we describe four different method families that have found widespread use for plagiarism detection.

Edit Distance or Levenshtein Distance

Edit Distance or the Levenshtein distance (Levenshtein, 1966) denotes the number of operations required to transform one string of characters into another. Using dynamic programming this algorithm finds the minimum cost operations required. Wagner and Fischer (1974) also use edit distance in the context of error-correction. The basic operations considered in edit distance are the following.

- *Insertion*: This represents the operation of inserting a character into a string, e.g., 'set' changes to 'se**a**t' after inserting an 'a'.

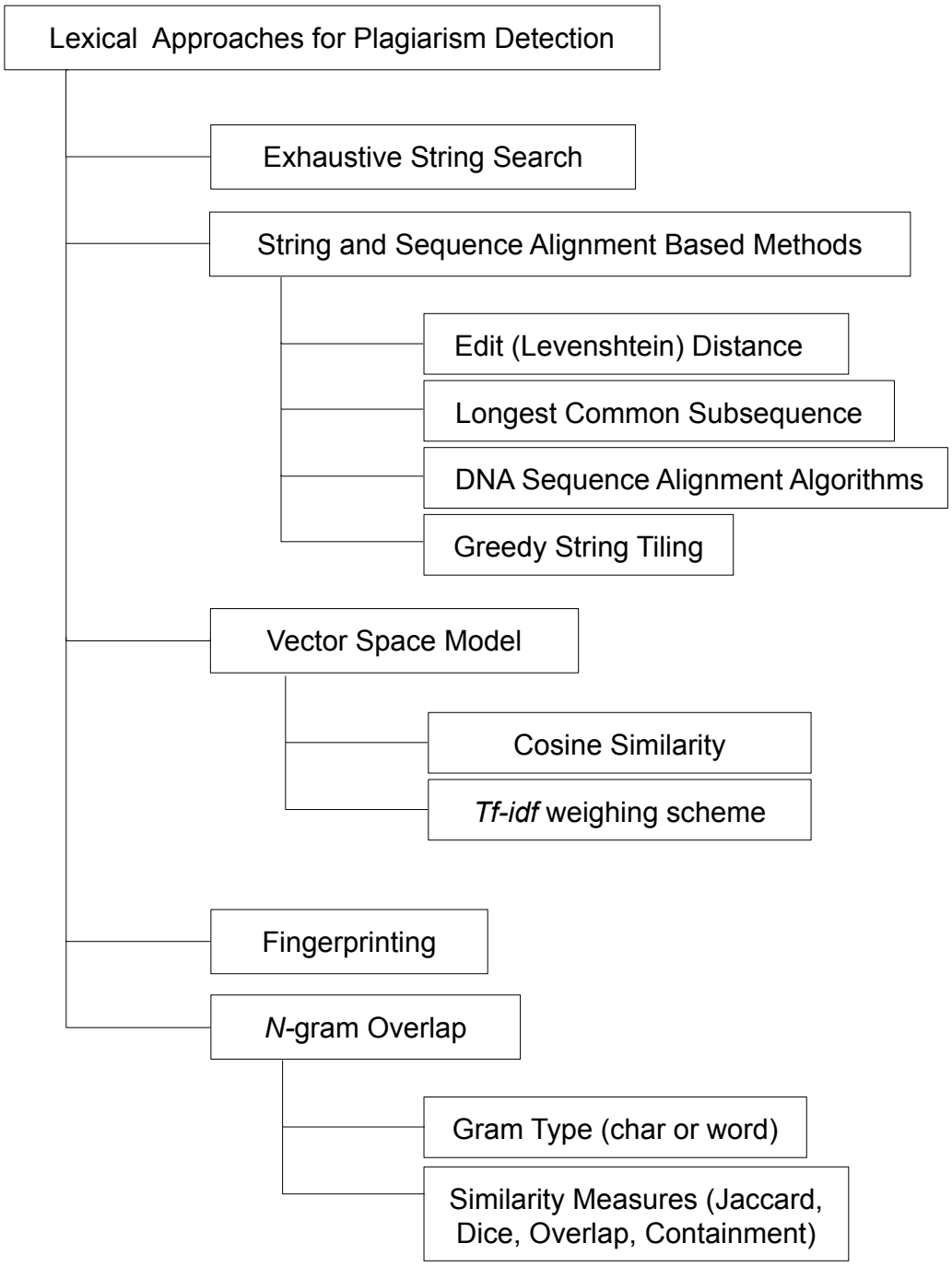


FIGURE 2.5: Classification of Lexical (non-NLP) Plagiarism Detection Approaches

- *Deletion*: This represents the operation of deleting a character from a string.
- *Substitution*: This represents the operation of substituting a character and is equivalent to an insert-delete or in-del.

Typically the cost of substitution is twice the cost of an individual in-del operation. However variants of the edit distance also employ variable costs for these three operations.

Edit distance can also be extended to find the editing cost of sentences where the tokens represent the words and the insertion, deletion and substitution operations are carried out for words. Dolan, Quirk, and Brockett (2004) have used the notion of *sentence edit distance* in the construction of Microsoft Research Paraphrase Corpus. Guégan and Hernandez (2006) have also used sentence edit distance in the context of finding textual parallelism.

The operation of transposition can also be added to the set of operations of edit distance to have a larger set of operations. Sankoff and Kruskal (1983) have suggested the use of these four basic operations, i.e., insertion, deletion, substitution and transposition. Although these are applicable in the context of biological sequence alignment, they have also been used for plagiarism detection.

A number of research works have used edit distance for plagiarism detection, for example (Su et al., 2008), (Rani and Singh, 2018) and (Carmona et al., 2018). In particular, Carmona et al. (2018) introduce the notion of semantically informed distance measures, including edit distance for paraphrase plagiarism detection. These distance measures are based on the Jaccard similarity measure and Levenshtein edit distance by considering WordNet and Word2Vec based similarity measures as cost functions, and have shown improved results for paraphrase plagiarism detection.

Longest Common Subsequence

The longest common subsequence problem (Apostolico and Guerra, 1987; Gorbenko and Popov, 2012) primarily deals with finding the longest common substring within a collection of strings. In terms of plagiarism detection, this implies finding the longest common substring between two strings, which are representations of the source and the suspicious documents. Solutions for finding the longest common subsequence typically involve dynamic programming. The normalized value of the length of the longest common substring can be used as a similarity measure, where normalization is carried out by dividing the length of the *lcs* by the length of one of the strings.

Clough and Stevenson (2011) have used the normalized *lcs* measure as a means to distinguish between various levels of document revision in the Corpus of Plagiarised Short Answers. Chong, Specia, and Mitkov (2010) have also used the *lcs* measure for plagiarism detection.

Bioinformatics Sequence Alignment Algorithms

Bioinformatics Algorithms for sequence alignment have also been used for plagiarism detection, since these can be applied to find matching sequences within two strings. The *Smith Waterman Algorithm* (Smith and Waterman, 1981) for local sequence alignment has been used for plagiarism detection extensively. Local sequence alignment means that the algorithm finds alignments of all possible lengths from two sequences in order to generate the optimal alignment. This is in contrast to global alignment where the alignment of the entire sequence length is carried out. The *Needleman-Wunsch Algorithm* (Needleman and Wunsch, 1970) is also used for sequence alignment however it uses global alignment only instead of local alignment.

Glinos (2014) have used a variant of the Smith Waterman Algorithm for finding similar passages in large document collections. Figure 2.6 shows the dynamic programming matrix for their variant of the Smith Waterman Algorithm using dynamic programming for plagiarism detection. Su et al. (2008) use both the Smith Waterman Algorithm and the Edit Distance for plagiarism detection. Their approach is an example of using two string comparison techniques to yield an efficient plagiarism detection method.

		This	essay	discusses	Hamlet	's	famous	soliloquy	in	relation	to	the	major	themes	of	the	play	.	tempus	fugit
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
This	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
article	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
discusses	0	0	0	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
the	0	0	0	2	2	1	0	0	0	0	0	2	1	0	0	2	1	0	0	0
famous	0	0	0	1	1	1	3	2	1	0	0	1	1	0	0	1	1	0	0	0
Hamlet	0	0	0	0	3	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0
monologue	0	0	0	0	2	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0
of	0	0	0	0	1	1	1	0	3	2	2	1	0	0	2	1	0	0	0	0
the	0	0	0	0	0	0	0	0	2	2	1	4	3	2	1	4	3	2	1	0
main	0	0	0	0	0	0	0	0	1	1	1	3	3	2	1	3	3	2	2	0
themes	0	0	0	0	0	0	0	0	0	0	0	2	2	5	4	3	2	2	1	0
of	0	0	0	0	0	0	0	0	2	1	2	1	1	4	7	6	5	4	3	2
the	0	0	0	0	0	0	0	0	1	1	1	4	3	3	6	9	8	7	6	5
game	0	0	0	0	0	0	0	0	0	0	0	3	3	2	5	8	8	7	6	5
.	0	0	0	0	0	0	0	0	0	0	0	2	2	2	4	7	7	10	9	8
carpe	0	0	0	0	0	0	0	0	0	0	0	1	1	1	3	6	6	9	9	8
diem	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5	5	8	8	8

FIGURE 2.6: Smith Waterman Matrix Used for Alignment of Passages (Glinos, 2014)

Greedy String Tiling

Greedy String Tiling was first proposed by Wise (1993, 1995) which can detect movement of 'tiles' or blocks efficiently. The worst case complexity of the algorithm is $O(n^3)$. However, optimizations using the Running Karp Rabin (RKR) Algorithm (Karp and Rabin, 1987) can make it run in linear time in the average case.

RKR-GST makes use of tiles or blocks that correspond to substrings of maximal length between two documents. RKR-GST runs in two stages: *scan-pattern* and *mark-arrays*. Scan-pattern finds all string matches of maximal length while mark-arrays is used to test whether a maximal match has not already been covered by a previous match.

The GST Algorithm was originally proposed for biological sequence matching, however it has been used for plagiarism detection in JPlag (Prechelt, Malpohl, and Philippsen, 2000), detecting text reuse in a set of journalistic articles (Clough et al., 2002), and in for plagiarism detection in PAN-2011 corpus (Nawab, Stevenson, and Clough, 2011).

2.4.3 Vector Space Model

Representation

Documents in the vector space model (Salton, Wong, and Yang, 1975) are represented as vectors in an N-dimensional space where N is the number of terms occurring in an entire collection of source and suspicious documents. A term in this sense may represent a single word, a word n-gram or an entire sentence. In the case of large documents an entire paragraph or a section may also be considered as a term (Dreher, 2007). In this sense a document is simply a collection of terms without resorting to the order of the terms appearing in the document. More formally, the k^{th} document in a collection of N documents is represented as a sparse vector:

$$\vec{d_k} = (t_{1,k}, t_{2,k}, ..., t_{N,k}) \tag{2.1}$$

Similarity Measures

Cosine similarity measure is the most common one used to find similarity between two documents. More formally this is the dot product,

$$sim(d_1, d_2) = \frac{\vec{d_1} \cdot \vec{d_2}}{|d_1| \times |d_2|} = \frac{\sum_{i=1}^{n} d_{1,i} \times d_{2,i}}{\sqrt{(\sum_{i=1}^{n} d_{1,i}^2) \times (\sum_{i=1}^{n} d_{2,i}^2)}} \tag{2.2}$$

Methodology

Once represented as vectors, documents that rank higher in terms of the cosine similarity measure with the suspicious document are ranked the highest and are returned back in decreasing order of similarity. The cosine similarity measure can give a range of values 0.0 to 1.0, with higher values closer to 1.0 implying higher similarity and corresponding documents as a possible source of plagiarism.

Variation

The *tf-idf* scheme (term frequency-inverse document frequency) (Baeza-Yates and Ribeiro-Neto, 2011) is one of the popular weighting schemes used to assign weights to terms within a document collection. The *term frequency* assigns a weight to a term based on its frequency in the document. The *inverse document frequency* assigns a higher weight to a term appearing in fewer documents. Together these two schemes identify terms that occur frequently within a document, while occurring infrequently within a collection. tf-idf forms the basis of many plagiarism detection schemes using the Vector Space Model.

Application

Hoad and Zobel (2003) have used VSM to detect duplicate and near-duplicate reports. Similarly, Zechner et al. (2009) have used VSM for both external (extrinsic) and intrinsic plagiarism detection. Within the PAN plagiarism detection competition, Sanchez-Perez, Sidorov, and Gelbukh (2014) have used a tf-idf like scheme based on Vector Space Model in PAN-2014.

2.4.4 Fingerprinting

Representation

In fingerprinting (Schleimer, Wilkerson, and Aiken, 2003), a document is represented by a collection of substrings which are then converted to shingles or fingerprints. These fingerprints are unique integers and are obtained by applying a hash function to the collection of substrings.

Design Parameters

When designing a fingerprinting based solution, Hoad and Zobel (2003) propose the following four important design parameters for consideration:

1. the function used to generate a fingerprint,
2. the size of the substrings selected for fingerprint generation (fingerprint granularity),

3. the number of fingerprints used to represent the document (fingerprint resolution),
4. the selection strategy for selecting the fingerprints for document representation.

Usually hash functions are used for fingerprint generation as these are fast and their output (integers) is reproducible and easily compared. The number of substrings selected may be all (the entire document), or a small subset. Likewise, the size of the substrings being selected is important, as short substrings generate a lot of false (positive) matches, while longer substrings are susceptible to small changes in text. Finally, if a subset of substrings is to be selected for fingerprinting instead of the entire document, a strategy must be devised for deciding which substrings to select for representation.

Potthast (2011) has given a comprehensive overview of fingerprinting algorithms by proposing a framework that encompasses a large number of fingerprinting algorithms. He proposes four steps for the construction of a fingerprint, namely, representation, dimensionality reduction, quantization, and encoding. Among these steps, quantization is the step where the conversion of document strings into integers is carried out through the use of hash functions. Figure 2.7 gives an overview of these four steps in a process.

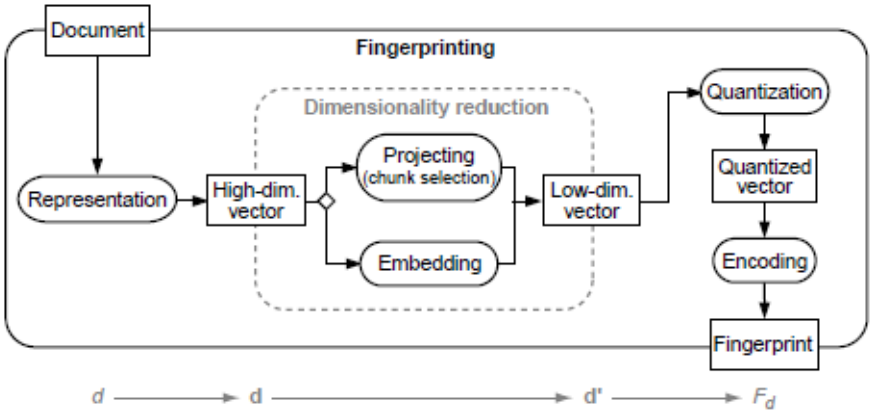


FIGURE 2.7: The Construction of a Fingerprint F_d of a Document d. (Potthast, 2011)

Similarity Measures

The fingerprints of the suspicious document are compared with those of the source document collection. For the similarity measure, the absolute number of matches can be used. A higher absolute number of matches implies likely plagiarism. The number of matches divided by the size of the source/suspicious document can also be used as a similarity measure, also known as the *containment* measure, as described in the next section.

Application

Seo and Croft (2008) have used fingerprinting towards local text reuse detection on the TREC newswire collection. Within the PAN Software competition for plagiarism detection, Grman and Ravas (2011) used fingerprinting for plagiarism detection in the PAN-2011 corpus. HaCohen-Kerner and Tayeb (2017) have applied fingerprinting to detect similarity in scientific papers.

2.4.5 N-Gram Overlap

Representation

An n-gram is a collection of n-tokens of a text, where a token might be a character, a word or even a sentence. This results in three different types of n-grams known as character n-grams, word n-grams and sentence n-grams in the literature. n-grams form a sliding window over a text, so that resulting n-grams are overlapping, or these might be distinct, non-overlapping collection of tokens. In terms of n-grams, a given text is represented entirely as a collection of its n-grams. Typically, 1-grams are known as unigrams, 2-grams as bigrams and 3-grams as trigrams respectively.

Skip n-grams

A k-skip n-gram (Guthrie et al., 2006) is a variation of the n-gram scheme where at most k of the tokens may be skipped. Figure 2.8 gives an example of various word n-grams and some associated k-skip n-grams for a given piece of text.

"*Insurgents killed in ongoing fighting.*"

Bi-grams = {insurgents killed, killed in, in ongoing, ongoing fighting}.
2-skip-bi-grams = {insurgents killed, insurgents in, insurgents ongoing, killed in, killed ongoing, killed fighting, in ongoing, in fighting, ongoing fighting}
Tri-grams = {insurgents killed in, killed in ongoing, in ongoing fighting}.
2-skip-tri-grams = {insurgents killed in, insurgents killed ongoing, insurgents killed fighting, insurgents in ongoing, insurgents in fighting, insurgents ongoing fighting, killed in ongoing, killed in fighting, killed ongoing fighting, in ongoing fighting}.

FIGURE 2.8: Examples of Word n-grams and Skip n-grams (Guthrie et al., 2006)

Similarity Measures

Once represented as a collection of n-grams, two texts may be compared and similarity computed using several similarity measures. The most raw form would be to find and compute the number of commonly occurring n-grams between the two texts. This however does not take into account the size of the source and suspicious documents and, therefore, a better similarity measure giving the ratio or percentage of similarity to the overall document size needs to be taken into account. Some of the more commonly used similarity measures are presented below.

1. The **Jaccard similarity coefficient** (Jaccard, 1912) is computed by dividing the common n-grams between two documents by the total number of n-grams between two documents. More formally, if $S(A,n)$ represents the number of n-grams in document A, and $S(B,n)$ represents the number of n-grams in document B, then the Jaccard similarity coefficient is given by

$$S_{jaccard} = \frac{|S(A,n) \bigcap S(B,n)|}{|S(A,n) \bigcup S(B,n)|} \tag{2.3}$$

2. The **Dice similarity coefficient** (Dice, 1945) is a variant of the Jaccard similarity coefficient and is given by

$$S_{dice} = 2 \times \frac{|S(A,n) \bigcap S(B,n)|}{|S(A,n) + S(B,n)|} \tag{2.4}$$

3. The **Overlap similarity coefficient** is also a variant of the Jaccard similarity coefficient and is given by

$$S_{overlap} = 2 \times \frac{|S(A,n) \bigcap S(B,n)|}{min(|S(A,n), |S(B,n)|)} \tag{2.5}$$

4. The **Containment similarity coefficient** is given by

$$S_{containment} = 2 \times \frac{|S(A,n) \bigcap S(B,n)|}{|S(A,n)|)} \tag{2.6}$$

Among the above, the first three similarity metrics (Jaccard, Dice and Overlap) are symmetric, while the Containment similarity metric is asymmetric, since it quantifies the degree of text (B) within a document (A). All of the above similarity metrics have values that range from 0.0 to 1.0, with a higher value signifying a greater degree of similarity.

Application

n-grams have been used widely for plagiarism detection, not only as a standalone technique but also in combination with other techniques. For example fingerprinting on n-grams is used widely. Clough and Stevenson (2011) have used the containment measure using n-grams for $n = 1, 2, 3, 4$ and 5 for detecting similarity in the Corpus of Plagiarised Short Answers. This corpus is a collection of answers to short questions by university students with various levels of simulated plagiarism. Barrón-Cedeño and Rosso (2009) have used word bigrams and trigrams for text reuse detection in the METER corpus. Palkovskii and Belov (2014) use several word n-gram variations for plagiarism detection in PAN-2014.

In addition skip n-grams have also been used for plagiarism detection by Chen, Yeh, and Ke (2010) using ROUGE Lin (2004) and WordNet. Their findings suggest that skip bigrams may be useful for detecting plagiarism when text is modified using simple operations like addition or deletion of words.

2.4.6 Other non-NLP Methods

Some of the other non-NLP methods used for plagiarism detection are briefly described here.

- Probabilistic plagiarism detection methods have been successfully used to reduce the search space needed for monolingual plagiarism detection (Barrón-Cedeño, Rosso, and Benedí, 2009). The Kullback-Leibler Divergence Model relies on the Kullback-Leibler distance (Kullback and Leibler, 1951) or KL_d, which calculates how different are two probability distributions P and Q. It is an asymmetric measure, with symmetric variants proposed as well for plagiarism detection.
- Structural Methods rely on document structure to detect plagiarism. In structural methods, the document structure is decomposed down from sections to subsections to paragraphs down to the lowest available unit. If the document structure of two documents is similar, then plagiarism can be suspected. Chow and Rahman (2009) have used block-specific tree structures for plagiarism detection on the Web.
- Citation Based Plagiarism Detection (Gipp, 2014; Meuschke, Gipp, and Breitinger, 2012) is another concept in the field of plagiarism detection. This relies on analysing the list of citations as well as their order of referencing and other citation information for detecting similarity in the citation analysis of two texts. Texts with significantly similar citation sequencing can be considered to have been plagiarised. This method is more suitable to scientific texts or texts with at least some citation information.

2.4.7 Comparison of Various Methods

In the previous sections, an overview of various non-NLP methods for plagiarism detection was presented. These methods find widespread use in plagiarism detection research, and give high detection scores. For example, from the PAN-2014 plagiarism detection results, it can be seen that approaches based on the Vector Space Model (Sanchez-Perez, Sidorov, and Gelbukh, 2014), Sequence Alignment Algorithm (Glinos, 2014) and *n*-gram comparison (Palkovskii and Belov, 2014) all produced high detection scores. However, each one of these approaches involved further modifications and customizations which complement the basic versions reviewed in the presented survey.

In particular, the approach by Sanchez-Perez, Sidorov, and Gelbukh (2014) used the vector space model with a tf-idf like weighting scheme considering sentences as documents. Glinos (2014) used the Smith Waterman Algorithm with recursive descent and matrix splicing procedures for plagiarism detection. Likewise, Palkovskii and Belov (2014) used "*contextual n-grams, surrounding context n-grams, named entity based n-grams, odd-even skip n-grams, functional words frame based n-grams*" for the detection of plagiarism.

However, the best performing approach by Sanchez-Perez, Sidorov, and Gelbukh (2014), updated in (Sanchez-Perez, Sidorov, and Gelbukh, 2015) has been particularly mentioned for its *adaptive* nature by Foltýnek, Meuschke, and Gipp (2019). This approach achieved the best scores by recognizing various forms of plagiarism based on the lengths of the passages, and subsequently setting parameters corresponding to each recognized obfuscation type. From a research perspective, this line of research (i.e., using an adaptive approach for each obfuscation type) merits further investigation.

2.4.8 Section Summary

In this section we defined plagiarism, plagiarism detection and provided definitions of extrinsic and intrinsic plagiarism detection. We also provided an outline of some of the most common non-NLP methods used for plagiarism detection including vector space model, string alignment, fingerprinting and *n*-gram overlap. Other non-NLP methods include probabilistic methods, citation based plagiarism detection and document structure analysis for plagiarism detection. These methods are fast, and often produce high detection rates, but involve little or no language processing. In the next section we present a survey of NLP methods for plagiarism detection which involve some kind of language processing for plagiarism detection.

2.5 Natural Language Processing Techniques for Plagiarism Detection

Natural language processing techniques (abbreviated henceforth as NLP techniques) have benefited a number of areas including plagiarism detection. In this section we present a host of NLP techniques which may form the whole or part of a given plagiarism detection method. Here, the question is, how do we consider a given technique as an NLP technique? Jurafsky and Martin (2008) provide an interesting example of the UNIX `wc` program. In terms of their description the UNIX `wc` program is a data processing application when used to count characters and lines. However, "*when it (`wc` program) is used to count the words in a file, it requires knowledge about what it means to be a word and thus becomes a language processing system*". This statement provides a useful guideline in classifying a given technique as an NLP technique. If a plagiarism detection method requires knowledge of the language under consideration, (such as tokenization of text into sentences and words or identification of parts of speech), it can be considered as an NLP technique.

From a plagiarism detection perspective, plagiarism detection techniques can be classified as lexical, syntactic and semantic (Foltýnek, Meuschke, and Gipp, 2019). In our classification of plagiarism detection approaches, we group together lexical approaches in Section 2.4, as these do not require language processing except at the stage of extraction of language units (characters or words). However, techniques requiring syntactic or semantic processing, i.e., knowledge of the language based on its structure or meaning are considered as NLP techniques described in this section.

2.5.1 Natural Language Processing

Natural Language Processing involves the understanding and processing of human language. More formally, "*Natural language processing is a theory-motivated range of computational techniques for the automatic analysis and representation of human language.*" (Cambria and White, 2014, p. 48). NLP techniques have benefited other fields for example machine translation (Somers, 1999), information retrieval (Manning, Raghavan, and Schütze, 2008) and sentiment analysis (Liu, 2012) among others. Earlier, Clough (2003b) motivated the use of NLP techniques in plagiarism detection and outlined possible improvements, later on also emphasised by Ceska and Fox (2009). Although researchers have used NLP techniques for plagiarism detection, however their use still remains underexplored (Chong, 2013).

Traditionally, Natural Language Processing largely consisted of classical notions of syntax, semantics and pragmatics, (Dale, 2010) also known as symbolic natural language processing. In contrast statistical natural language processing, also known as empirical natural language processing, is based on probabilistic models to learn

linguistic rules by running through large numbers of examples. In the words of Jackson and Moulinier (2007), "*Symbolic natural language processing tends to work top-down by imposing known grammatical patterns and meaning associations upon texts. Empirical natual language processing tends to work bottom-up from the texts themselves, looking for patterns and associations to model, some of which may not correspond to purely syntactic or semantic relationships.*" In the context of plagiarism detection, it is (mostly) the linguistic (symbolic) natural language processing that is meant by NLP techniques.

In the next few subsections we present an overview of NLP techniques used in plagiarism detection. As stated previously, we consider a technique as an NLP technique if it involves language processing. Another aspect to consider is that an NLP technique may form the whole or part of a plagiarism detection approach.

2.5.2 Text Preprocessing

Text preprocessing is "*the task of converting a raw text file, essentially a sequence of digital bits, into a well-defined sequence of linguistically meaningful units.*" (Palmer, 2010). Text preprocessing is the fundamental part of natural language processing, since it involves understanding of the constituent units of the natural language. In conventional natural language processing literature, text preprocessing consists of *segmentation* – i.e., breaking a large piece of text into segments, and *tokenization* – i.e., breaking up a unit of text into words or morphemes. Both segmentation and tokenization have been in use for plagiarism detection as part of text preprocessing. For example, segmentation is used as part of the framework for intrinsic plagiarism detection (Alzahrani, Salim, and Abraham, 2012). Likewise Chong, Specia, and Mitkov (2010) use segmentation and tokenization as text preprocessing. For segmentation they break the text into sentences, while tokenization is used to find word boundaries.

However, some authors use a broader definition of text preprocessing in plagiarism detection literature. This view is based on the idea that all operations done on text prior to actual similarity computation can be considered to be text preprocessing. For example, Ceska and Fox (2009) highlight the effects of several text preprocessing techniques on plagiarism detection. They list several preprocessing techniques in the context of plagiarism detection which include stopword removal, lemmatization, synonymy recognition, number replacement and word generalization. For stopword removal and lemmatization, they have used dictionaries, while WordNet has been employed for synonymy recognition and word generalization as shown in Figure 2.9. Among these techniques, synonymy recognition may be considered to be a semantic method, since it involves recognizing meanings of words, with some mechanisms used from traditional (non-NLP) techniques as well (e.g. *n*-grams).

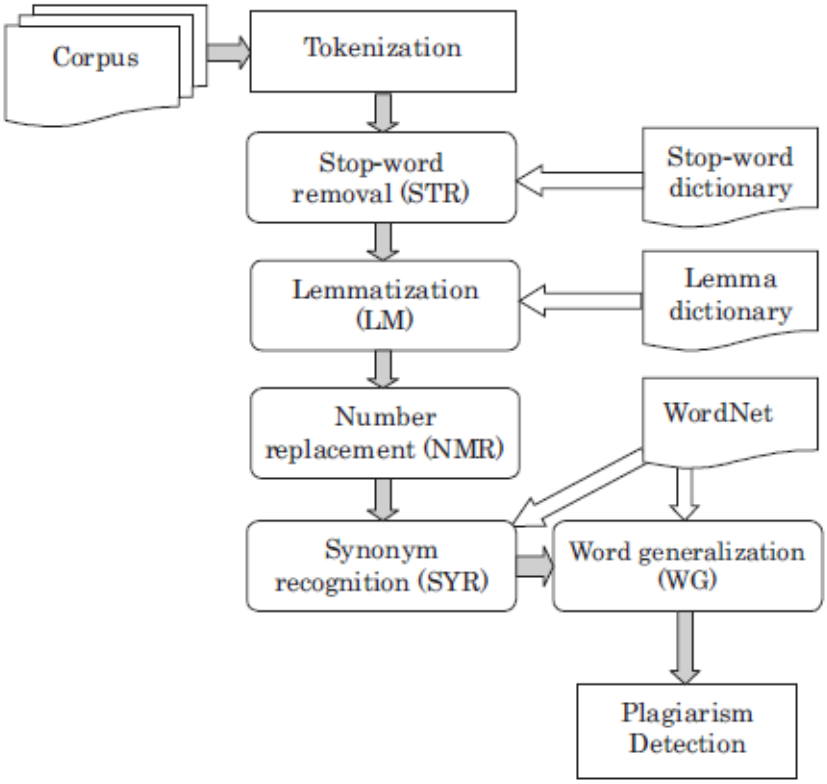


FIGURE 2.9: Text Preprocessing Techniques for Plagiarism Detection (Ceska and Fox, 2009)

Hoad and Zobel (2003) use tokenization, stopword removal and stemming as text preprocessing before candidate document selection, while Nawab, Stevenson, and Clough (2011) use tokenization, case folding and number replacement as preprocessing operations. Although text prerocessing may be considered to be a fundamental part of any plagiarism detection approach, some approaches – notably those involving character *n*-grams – do not involve any text preprocessing. For example Stamatatos (2009) does not use any text preprocessing at all for his approach towards intrinsic plagiarism detection. Obviously this is not an NLP-based approach, since as stated by the author, it requires "*no language dependent resources... and no segmentation or preprocessing*".

NLP techniques such as synonymy recognition require computational resources such as comparison with a database (e.g., WordNet). However, the overall improvements of using NLP techniques at the preprocessing stage can be considered as marginal in comparison to their computational overhead. This point is emphasised by Ceska and Fox (2009) who have stated that, "*On the basis of our experiments, text preprocessing cannot significantly improve the accuracy of plagiarism detection. Only number replacement (NMR), synonymy recognition (SYR) and word generalization (WG) improve the accuracy slightly.*"

2.5.3 Stopword *n*-grams and Stopword Removal

Stopwords can be considered as words that do not have a meaning on their own but support the content words of a text in giving structure to text. Hoad and Zobel (2003) state that stopwords are "*... closed-class words words–for example "the", "of" and "may"–that indicate the structure of a sentence and the relationships between the concepts presented, but do not have any meaning on their own*". Rajaraman and Ullman (2011) define stopwords as "*(stopwords) are the most frequent in a document and they are removed... infact the indicators of topics are relatively rare words*". For indexing, stopwords are usually removed (Witten, Moffat, and Bell, 1999). From a plagiarism detection perspective, two techniques are in use: (a) stopword removal and (b) stopword *n*-grams.

Stopword Removal

Stopword removal is a common technique at the candidate document selection stage, as well as at the detailed analysis stage. One rationale behind this technique is that it reduces the size of data for searching and matching and is based on more relevant terms. A lot of research works involve stopword removal prior to similarity computation for plagiarism detection. For example, Hoad and Zobel (2003), Gustafson, Pera, and Ng (2008) and Chong, Specia, and Mitkov (2010) remove all stopwords. Similarly, Vania and Adriani (2010) apply stopword removal for their PAN 2010 software submission entry.

However there are plagiarism detection approaches as well where stopwords are not removed. For example, Oberreuter et al. (2011) do not remove any stopwords at all for their PAN 2011 entry – according to them, one advantage is that this approach could be used for multiple languages since it does not use language dependent features like stopword removal.

Stopword *n*-grams

Apart from stopword removal, there's a very interesting line of research – given a piece of text, just retain stopwords and remove everything else. The resulting text containing stopwords only can be converted to *n*-grams. This approach is called stopword *n*-grams and has been proposed by Stamatatos (2011). This approach seems counterintutive to stopword removal at first, since all other relevant words have been removed. However, stopword *n*-grams have are helpful in revealing the underlying syntactic structure of text. Figure 2.10 (a) gives a list of 50 most frequent stopwords in the British National Corpus. Furthermore, Figure 2.10 (b) shows the text after removing all (content) or non-stopwords from the text, while Figure 2.10 (c) gives the entire list of stopword 8-grams extracted from the text.

Stopword *n*-grams have shown good performance in plagiarism detection cases where most of the words have been replaced by synonyms and therefore have been used as features by many researchers. Bär, Zesch, and Gurevych (2012) use a composition of text similarity measures including stopword *n*-grams for various values of *n*. Bär et al. (2012) use both stopword removal and stopword *n*-grams for the STS Sentence Similarity Task as two of several features used to determine sentence similarity.

TABLE 2. The list of 50 most frequent words of BNC corpus used in this study.

1. the	11. with	21. are	31. or	41. her
2. of	12. he	22. not	32. an	42. n't
3. and	13. be	23. his	33. were	43. there
4. a	14. on	24. this	34. we	44. can
5. in	15. i	25. from	35. their	45. all
6. to	16. that	26. but	36. been	46. as
7. is	17. by	27. had	37. has	47. if
8. was	18. at	28. which	38. have	48. who
9. it	19. you	29. she	39. will	49. what
10. for	20. 's	30. they	40. would	50. said

These savage birds are very common in Maine, where they make great havoc among the flocks of wild-ducks and Canada grouse, and will even, when driven by hunger, venture an attack on the fowls of the farm-yard.

(a) A text passage

are in they the of and and will by an on the of the

(b) The text after removing all tokens not found in the stopword list

[are, in, they, the, of, and, and, will]
[in, they, the, of, and, and, will, by]
[they, the, of, and, and, will, by, an]
[the, of, and, and, will, by, an, on]
[of, and, and, will, by, an, on, the]
[and, and, will, by, an, on, the, of]
[and, will, by, an, on, the, of, the]

(c) The stopword 8-grams of the text

FIG 2. An example of transforming a text to stopword *n*-grams.

FIGURE 2.10: Stopword *n*-grams for Plagiarism Detection (Stamatatos, 2011)

2.5.4 Stemming and Lemmatization

Stemming and lemmatization are similar in terms of their goal of finding the base form of a given word. For example the words *derive*, *derived*, *derives* and *derivation* can all be linked back to the base form *derive*. However, stemming and lemmatization differ in their methods of extraction.

Stemming refers to "*a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time*", while ***Lemmatization*** "*usually refers to doing things properly with the use of a vocabulary and morphological analysis of words*" (Manning, Raghavan, and Schütze, 2008). Stemming does not take context into account, unlike lemmatization which does take context into consideration. For stemming, Porter's algorithm (Porter, 1980, 2001) for suffix removal is used along with its variants. On the other hand lemmatization reduces a word to its lemma, which is usually achieved by searching for the root word in a database, e.g., WordNet.

Both lemmatization and stemming have been used for plagiarism detection, e.g., Ceska and Fox (2009) use lemmatization. Similarly Hoad and Zobel (2003) use both stopword removal and stemming for plagiarism detection. Usually, language independent approaches do not use stemming or lemmatization at all. The effect of lemmatization has been reported as statistically insignificant on the detection accuracy in addition to increased execution time. A similar view is also stated by Manning, Raghavan, and Schütze (2008) that "*Doing full morphological analysis produces at most very modest benefits for retrieval...While it helps a lot for some queries, it equally hurts performance a lot for others.*"

Tools for Stemming and Lemmatization

Several online and downloadable tools are available that perform both stemming and lemmatization. Implementations of Porter's Algorithm are available online as Porter Online Stemmer[3] and as part of the NLTK Python Natural Language Processing toolkit. Similarly, the NLTK Online Lemmatizer[4] is also available which is based on extract of lemma(s) from Wordnet. Here we give an example of both stemming and lemmatization from these two sources.

Stemming and Lemmatization Example

Let's consider the following sentence and its stemmed and lemmatized versions.

- Sentence: "Our outcomes are very encouraging" [MSR Paraphrase Corpus]
- Stemmed: "Our outcom ar veri encourag" [Porter Online Stemmer]
- Lemmatized: "Our outcome be very encourage" [NLTK Online Stemmer]

If we observe at the base form of "are", in the stemmed output it is "ar", while in the lemmatized output it is "be". This clearly points to a deeper level of analysis in lemmatization, but it also suggests additional performance overhead.

[3] http://9ol.es/porter_js_demo.html
[4] http://textanalysisonline.com/nltk-wordnet-lemmatizer

2.5.5 Other Preprocessing Operations

Some other more notable techniques are mentioned here, but their impact is not very significant.

Case Folding

Case Folding refers to replacing all uppercase letters in a string with lowercase (Witten, Moffat, and Bell, 1999). For example, the words `The`, `the`, and `THE` should all be represented by `the`. However, there is a possibility that two different words may be treated as the same in case-folding (Barrón-Cedeño, 2012), e.g. the name `Brown` and the color `brown` may be interpreted to be the same. In addition to the previously mentioned approaches, case folding is also used as text preprocessing for plagiarism detection.

Punctuation Removal

Punctuation Removal refers to the complete removal of punctuation from a sentence or text. Punctuation Removal as part of text preprocessing is important since alphabetic textual information is revealed. However, care has to be taken as to which punctuation marks have to be removed. This is particularly true for the dot ".", where a dot identifies sentence boundaries, abbreviations and sometimes decimal point in numerical information.

2.5.6 POS Tagging

Parts-of-speech tagging (or POS Tagging) (Derose, 1990) is the process of marking a word in a text as being a part-of-speech (e.g. noun, verb, adjective, etc). The number of tags available is dependent on the tagset employed. Commonly available tagsets include the Penn Treebank tagset (36 POS tags + 12 other tags = 48 tags) (Marcus, Santorini, and Marcinkiewicz, 1993) and the British National Corpus tagsets C5 and C7 (Rayson and Garside, 1998). Several issues and design decisions need to be considered for POS-tagging, e.g., whether to employ rule-based, stochastic or hybrid taggers. Ambiguity in tag determination is another issue which has to be considered in the context of text reuse detection.

As an example, let us consider the POS tagged version of the following two sentences as determined by the Stanford POS tagger:

- This is a cat. ⇒ This/`DT` is/`VBZ` a/`DT` cat/`NN`.
- That was a rat. ⇒ That/`DT`, was/`VBD` a/`DT`, rat/`NN`.

Here the tags have been taken from the Penn Treebank tagset[5]. Briefly, these correspond to, `DT` = Determiner, `VBZ`, `VBD` = verb and `NN` = noun). It is evident that the underlying structure of the two sentences is the same.

Application

Chong, Specia, and Mitkov (2010) have used POS tagging towards plagiarism detection, especially "*for cases where words have been replaced but the style in terms of grammatical category remains the same*". They used POS tags along with other features and used a trigram similarity metric to compute similarity. Effectively, this suggests POS tagging is being used to determine word replacement while maintaining the text structure. Another usage of POS tags has been undertaken by Elhadi and Al-Tobi (2008) where they use POS tags along with LCS (Longest Common Subsequence) algorithm to compute similarity. Here also the idea underlying the usage of POS tags is similarity of syntactical structure.

Apart from being directly used to determine sentence structure, POS tags are also used for determining synonyms using WordNet. Chen, Yeh, and Ke (2010) state that "*Besides the word itself, the POS of the word is necessary as well to retrieve its information in WordNet.*" This type of usage indirectly contributes towards plagiarism detection.

2.5.7 Syntactic Methods

In the previous sections we outlined methods that use syntactical features for plagiarism detection. In particular, the use of stopword *n*-grams and the use of POS tags exploit the similarity in syntactical structures to determine document similarity. In this section we look at parsing and related syntactic features to evaluate document similarity.

2.5.8 Syntactic Parsing for Plagiarism Detection

Parsing, more specifically, "*grammar-driven natural language parsing... is analyzing a string of words (typically a sentence) to determine its structural description according to a formal grammar*" (Ljunglöf and Wirén, 2010). The output of parsing is a syntactic structure (usually a parse tree) that is suitable for further analysis. Parsing is widespread in programming language code and is used to determine the structure of a program, based on the specified grammar of the underlying programming language. However, parsing of programming language code is much simpler than parsing natural languages, the reason being that "*the complete grammar for a programming language can be defined and specified, but natural language is much more complex and ambiguous making it much harder to build a successful plagiarism system.*" (Clough, 2000).

[5] http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

This is one of the reasons why parsing has not been used as widely in natural language plagiarism detection as compared to plagiarism detection in software. Figure 2.11 shows a syntax (dependency) tree and the associated dependency relations between words, discussed next.

Dependency Relations

A dependency relation (Tesnière, 1959) is a binary asymmetric relation between words in a sentence. In the words of (Nivre, 2006) "*Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies.*" Dependency Relations are used to show the relationships between words in a sentence; in other words they can be considered as flat representations of a parse tree. In some of the works on plagiarism detection stated here, researchers have also used dependency relations for plagiarism detection. Figure 2.11 shows a dependency tree and the dependency relations for the sentence: "*Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas*".

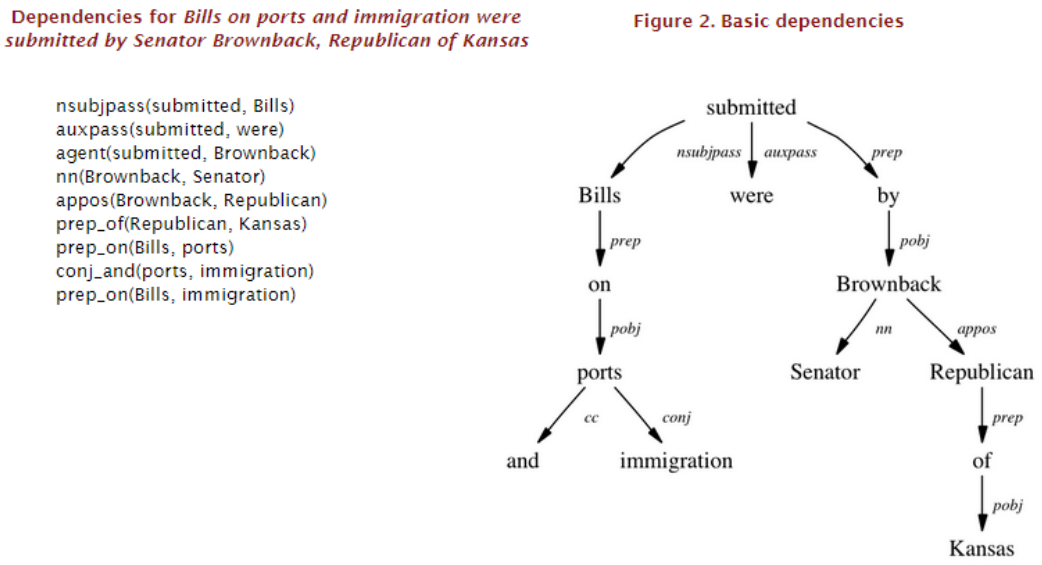


FIGURE 2.11: (Stanford) Dependency Relations and Tree for a Sentence

Application for Plagiarism Detection

In this subsection, we present an overview of some of the works applying parsing and dependency trees towards plagiarism detection.

Mozgovoy, Kakkonen, and Sutinen (2007) use syntactic parsing to address the split match problem in plagiarism detection. The split match problem refers to the situation when string comparison algorithms find identical fragments of strings in

the source and suspicious documents but are unable to combine these into larger segments due to intentional word swapping (Mozgovoy, Kakkonen, and Sutinen, 2007). The work uses parsing, more specifically the Stanford parser, to find dependency trees in the text. These dependency trees are converted into grammatical relations in order to combine isolated matching fragments into larger sections. The system was tested on 128 messages obtained from the BBC Website and an unspecified number of intentionally plagiarised texts with swapped words. The results show a slight increase in the similarity ratios of by up to 7%-13%.

Leung and Chan (2007) use syntactic parsing as part of a group of NLP techniques for plagiarism detection. They use parsing at the sentence level and construct parse trees for each sentence. This is done in addition to word replacement and semantic processing. The primary motivation for parse tree construction is to compare sentences for structural similarity. However, their work does not refer to any experiments carried out or the corpus on which this approach was tried. Primarily, this is a theoretical work that proposes a method for plagiarism detection.

Chong, Specia, and Mitkov (2010) also use the Stanford Parser (Manning, Raghavan, and Schütze, 2008) on the Corpus of Plagiarised Short Answers (Clough and Stevenson, 2011) to produce dependency relations from the syntax trees of sentences in both the original and the plagiarised documents. These dependency relations are textual representations of the various words in the sentence exhibited in the parse tree. Chong, Specia, and Mitkov (2010) used the number of common relations divided by the number of dependency relations in the suspicious document as a similarity measure. This metric achieved the best correlation score between a document and its plagiarised class.

Tschuggnall and Specht (2013) use grammar trees of texts for intrinsic plagiarism detection. Under the assumption that text by the same author uses similar grammar (parse) trees for sentences, they detect suspicious portions of text by finding and grouping sentences that have substantially different grammar trees than the remaining text. They tested their approach on the PAN 2011 corpus with randomly plagiarised portions of text.

2.5.9 Semantic Methods

In this section we consider semantic techniques used to conduct plagiarism detection. By semantic techniques we mean methods that involve the meaning of linguistic units. Goddard and Schalley (2010) define semantic analysis as "*semantic analysis refers to analyzing the meanings of words, fixed expressions, whole sentences and utterances in context.*" This will be the guiding principle in this section, i.e., any technique that is concerned with the meaning of linguistic units will be considered as a semantic method.

2.5.10 Synonym Recognition Using WordNet

Synonym recognition is a key technique used for plagiarism detection. Ceska and Fox (2009) use synonymy recognition for plagiarism detection with the motivation that "*synonymy recognition comes from human behaviour whereby people may see to hide plagiarism by replacing words with appropriate synonyms*". A plagiarist may also replace multiple words or phrases with synonymous phrases in order to obfuscate text reuse. In the context of plagiarism detection, one of the most commonly used data sources for finding synonyms is WordNet (Miller, 1995).

For plagiarism detection, Ceska and Fox (2009) use three different techniques for synonymy recognition. This is partly because words may have multiple meanings, and therefore **Word Sense Disambiguation (WSD)** is also used to find the closest meaning. They use WordNet thesaurus to find the meanings, since words in WordNet are linked into synsets with each synset having an Inter Lingual Index (ILI) to identify the synset as follows.

1. The **First Meaning Selection** finds and returns the first ILI corresponding to the first match in WordNet.

2. The **Disambiguation and Proper Meaning Selection or DPMS** aims to select the best meaning based on the adjacent (context) words, however success is not guaranteed.

3. The **Every Meaning Selection or EMS** selects and returns all meanings of a given word from WordNet. The advantage of this technique is that it finds all possible meanings, however it is too permissive.

They perform their experiments of synonymy recognition on the METER Corpus, as no training data is available on the Czech language corpus. Their results suggest that EMS performs better than FMS while DPMS performs worse than EMS. The reason for this performance is that, if an appropriate meaning is not selected through the disambiguation process, then a random meaning is selected. In general, they conclude that EMS is the best choice not only for their tested corpora, but for plagiarism detection in general.

Nawab (2012) adopts a slightly different approach by generating modified n-grams. They generate modified n-grams by replacing words with their synonyms using three different resources: (a) WordNet (Miller, 1995), (b) Paraphrase Lexicon (Callison-Burch, 2008) and (c) UMLS (Unified Medical Language System) Metathesaurus (Bodenreider, 2004). In this approach, they also generate expanded versions of n-grams for queries (for the candidate document selection stage) in addition to

replacement of words. Similar to Ceska and Fox (2009)'s EMS they generate all possible modified n-grams by generating all possible synonyms from synsets. To relieve the large number of n-grams generated, weighting schemes are used whereby weights are assigned to n-grams based on their frequency. Figure 2.12 shows n-grams generated from WordNet. The modified text with the newly generated n-grams is then tested for similarity measures with the suspicious text. In the following example, a list of modified n-grams for the original sentence is presented (Nawab, 2012).

- Original Sentence: He rides a new car.

Newly generated text using Modified n-grams:

- He rides a new motorcar;
- He rides a new automobile;
- He rides a fresh motorcar;
- He rides a fresh automobile.

Chen, Yeh, and Ke (2010) have also used WordNet for plagiarism detection by searching for synonymous words in a given text. They employed the Jaccard measure to calculate the similarity between two synsets where there are overlapping synonyms. Figure 2.12 gives a view of their approach.

Synsets of *shouts (verb)*:

1. shout
2. shout, shout out, cry, call, yell, scream, holler, hollo, squall
3. exclaim, cry, cry out, outcry, call out, shout
4. abuse, clapperclaw, blackguard, shout

Synsets of *yells (verb)*:

1. shout, shout out, cry, call, yell, scream, holler, hollo, squall
2. yell, scream

As shown above, the verb, *shouts*, has four synsets and the verb, *yells*, has two synsets in WordNet. By looking at synsets 1 of both words, there is only one overlapping synonym while the distinct union of synonyms is 9. Therefore, the Jaccard's coefficient for these two synsets will be 1/9.

Fig. 5 Example of Jaccard's Coefficient between Two Synsets

FIGURE 2.12: Synset Comparison Using Jaccard Coefficient in WordNet

2.5.11 Word Generalization

In addition to the semantic relationship of synonym, i.e., words having similar meaning, there are other semantic relations between words which are used in WordNet to denote relationships between synsets. Some of the more important ones are stated here as follows.

- **Hypernym**: A word that represents a more general category of a given word (for example, 'bird' is a hypernym of 'crow', i.e., 'crow' is a kind of 'bird'). This relation is applicable to both nouns as well as for verbs.
- **Hyponym**: This is the converse relationship, i.e, a hyponym represents a specific type of hypernym (e.g. 'crow' is a hypernym of 'bird').
- **Troponym:** This is the converse relationship of hypernymy for verbs. For example ('to read' is a troponym of 'to study').

These relationships can also be used for plagiarism detection, especially when words replaced by a plagiarist are not synoynms but they belong to the same category. For example the words "dog" and "cat" can be replaced by the word "animal" – an example of word generalization (Ceska and Fox, 2009). WordNet (Miller, 1995) provides synsets with each synset being labeled by an Interlingual reference or ILI (Inter Lingual Index in WordNet) thereby stating the general category of the word. One advantage of word generalization is that it could be used to detect plagiarism when a word has been replaced by its **co-hyponym** (where a co-hyponym is another noun of the same hypernym). For example a replacement of the word 'dog' by a 'cat' is a co-hyponym replacement, as both these words have a common hypernym 'animal'.

A design issue that needs to be considered is: what level of generalization to incur? Ceska and Fox (2009) state that too much generalization can lead to loss of information with the result being that plagiarism might not be detected at all. They suggest a middle approach whereby each word is replaced by its corresponding ILI for deeper (more specific) words, while shallower (more general) words are not replaced at all. In practice, their results suggest that fourth (4^{th}) level word generalization gives good value of the F_1 measure when combined with synonym replacement.

Chong and Specia (2011) use lexical generalization for plagiarism detection on the PAN 2010 corpus. For lexical generalization, "*functional words (stop words) were removed and all remaining (content) words were generalised using their WordNet synsets, that is, groups of synonym words.*" In this approach they did not use any word-sense

disambiguation since words were replaced by all possible counterparts in their corresponding synsets. Strictly speaking, this is slightly different to word generalization as words are being replaced not by their general category, but rather by all of the words within that category.

For similarity detection, they use a binary classification system of plagiarised or non-plagiarised pairs in contrast to PAN's segment level passage alignment. Their results suggest an improved detection accuracy as according to them, "*this strategy has identified significantly more simulated and obfuscated plagiarism cases than the baseline.*"

2.5.12 Fuzzy Semantic Methods

In Fuzzy Semantic Methods, fuzzy set theory (Zadeh, 1965) is used for plagiarism detection. A fuzzy set is based on the idea of partial membership function for a set. Crisp sets assign a value of '0' if an element is not a member of a set and a value of '1' for a membership. In contrast, a fuzzy set can have values between '0' and '1' representing partial membership within the set.

By applying the same idea to plagiarism detection, a spectrum of similarity values between '0' and '1' is assigned to two fragments of text. For example, similarity values can be assigned to sentences that "*range from one (exactly matched) to zero (entirely different)*" (Alzahrani, Salim, and Abraham, 2012). The key design issue here is the construction of the fuzzy set, i.e., how to assign values of partial similarity.

Alzahrani and Salim (2010) use fuzzy set theory for plagiarism detection in their submission for PAN 2010. They assign a fuzzy value of 0.50 for words that occur within synsets of each other within WordNet, and then calculate the overall fuzzy similarity between corresponding sentences using word-to-word similarity values. Figure 2.13 gives a pictorial representation of their approach.

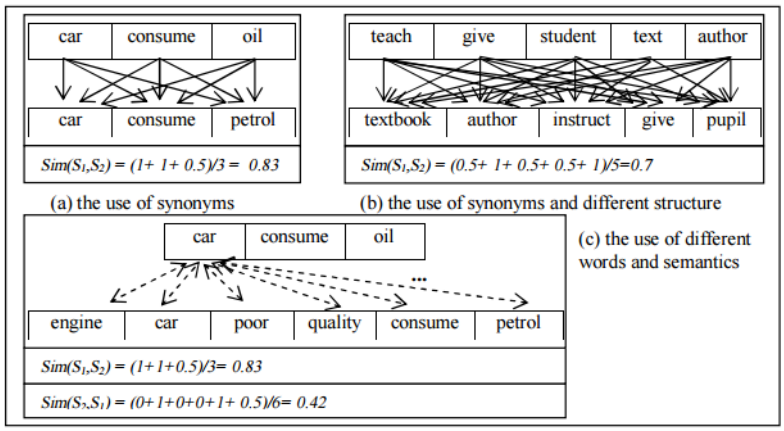


Figure 3. Examples of different sentence pairs.

FIGURE 2.13: Example of Similarity Detection using Fuzzy Semantic Methods (Alzahrani and Salim, 2010)

Yerra and Ng (2005) take a sentence based similarity detection approach where they use fuzzy set information retrieval for copy detection on web documents. They use a term-to-term correlation matrix where they employ fuzzy values to denote similarity between individual terms. This idea of similarity between individual terms is then extended to entire sentences. Their results suggest an improvement in the detection accuracy over traditional *n*-gram model for copy detection.

2.5.13 Section Summary

In this section we reviewed a number of plagiarism detection techniques that involve language processing. We gave an overview of preprocessing methods including stopword removal, stopword n-grams, syntactic parsing and semantic methods including synonym recognition, word generalization and fuzzy semantic methods.

Text preprocessing, which is typically the first step in application of NLP techniques includes tokenization and segmentation. NLP Techniques such as stemming, lemmatization and POS tagging may also considered to be part of text preprocessing, however these are relatively resource intensive. These NLP techniques may be integrated to form a plagiarism detection approach, such as the approach illustrated in Figure 2.9.

NLP techniques using stopwords, such as stopword removal and the use of stopword *n*-grams are useful in revealing the underlying structure of text, thereby proving to be effective for detecting plagiarism. Syntactic parsing which includes the generation of dependency relations as well as grammar trees has also been used for plagiarism detection. Finally, the use of semantic methods such as the application of WordNet relations as well fuzzy semantic methods have also been used successfully for plagiarism detection.

Natural Language Processing is helpful in giving us a better understanding of changes within text from a language processing perspective. The techniques presented in this section may form whole or part of a plagiarism detection approach. However since language processing based methods are more resource intensive, these methods can be used in combination with lexical (non-NLP) methods for plagiarism detection.

2.6 Paraphrase Typologies

In the last few years, research on plagiarism has focused on classifying paraphrase types within plagiarised text. Since paraphrase plagiarism has already been a well recognized form of plagiarism, paraphrase types take this research trend further by proposing to identify textual rewrite operations within paraphrased, plagiarised text into various paraphrase types.

In this section, we provide a background on paraphrase types (which can also be considered as textual rewrite operations) and compare three research works that list and annotate paraphrase types in text.

2.6.1 Overview of Textual Transformations

Researchers have listed a variety of textual transformations in various works. These sets of textual transformations come from a wide variety of contexts, more notable among these being paraphrasing, summarization and academic writing. The terminology of these transformations also varies across the literature. In this section we give a brief overview of the various transformation lists given in the literature. In the context of paraphrasing, such lists have been called as paraphrase typologies in the literature.

2.6.2 Earlier Works

There are works in the area of discourse analysis where classifications of paraphrasing have been given. For example, Gülich (2003) classifies paraphrases into four types: metaphors, exemplification, scenarios and concretization. However their emphasis is on communication, and not on the linguistic operations that underlie the paraphrases themselves.

From the perspective of paraphrasing as well as summarization, a number of typologies have been proposed. Here we give a short description.

1. Barzilay, McKeown, and Elhadad (1999) give a list of seven paraphrasing rules: (1) sentence components' ordering, (2) division of clauses, (3) realization, (4) change in grammatical features, (5) head omission, (6) part of speech transformation, and (7) synonym usage.

2. Jing and McKeown (1999) give a list of sentence transformations in the context of summarization. These include sentence reduction, sentence combination, syntactic transformation, lexical paraphrasing, generalization/specialization and reordering. In contrast to the earlier list, most of these transformations involve operations on sentences.

3. In the context of the METER corpus, Clough (2003a) has identified several textual transformations including substitution of synonyms, temporal changes, change of tense, to name a few. Some of these transformations are relevant to the news domain, for example, exaggeration.

4. Several other works give lists of various typologies, which are either based on the idea of paraphrase equivalence (Dorr et al., 2004), or as detailed lists

of linguistic operations (Fujita, 2005). An overview based on paraphrasing perspective appears in (Vila, Martí, Rodríguez, et al., 2014).

In addition to lists of paraphrase types, another trend observed in the research works such as (Barrón-Cedeño et al., 2013; Bhagat and Hovy, 2013; Sun and Yang, 2015) is the percentage occurrence of each type of paraphrase in a given data collection. Such a quantitative breakdown gives us a useful estimate into the frequency of a particular type of textual transformation within a dataset.

2.6.3 What is a paraphrase?

Bhagat and Hovy (2013) have used the notion of *quasi-paraphrases* to identify 25 different types of paraphrases. Quasi-paraphrases are based on the idea of approximate equivalence and are defined as "*sentences or phrases that convey approximately the same meaning using different words*".

Using this idea of approximate equivalence, they have identified 25 different types of quasi-paraphrases from a lexical perspective – i.e. "*the kinds of lexical changes that can take place in a sentence/phrase resulting in the generation of its paraphrases*". In other words, their paraphrase types correspond directly to textual transformations that can be undertaken to transform one sentence/phrase into another.

Some of the more important paraphrase types (based on the frequency of occurrence) identified in their list include:

- synonym substitution;
- function word variations;
- repetition/ellipsis;
- general/specific substitution;
- verb/(noun, adjective, adverb) conversion;
- change of tense;
- approximate numerical equivalence.

Using the list of paraphrase types, they have annotated sentence pairs into one or more of their types from a subset of MTC (Multiple Translations Corpus) (Huang et al., 2002) and the MSRP (Microsoft Research Paraphrase) corpus (Dolan, Quirk, and Brockett, 2004). They have reported synonym substitution and function word variation as two of the most frequent paraphrase types both in MTC and the MSR corpora, where they account for a combined percentage of approximately 75% of the paraphrases in MTC and 50% of the paraphrases in the MSR corpora respectively.

Furthermore, it can also be observed that a majority of the transformations have minimal occurrence (0% – 1%) in the two corpora. These findings suggest that synonym substitution and function word variation constitute a significant proportion of

paraphrases in the subsets selected of the two corpora. Figure 2.14 gives a description of synonym substitution as stated in their work.

1. **Synonym substitution:** Replacing a word/phrase by a synonymous word/phrase, in the appropriate context, results in a paraphrase of the original sentence/phrase. This category covers the special case of genitives, where the clitic *'s* is replaced by other genitive indicators like *of*, *of the*, and so forth. This category also covers near-synonymy, that is, it allows for changes in evaluation, connotation, and so on, of words or phrases between paraphrases. *Example:*

(a) Google *bought* YouTube. ⇔ Google *acquired* YouTube.

(b) Chris is *slim*. ⇔ Chris is *slender*. ⇔ Chris is *skinny*.

FIGURE 2.14: A Description of Synonym Substitution (Bhagat and Hovy, 2013)

2.6.4 Plagiarism Meets Paraphrasing

In another work, Barrón-Cedeño et al. (2013) highlight the connection of paraphrasing in the context of plagiarism and plagiarism detection. They forward their observation that state-of-the-art plagiarism detectors are limited in their ability to detect paraphrase plagiarism. Furthermore, they present a typology of paraphrase types that is relevant to plagiarism and its detection. This typology is an extended version of an earlier typology presented in (Vila, Martí, and Rodríguez, 2010); likewise, a similar typology also appears in (Vila, Martí, Rodríguez, et al., 2014). Some of the more important types in their work are:

- Same Polarity Substitution;
- Opposite Polarity Substitution;
- Spelling and Format Changes;
- Inflectional Changes;
- Modal Verb Changes;
- Punctuation and Format Changes;
- Direct/Indirect Style Alternations;
- Change of Order;
- Addition/Deletion.

Among these types, same polarity substitution is the most frequent type reported in the P4P (Paraphrase for Plagiarism) corpus which is derived from the PAN-2010 corpus. Same Polarity Substitution can be considered to be a broader definition of synonym substitution where other general substitutions or quantitative substitutions might also be carried out. Figure 2.15 shows the definition of same polarity substitution along with an example. As compared to synonym substitution in Figure 2.14, this definition includes a broader range of substitutions.

Same-polarity substitutions change one lexical (or functional) unit for another with approximately the same meaning.[6] Among the linguistic mechanisms of this type, we find synonymy, general/specific substitutions, or exact/approximate alternations. In Example (9), *very little* is more general than *a teaspoonful of*.

(9) a. *a teaspoonful of* vanilla
b. *very little* vanilla

FIGURE 2.15: A Description of Same Polarity Substitution (Barrón-Cedeño et al., 2013)

2.6.5 A Mapping Between the Two Paraphrase-Type Lists

The paraphrase types listed in (Barrón-Cedeño et al., 2013) are based on the "*general linguistic phenomena*" of paraphrasing, in contrast to a more fine-grained list presented in (Bhagat and Hovy, 2013). However, even with this difference in point-of-view, the two lists broadly represent similar paraphrase types, with some types being exactly the same both in name and description. Hence, a general mapping can be carried out between the two lists. For example, we observe that the type "*same-polarity substitution*" in Barrón-Cedeño's work corresponds to "*synonym substitution*" in Bhagat's list. Likewise, *converse substitutions* in Bhagat's list map to *converse substitutions* in Barrón-Cedeño's list; similarly *repetition/ellipsis* maps to *ellipsis*.

In Table 2.1, we provide a mapping between the paraphrase types of the two lists. The purpose of this mapping is to show a general sense of similarity within the paraphrase types, even with the diverse viewpoints taken in the formulation of the lists. In this sense, this partial mapping of paraphrase types shows a general agreement between the paraphrase types in the two lists.

No	Bhagat and Hovy (2013)'s List	Barrón-Cedeño et al. (2013)'s List
1	Synonym Substitution	Same Polarity Substitution
2	Converse Substitution	Converse Substitution
3	Repetition/Ellipsis	Ellipsis
4	Antonym Substitution	Opposite Polarity Substitution
5	Change of Person	Direct/Indirect Style Alternations
6	General/Specific Substitution	Same Polarity Substitution
7	Change of modality	Modal Verb Changes
8	Approximate numerical equivalences	Same Polarity Substitution
9	-	Punctuation and format changes
10	External Knowledge	-

TABLE 2.1: A Mapping Between Paraphrase Types from Two Works

2.6.6 The P4P (Plagiarism for Paraphrase) Corpus

Barrón-Cedeño et al. (2013) refer to paraphrasing from a plagiarism detection point of view. Building upon this viewpoint, the authors constructed the P4P corpus, also called the Plagiarism for Paraphrasing Corpus. This corpus consists of cases of paraphrasing from the PAN-2010 plagiarism detection corpus. The P4P corpus contains 847 cases of simulated plagiarism as snippet pairs from source and suspicious documents. Each snippet pair has been annotated manually, based on the one or more of the paraphrase types given in their paraphrase typology.

Figure 2.16 gives a distribution of the different types of paraphrases in the P4P corpus. In order to have a realistic comparison, the distribution of paraphrase types in the sub-METER and RWP corpora are also given in the Figure. The sub-METER corpus is a subset of the METER corpus with real cases of text reuse, while the RWP corpus is a collection of Real-Web Plagiarism cases reported online (Barrón-Cedeño et al., 2013). These two corpora were also annotated with various paraphrase types in order to observe the distribution of the various paraphrase mechanisms in real cases of plagiarism.

From Figure 2.16 it can be seen that the same-polarity substitutions dominate the list with the highest frequency. This is in agreement with the other research works, where synonym substitutions had a very high frequency among other paraphrase types. Furthermore, addition/deletion has the second highest frequency. Another important result from this list of annotated paraphrase types is that most paraphrase types have low relative frequency in the corpora.

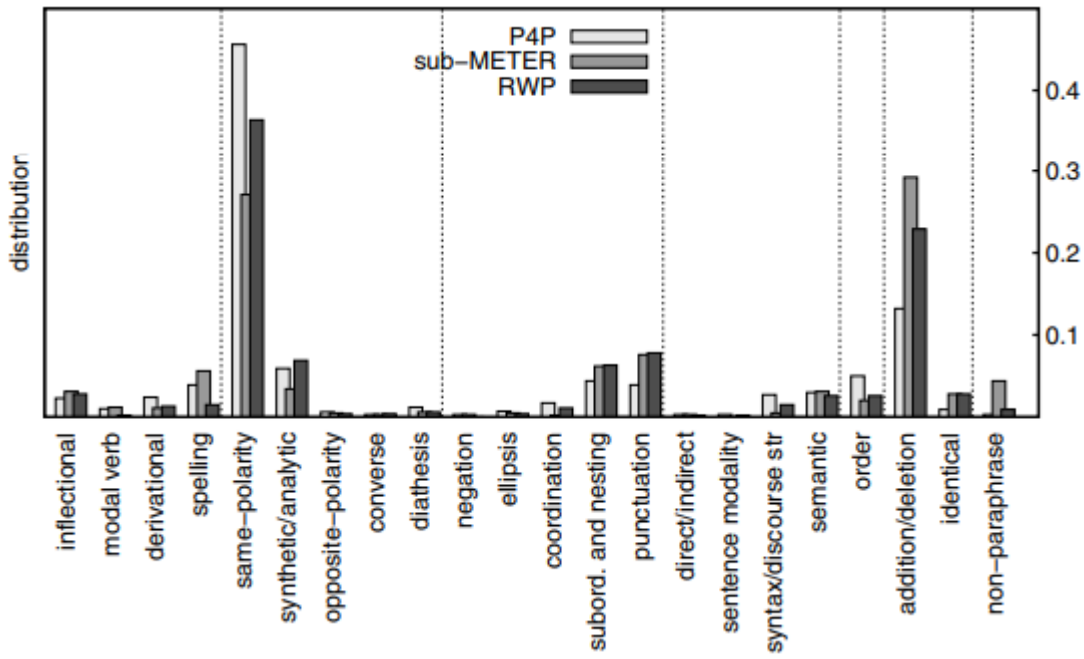


FIGURE 2.16: A List of Paraphrase Types with their Frequency of Occurrence (Barrón-Cedeño et al., 2013)

2.7 Methodologies for Paraphrase Type Identification

Identification of paraphrase types in the context of plagiarism detection is a problem that has not been formally addressed in the literature to the best of our knowledge. However, literature on identifying synonyms and the use of contextual information is helpful towards paraphrase type identification in general, and same polarity substitution in particular. In this section we give an outline on some of the methods that direct towards methods suitable for detecting paraphrase types in plagiarised text.

2.7.1 Distributional Hypothesis

The *distributional hypothesis* provides the foundation to the problem of detecting synonymy in general, and same polarity substitutions in particular. Some of the basic definitions of the distributional hypothesis state that:

- *words which are similar in meaning occur in similar contexts* (Rubenstein and Goodenough, 1965);
- *words with similar meanings will occur with similar neighbors if enough text material is available* (Schütze and Pedersen, 1995);
- *words are similar if their contexts are similar* (Freitag et al., 2005).

In general, the distributional hypothesis links the meanings of the words to the contexts in which they occur, which helps in inferring the meaning of a word as based on the environment in which it occurs and vice versa. Several models of semantics including word vectors (word embeddings) (Mikolov et al., 2013) and latent semantic analysis (Landauer, 2006) have been proposed using the distributional hypothesis.

2.7.2 Automatic Synonym Detection

Automated discovery of synonymous relations is a well studied problem in the literature with various proposed solutions. Here we give an overview of pattern based approaches and context based approaches.

Pattern Based Approaches

Pattern based approaches were first proposed by Hearst (1992) and extended in (Hearst, 1998). These approaches rely on acquisition of lexico-syntactic patterns in the target corpora, such as the pair "*X such as Y*" and "*X and other Y*". Patterns can be manually specified or automatically discovered.

As an example of automatic pattern discovery in (Hearst, 1992), the text is parsed for known pairs of words which are in a hypernym/hyponym relationship in close proximity to each other (e.g., *bicycle* and *vehicle*). Then those contexts where such pairs occur are considered as patterns and are re-examined for newer pairs of semantically related words occurring within them.

Systems which are based on pattern based approaches have a very high precision but low recall, since use of patterns guarantees a correct semantic relationship, while the incidence of many patterns is rare. Several improvements have been suggested such as in (Cederberg and Widdows, 2003) and (Ohshima and Tanaka, 2009) to improve high recall figure as well.

Context Based Approaches

Several approaches for automatic discovery of synonyms based on the context have been proposed in the literature, for example (Grigonyte et al., 2010; Moraliyski, 2013). The use of *paraphrase casts* was proposed by Grigonyte et al. (2010) and is based on local specific environment of words. In this approach, a pair of sentences is aligned using Sequence Alignment Algorithms and then words or phrases surrounded by the exact same context are considered to be synonymous and therefore substitutable. These have been names as 'paraphrase casts'. Figure 2.17 shows an example:

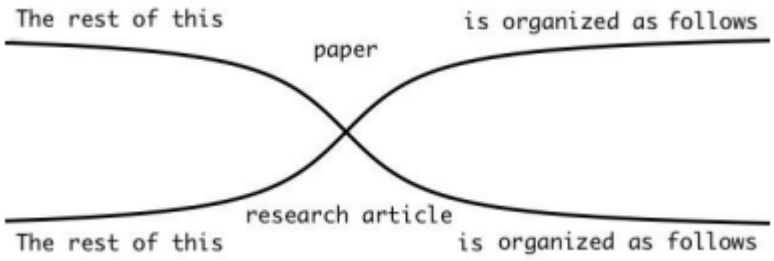


FIGURE 2.17: An Example of a Paraphrase Cast with Matching Left and Right contexts (Grigonyte et al., 2010)

This approach gave a precision of 67% on a domain specific corpus. However, the limitation of paraphrase casts is the discovery of only one synonymous pair per sentence pair and the exact matching of contexts.

A number of other approaches based on context have been proposed in various works, for example (Charles, 2000) and (Moraliyski, 2013) which use context in various forms to derive synonymous relations for various parts of speech.

2.7.3 Vector Representations of Words (Word Embeddings)

Word Embeddings (Ruder, Vulić, and Søgaard, 2019) are vector representations of words that have gained widespread acceptance in the last few years for a variety of Natural Language Processing tasks. Word Embeddings can be considered as a group of mathematical models, where each word is mapped to a vector space having a large number of dimensions (typically 300 or more). Word Embeddings are generated using a (typically large) text corpus (Allen and Hospedales, 2019) using a variety of methods such as neural networks, co-occurrence matrices and semantic networks. Word2Vec (Mikolov et al., 2013) is a popular model for generation of word embeddings from typically a large text corpus. Word2Vec has 2 architectures for the generation of word embeddings: (a) the CBOW (continuous bag of words) model, and (b) the skip n-gram model. GloVe (Pennington, Socher, and Manning, 2014) is another alternative model for generating word embeddings.

To compute similarity between word vectors, cosine similarity is used to find a similarity score between 0.0 and 1.0 between two words. Furthermore, vector operations such as vector addition and subtraction have been used in tasks such as detecting word analogies. Word Embeddings have proved to be successful in a number of NLP tasks, however polysemy remains a major issue. Figure 2.18 shows the vector space corresponding to word embeddings for an analogy task

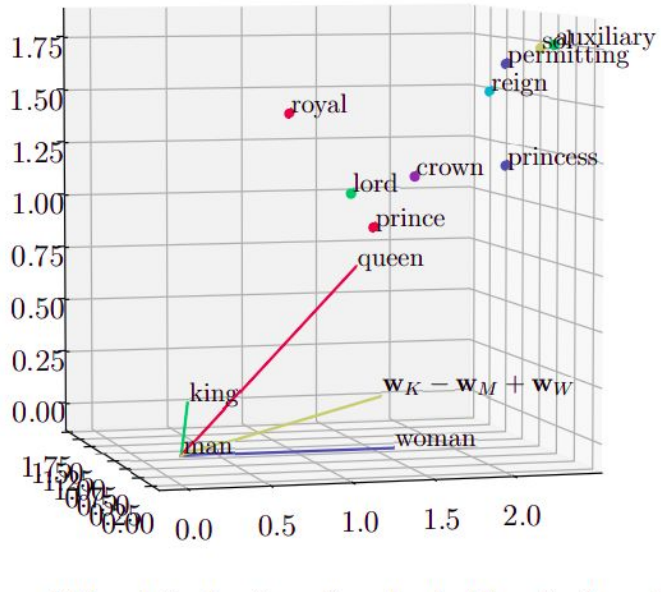


FIGURE 2.18: Word Embedding Vector Space corresponding to the Analogy "*man* is to *king* as *woman* is to *queen*"

(Allen and Hospedales, 2019)

Pretrained Word Embeddings

Pretrained word embeddings have been made available by several research groups based on a variety of models such as semantic networks, co-occurrence matrices and neural networks. Pretrained word embeddings have proved to be very useful in NLP tasks that *"suffer from a paucity of data"* (Qi et al., 2018). Pretrained word embeddings have shown their usefulness in a variety of tasks such as semantic textual similarity and neural machine translation. In this section we provide an overview of three off the shelf pretrained word embeddings, i.e., ConceptNet Numberbatch, FastText and GloVe.

1. **ConceptNet Numberbatch**: ConceptNet Numberbatch pretrained word embeddings have been generated from the ConceptNet Semantic Network, which is a multilingual semantic network originating from the MIT Media Lab. ConceptNet Numberbatch has proven its usefulness for word similarity task in Semeval 2017 (Speer, Chin, and Havasi, 2017) in the best performing approach. Figure 2.19 shows a part of the ConceptNet Numberbatch word vectors.

flagpole	0.0733	0.0615	0.011	0.0096	-0.167	0.0137	-0.1886
flagpoles	0.2564	-0.0249	0.0411	-0.0228	-0.1837	-0.043	-0.1435
flagrancy	0.3973	0.0156	0.0901	-0.1335	0.0726	-0.0185	0.1834
flagrant	0.0982	0.1119	-0.0291	-0.1667	0.0384	0.0202	0.1795
flagrante	0.6465	-0.0417	0.249	-0.0359	0.0142	0.0099	0.1038
flagrantly	0.1634	0.1004	-0.0546	-0.1862	0.0408	0.0585	0.1297
flagrate	0.5563	-0.0782	0.1586	0.0581	-0.0416	-0.1751	0.0547
flags	0.1162	0.0655	-0.0234	-0.0744	-0.0915	0.0678	-0.1311
flagship	0.0329	0.0332	0.0496	-0.0284	-0.0142	0.0571	-0.1852
flagship_store	0.1229	-0.2073	-0.0695	-0.1053	-0.0199	-0.0439	-0.1727

FIGURE 2.19: Part of ConceptNet Numberbatch Word Vector Matrix

2. **FastText**: FastText (Mikolov et al., 2018) also refers to a set of pretrained word embeddings with word embeddings provided in several languages trained on CommonCrawl and Wikipedia using the CBOW model. FastText word embeddings have proven useful for a variety of NLP tasks such as sentiment analysis.

3. **GloVe**: GloVe (Global Vectors for Word Representation) (Pennington, Socher, and Manning, 2014) are pretrained word embeddings made available by the NLP labs at Stanford University. These are trained on large corpora (CommonCrawl and Wikipedia) using co-occurrence matrices, and are available in a variety of dimensions.

2.8 Conclusion

This chapter provided a comprehensive background on plagiarism, plagiarism detection and paraphrase types. In particular, a detailed overview of plagiarism detection approaches was provided which included both lexical and language processing based approaches. Furthermore, paraphrase types and their relationship with paraphrase plagiarism was also presented along with methods to detect paraphrase types.

From the overview of the plagiarism detection approaches, we conclude that lexical (or non-NLP) plagiarism detection methods generally perform well in detecting plagiarism. These methods have demonstrated their efficacy in detecting plagiarism when text has been plagiarized using various types of obfuscation. In contrast, while language processing based methods also effective in detecting plagiarism, these are computationally resource intensive as compared to lexical methods.

However, recent lines of research suggest that plagiarism detection approaches involving a combination of methods towards various types of obfuscation give improved performance as compared to methods from a single class of methods only (Section 2.4.7). This line of research can be considered as the next logical step in the development of plagiarism detection methods.

We present our research in the next chapter using this line of research, i.e. extending the idea of using an adaptive strategy by proposing a modular framework for detecting several types of obfuscation. The proposed approach in the next chapter is a natural extension to the current trend of developing hybrid approaches. This is followed by chapters that present research on obfuscation types, homoglyph substitution and paraphrase type identification, respectively.

Chapter 3

Cascade of Approaches for Plagiarism Detection

Research on plagiarism detection has produced several classes of methods including lexical and NLP based methods as described in Section 2.2. However, it has been observed that combinations of several classes of methods achieve better results than use of individual methods (Foltýnek, Meuschke, and Gipp, 2019). In the context of plagiarism detection research, the PAN plagiarism detection labs from 2009-2014 have provided a standard framework for evaluating plagiarism detection techniques. Participants have used several types of approaches for the text alignment task in the PAN plagiarism detection track (Potthast et al., 2014). However, the best performing approach (Sanchez-Perez, Sidorov, and Gelbukh, 2014) in PAN-2014 used an adaptive strategy by selecting parameters based on the recognized obfuscation type.

This chapter describes our proposed framework for detecting plagiarism using a cascade of detection approaches corresponding to various obfuscation types. The advantage of using a cascade of approaches is the ability to integrate similar but distinct detection strategies corresponding to each obfuscation type. We use the Greedy String Tiling Algorithm (Wise, 1995) as the seeding strategy. Furthermore, approaches corresponding to each obfuscation type, i.e., no-obfuscation (copy-paste), summary obfuscation and random and translation obfuscation, are integrated as a block of cascading if-else statements. Results show state-of-the-art scores for no-obfuscation (copy-paste) and summary obfuscation as compared to the current best performing results.

This chapter is organized as follows: Section 3.1 provides a brief introduction to the task of textual alignment for extrinsic plagiarism detection. Section 3.2 presents an overview of the proposed framework with an outline of the corresponding detection approaches. Section 3.3 provides a detailed description of each approach thereby elucidating the overall methodology. Section 3.4 provides results on the PAN-2014 test corpus and a comparison with other approaches on the test corpus. Section 3.5 concludes this chapter with future directions.

3.1 Introduction

The textual alignment problem in extrinsic plagiarism detection is based on aligning passages in a given source document with corresponding passages in the plagiarised (suspicious) document. The outcome of this stage of analysis is one or more pairs of passages from the source document and the suspicious document, highlighting boundaries of (possibly) plagiarised passages. For a given approach in textual alignment, F_1 score is calculated as the harmonic mean of macro-averaged precision and recall. The *granularity* of an approach can be defined as the average size of detections corresponding to plagiarized sections of text (Potthast et al., 2010). Finally, evaluation is carried out using the **plagdet** (Potthast et al., 2009) score which is calculated from macro-averaged F_1 score and granularity as follows.

$$plagdet = \frac{F_1 \quad score}{1 + log_2(granularity)}$$

The rationale of dividing the F_1 score by the $1 + log_2(granularity)$ is to assign a higher score to results that report plagiarism as integrated results, as opposed to approaches that report the results in a fragmented manner. Figure 3.1 shows a snapshot of our corpus viewer software, a utility developed to view text alignment of passages in the source document (left) and the suspicious document (right) for various PAN corpora.

Both the PAN-2013 and PAN-2014 textual alignment corpora consist of five different types of obfuscation (Potthast et al., 2013).

1. **No Plagiarism**: This type of obfuscation represents pairs of documents with no plagiarism at all. However, minor textual overlap is present which might present entities, names, places or events.
2. **No Obfuscation**: This represents copy-and-paste plagiarism, i.e., the copied text is reproduced verbatim in the suspicious document.
3. **Cyclic Translation Obfuscation**: Cyclic translation obfuscation represents repeated automatic translation of text from a source language into a number of target languages, sequentially (e.g., from language X to language Y to language Z and back to language X). The final language of translation is the source language itself, thereby representing plagiarised text.
4. **Random Obfuscation**: This represents random operations on source text such as insertion, shuffling, adding, deleting and replacing words at random. However, randomly obfuscated text is not human readable and has no semantics.
5. **Summary Obfuscation**: Summary obfuscation represents an unattributed automatic summary of text from the source document using automatic text summarization tools.

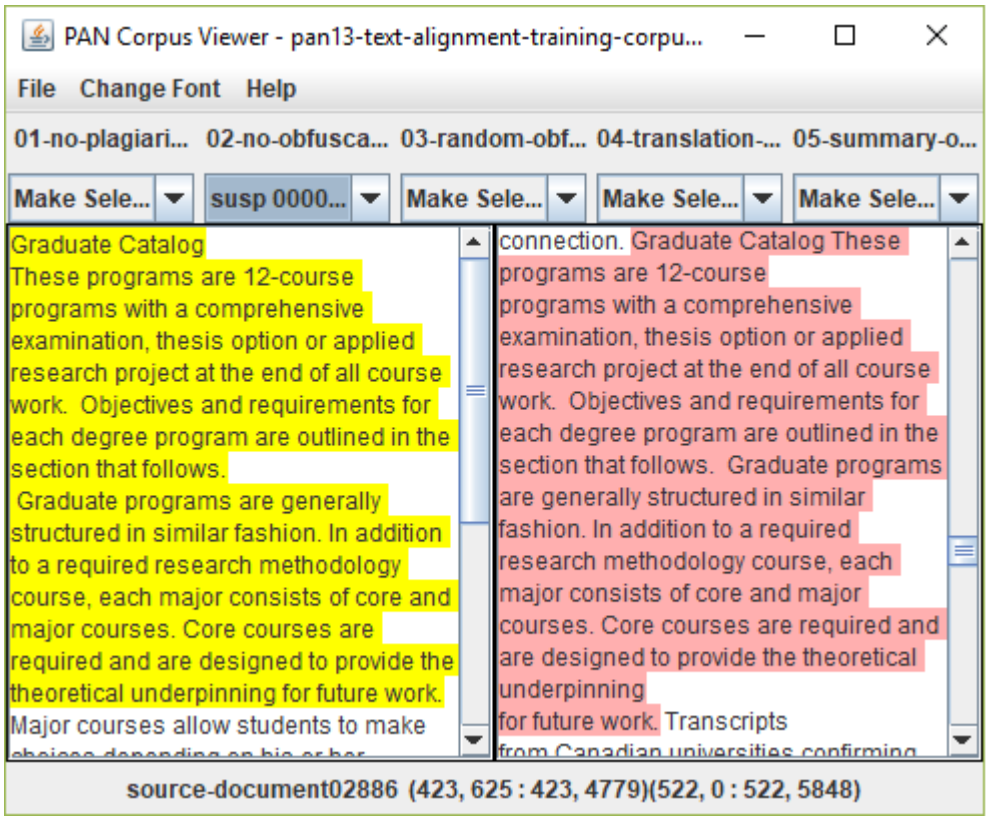


FIGURE 3.1: CORPUSVIEWER: Passage Alignment in the PAN-2013 Training Corpus showing the No-obfuscation Plagiarism Type

3.2 Overview of the Proposed Approach

Plagiarism detection approaches for the textual alignment task are based on the seed-and-extend paradigm, which consists of the three stages of seeding, extension and filtering. Several approaches have been employed for the seeding phase, including character and word *n*-gram similarity, alignment algorithms and string tiling. The extension and merging phases can be based on rules specific to the obfuscation type, while the filtering phase depends on cutoff thresholds. These distinct phases (Potthast et al., 2014) can be described briefly as follows.

1. *Seeding*: Seeding is the process of identifying 'seeds' that represent textual matches between the source and the suspicious document. These text matches are either exact or approximate matches using domain-specific or linguistic information .

2. *Extension*: Seed matches found in the previous phase are extended to aligned passages in the source and suspicious documents using some rule based heuristics. These aligned passage pairs form the source and plagiarised pairs of passages in the document collection.

3. *Filtering*: Filtering is used to remove out certain detected passages that do not satisfy certain criteria. The purpose of this phase is to remove false positives or other smaller matches that do not contribute towards plagiarism.

3.2.1 Brief Description of the Framework

Figure 3.2 shows the block diagram of the proposed framework of approaches. This framework also consists of the three phases of seeding, extension and filtering which have been clearly identified in the diagram along with a preprocessing phase. The input to the approach is a list of source-suspicious file pairs, sorted according the source file names. Each file pair goes through a common preprocessing stage where punctuation and whitespace are removed.

3.2.2 Seeding

We use the Greedy String Tiling Algorithm (Wise, 1995) with words as units for our seeding strategy. Our goal here is the detection of maximal segments of exact matches as seeds and the GST Algorithm is appropriate for this purpose. The Greedy String Tiling Algorithm detects common substrings between the source and the suspicious files in decreasing order of length, as described in Section 2.4.2. Both minimum and maximum lengths of these strings can be adjusted in terms of the number of words. However at the seeding stage we find all common strings consisting of at least three words (i.e., word n-grams of size 3 or higher).

3.2.3 Extension

String tiles detected as seeds are then sent as input to a cascading block of modules, where each module attempts to detect plagiarism based on a particular obfuscation type. The overall obfuscation detection scenario is based on a specific order of detection as shown in Figure 3.2 in the extension phase, described as follows:

1. The first module attempts to detect plagiarism for the no-obfuscation (copy-paste) type. If plagiarism is detected, control flow transfers to filtering (Step 5), else proceed to Step 2.

2. The second module attempts to detect plagiarism for the summary obfuscation type. If plagiarism is detected, control flow transfers to filtering (Step 5), else proceed to Step 3.

3. The third module attempts to detect plagiarism for both the random and translation obfuscation types. If plagiarism is detected, control flow transfers to filtering (Step 5), else proceed to Step 4.

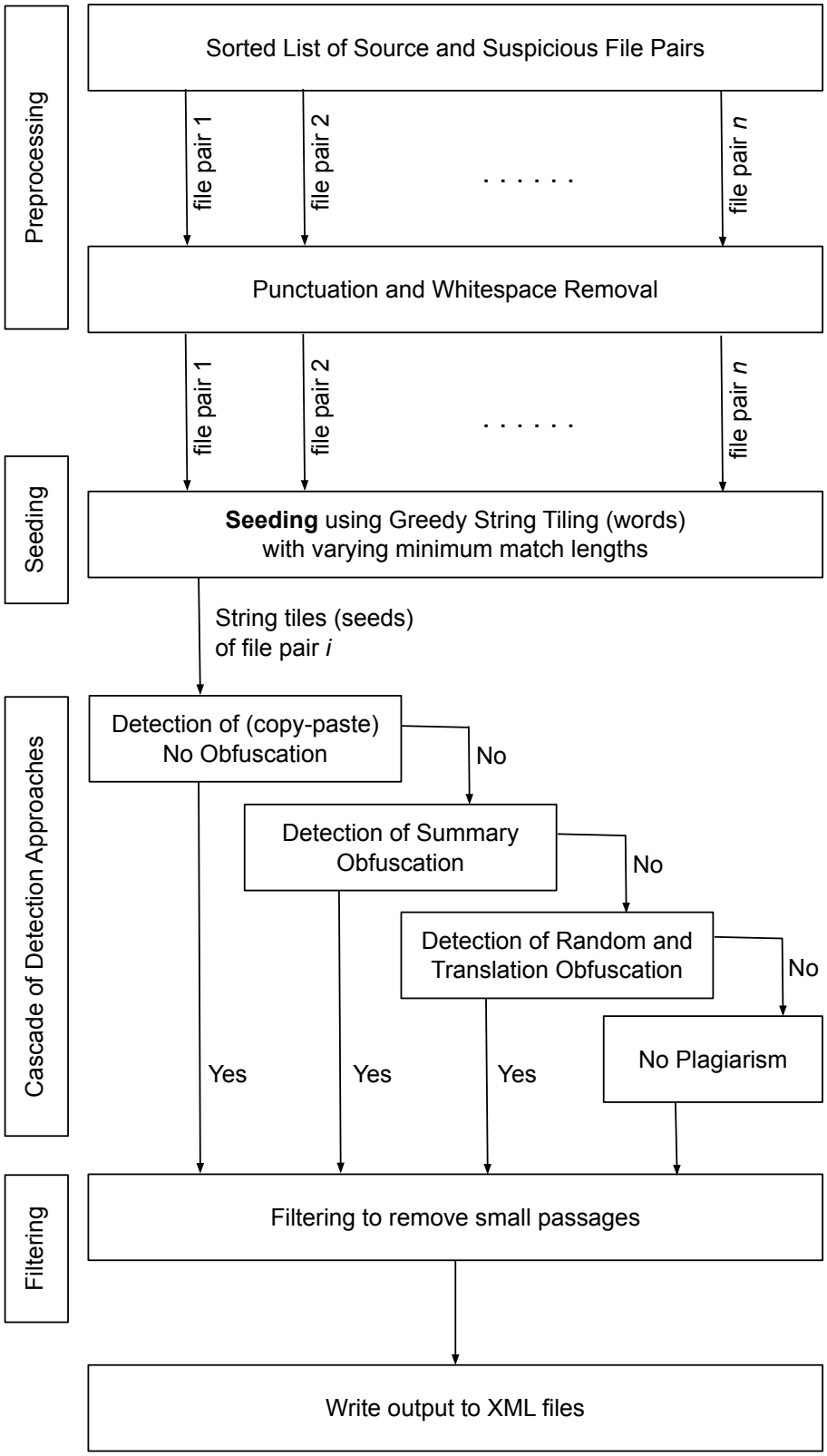


FIGURE 3.2: Block Diagram of the Cascade Framework

4. If no plagiarism is found in the preceding three steps, report the file pair as a No-plagiarism pair.

5. Apply passage filtering to remove short passage pairs.

6. Write the output as XML files.

For a given file pair, the cascade of approaches attempts to detect plagiarism in a particular order, as follows: 'no obfuscation' $\rightarrow$ 'summary obfuscation' $\rightarrow$ 'random/translation obfuscation' $\rightarrow$ 'no plagiarism'. The rationale for the selection of this order is based on the relative lengths of string tiles found between the source and suspicious documents and the distance between matching tiles corresponding to each obfuscation type. Below, we describe the details of conditions corresponding to the detection approaches for each obfuscation type in more detail.

1. **No Obfuscation:** For a matching string tile to be considered as a case of verbatim copy-paste, it has to be of a certain minimum length. This is because shorter string tiles might correspond to common names, entities or word sequences. For the PAN-2014 training and test corpora, we consider this length to be a minimum contiguous sequence of 40 words, found during seeding after experimenting with several values of length ranging from 20 to 50 words.

2. **Summary Obfuscation:** String tiles found for other obfuscation types are of a much shorter length. Therefore, the criterion for considering a particular document pair as a case of summary obfuscation depends on the distance between matching tiles within the source document as opposed to this distance within the suspicious document. This is because a summary is a condensed version of a longer document with key ideas and statements spread throughout the source document, but confined in a short passage within the suspicious document. Hence we expect that matching tiles within the source document are spaced at a greater distance between each other as compared to the distance between the corresponding tiles within the suspicious document.

3. **Random and Translation Obfuscation:** Finally, string tiles found within the random and translation obfuscations are also of a much shorter length but placed closer together. These matching tiles constitute plagiarised text that has been modified, relocated or complemented using other words, thereby inhibiting the formation of longer matching textual segments. Furthermore, as these represent plagiarism of contiguous segments of text, we expect these tiles to be found at a much shorter distance within both the source and the suspicious documents, as compared to summary obfuscation.

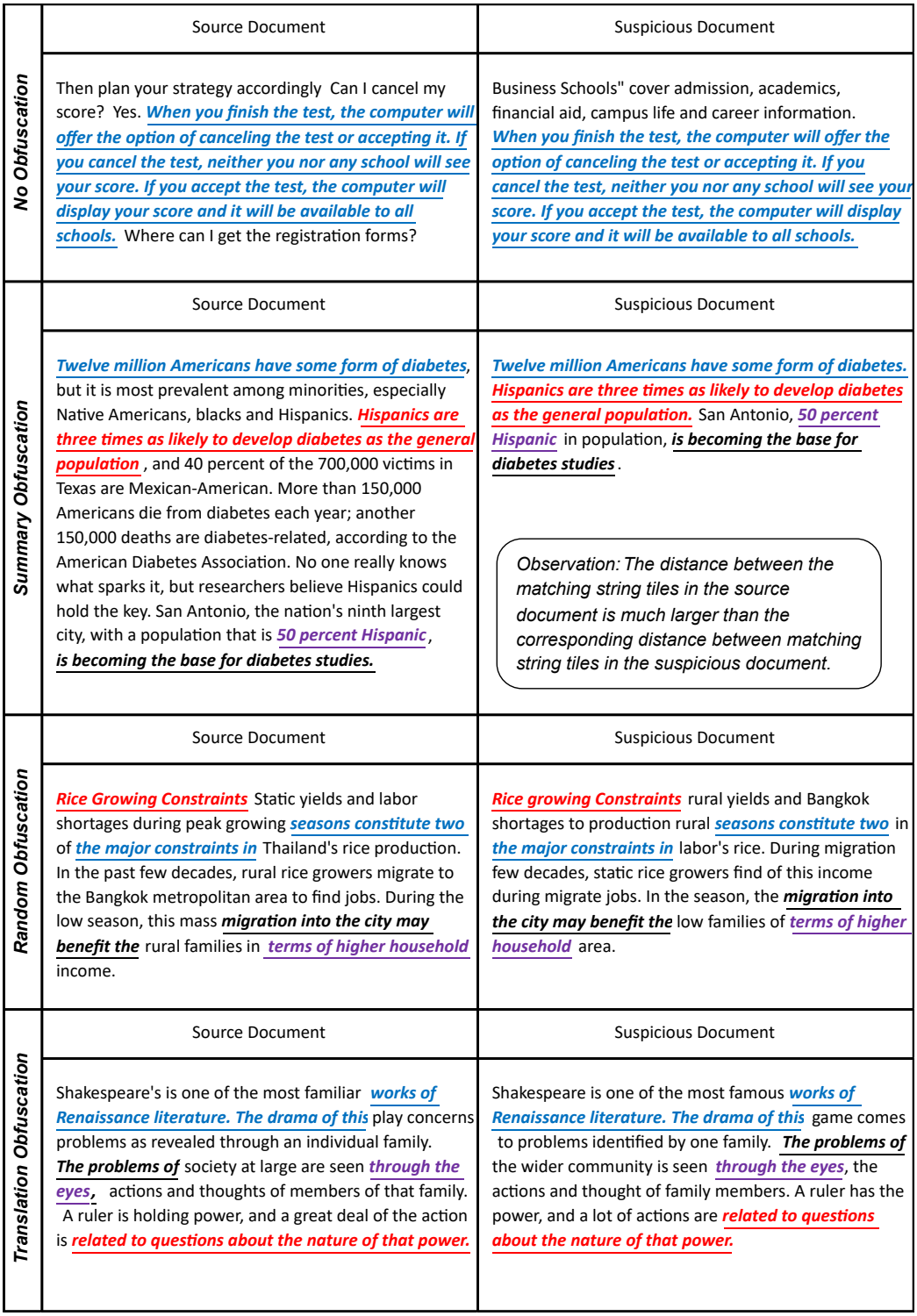


FIGURE 3.3: Placement of String Tiles (size 3 or higher) within Various Obfuscation Types

Figure 3.3 shows examples of string tile lengths and distances for various obfuscation types. It is important to mention here that the underlying assumption in the previously mentioned procedure is that each file pair consists of exactly one type of

obfuscation as found in the PAN corpora. However, the procedure can be easily generalized to include file pairs with multiple obfuscation pairs, by relaxing the if-else clauses in the cascade of approaches. Another point worth mentioning is that the order of recognizing obfuscation types is based on decreasing tile length followed by decreasing inter-tile distance.

3.3 Detailed Description of Approaches

In this section we provide a detailed description of each of these approaches:

3.3.1 No Obfuscation

For the no-obfuscation plagiarism detection, we find matching string tiles using the greedy string tiling algorithm. File pairs having string tiles greater than or equal to a minimum length threshold are considered as no-obfuscation cases. For the PAN-2014 test corpus, this length is considered to be 40 words. This value of a minimum of 40 words was found after testing with several values ranging from $30 \leq$ seed length ≤ 50. For the PAN-2014 test corpus, seed length = 40 classified 99.9% (999 out of 1000 files for no obfuscation) correctly. This procedure is applied repeatedly to detect multiple instances of copied passages.

3.3.2 Summary Obfuscation

Summarization of a document is the process of producing a short abstract identifying the key ideas and statements of the original document. The primary objective in automatic summarization is "*to extract content from an information source and present the most important content to a user in a condensed form*" (Mani, 2001). Extractive summarization represents the process of selecting important parts of text to produce a summary of a given document. In contrast, abstract summarization is the process of writing a summary based on a writer's understanding of the entire text, primarily in their own words.

From a plagiarism detection perspective, a summary obfuscation can be detected if exact matches of text are found spread apart in the entire source document, but placed closer together in the suspicious document. This can be observed from Figure 3.3, where the distance between matching string tiles in the summary is much shorter than the corresponding distance between matching tiles in the source.

The standard deviation of a variable is a measure of the spread of a parameter from its mean value and can be utilized as a measure for detecting summary obfuscation. For this purpose, we use the difference of the standard deviation of the

distances between the source document tiles and between the suspicious document tiles as a metric to determine the presence of summary obfuscation.

More formally, let d_{1src}, d_{2src}, ..., $d_{(k-1)src}$ be the distances in terms of the number of characters between k matching tiles in the source document. Likewise, let d_{1susp}, d_{2susp}, ..., $d_{(k-1)susp}$ be the corresponding distances between k matching tiles in the suspicious document. The standard deviation of these distances in the source and the suspicious documents is given by,

$$\sigma_{src} = \sqrt{\frac{\sum_{i=1}^{k-1}(d_{isrc} - \overline{d}_{src})^2}{k-1}}, \qquad \sigma_{susp} = \sqrt{\frac{\sum_{i=1}^{k-1}(d_{isusp} - \overline{d}_{susp})^2}{k-1}}$$

where $\overline{d}_{src}, \overline{d}_{susp}$ represent the average of the distances in the source and the suspicious documents, respectively. We consider the following criteria for deciding on the existence of summary obfuscation in the PAN-2014 test corpus stated as follows: if $|\sigma_{src} - \sigma_{susp}| \geq 150$ characters, then the document can be considered to have summary obfuscation. This value was arrived after testing with a range of values from 100 to 200 characters in steps of 10, with the highest plagdet score (exceeding the current best performing result by 10%) found for $|\sigma_{src} - \sigma_{susp}| \geq 150$ characters. Therefore, the set of criteria for the cascade of approaches is shown in Figure 3.4.

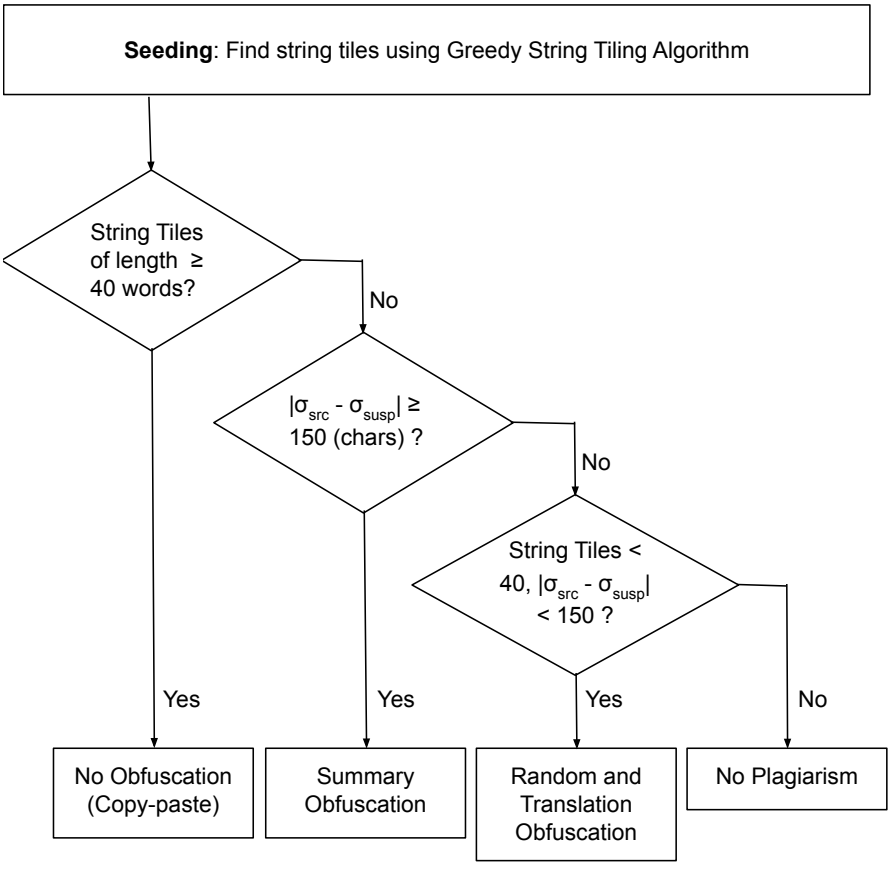


FIGURE 3.4: Criteria for Selecting Obfuscation Type in the Cascade of Approaches

3.3.3 Random and Translation Obfuscation

We group the approaches for translation and random obfuscation, as it can be observed that the string tiles for both these types are generally at a shorter distance in both the source and suspicious documents (Figure 3.3). In particular, if the length of string tiles < 40 words, and $|\sigma_{src} - \sigma_{susp}| < 150$ characters, then we consider the document pair to have a case random or translation obfuscation. For this case, we use the merging phase described in our earlier PAN-2014 approach (Alvi, Stevenson, and Clough, 2014) but with word tiles instead of character-n grams as seeds.

Once the list of seeds (matching string tiles) is generated between a pair of source and suspicious documents, the next step is the extension and subsequent merging of these matches into contiguous, aligned passages. This is done by combining seeds or matches that satisfy certain criteria both in the source and suspicious documents into larger matches. In our earlier approach, we identified four distinct categories of matches as follows based on their vicinity towards each other. Figure 3.1 shows the four categories for extension and merging.

Consider two matching pairs of seeds, i.e., $(x_1, y_1, s_1) \rightarrow (a_1, b_1, s'_1)$, $(x_2, y_2, s_2) \rightarrow (a_2, b_2, s'_2)$ where (x_1, y_1, s_1) indicates text from position x_1 to position y_1 of size s_1 in the source document, and (a_1, b_1, s'_1) indicates the corresponding text from position a_1 to b_1 of size s'_1 in the suspicious document. In order to merge these pairs in each of the source and suspicious documents, we identified four categories of matches as follows:

1. **Containment**: Containment is the relationship between two matches within the same document when the text positions of one match are fully contained within the text positions of the other. More formally, the match (x_2, y_2, s_2) is contained within (x_1, y_1, s_1) if $x_2 \geq x_1$, $y_2 \leq y_1$ and the size $s_1 \geq s_2$.

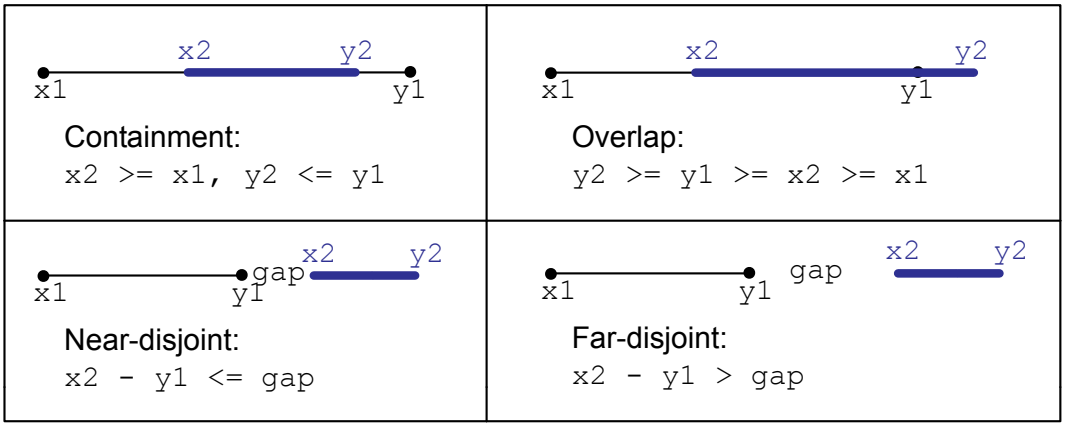


FIGURE 3.5: Classes of Matching Pairs

2. **Overlap**: Two matches (x_1, y_1, s_1) and (x_2, y_2, s_2) overlap if some, but not all text positions of the first one are contained within the text positions of second match. More formally, the match (x_1, y_1, s_1) overlaps (x_2, y_2, s_2) if $y_2 \geq y_1 \geq x_2 \geq x_1$ i.e., x_2 lies between x_1 and y_1, y_1 lies between x_2 and y_2.

3. **Near-Disjoint:** Two matches are near-disjoint, if they are close enough to be combined into a single chunk of text, i.e., separated by a small number of characters, but have no overlapping text-positions. More formally if (x_1, y_1, s_1) and (x_2, y_2, s_2) are two matches such that $0 \leq x_2 - y_1 \leq gap$, where the parameter gap depends on the typical plagiarised passage size within the training corpus, then the matches may be categorized as near-disjoint.

4. **Far-Disjoint**: Two matches (x_1, y_1, s_1) and (x_2, y_2, s_2) are far-disjoint if they are separated by such a large number of characters that they cannot be adequately merged into a single chunk of text. In this case $x_2 - y_1 > gap$.

3.3.4 Strategy for Merging

Table 3.1 gives a case-by-case listing of the merging strategies that we used for merging. The overall strategy for merging is based on the idea that two matching pairs that have either one of the containment, overlap or near-disjoint relationship can be merged towards forming a larger match. Matches that are far-disjoint cannot be merged. Furthermore, if two mapping pairs of 3-tuples have a different relationship in the source and suspicious documents then these tuples may be combined according to their respective categories. However, in a particular case of 3-tuples, i.e., when the 3-tuples have a near-disjoint relationship in one document and overlap or containment relationship in the other document, then no merging may be carried out. This is because it is a likely case of *term-repetition*, a situation in which a term is repeated within one of the documents, but it may mistakenly map to a large portion of text in the other document.

3.3.5 Filtering

After seeding and merging, the next step is filtering. For this step, short passage matches typically less than 100 characters resulted in a lot of false positives. Hence all passages which were less than 100 characters in the source document aligned with passages less than 100 characters in the suspicious document were removed. The final output was then written to the XML files.

TABLE 3.1: Strategies for Merging Various Cases

Relationship in Source $(x_1, y_1, s_1), (x_2, y_2, s_2)$	Relationship in Suspicious $(a_1, b_1, s'_1), (a_2, b_2, s'_2)$	Replacement Action Replacement 3-tuple
Containment: (x_1, y_1, s_1) contains (x_2, y_2, s_2)	Containment	$(x_1, y_1, s_1) \rightarrow (a_1, b_1, s'_1)$
	Overlap	$(x_1, y_1, s_1) \rightarrow (a_1, b_2, b_2 - a_1)$
	Near-disjoint	No change (Term Repetition likely)
	Far-disjoint	Merging not possible
Overlap: (x_1, y_1, s_1) overlaps (x_2, y_2, s_2)	Containment	$(x_1, y_2, y_2 - x_1) \rightarrow (a_1, b_1, s'_1)$
	Overlap	$(x_1, y_2, y_2 - x_1) \rightarrow (a_1, b_2, b_2 - a_1)$
	Near-disjoint	No change (Term Repetition likely)
	Far-disjoint	Merging not possible
Near-disjoint: $x_2 - y_1 \leq gap$	Containment	No change (Term Repetition likely)
	Overlap	No change (Term Repetition likely)
	Near-disjoint	$(x_1, y_2, y_2 - x_1) \rightarrow (a_1, b_2, b_2 - a_1)$
	Far-disjoint	Merging not possible
Far-disjoint $x_2 - y_1 > gap$	Containment	Merging not possible
	Overlap	
	Near-disjoint	
	Far-disjoint	

3.4 Results and Discussion

The proposed approach was tested for plagiarism detection on the PAN-2014 text alignment test corpus consisting of the five obfuscation types. The PAN-2014 text corpus consists of 5185 source-suspicious document pairs. Results are available which include precision, recall and plagdet scores of our approach as well from other selected approaches in Table 3.2. The best performing plagdet scores for each obfuscation type are highlighted in bold.

	Obfuscation Type				
Approach	Metric	No Obf.	Random	Translat.	Summary
Alvi (2019)	Plagdet	**0.99686**	0.82503	0.70050	**0.69861**
	Precision	0.99815	0.85675	0.71252	0.68463
	Recall	0.99558	0.83985	0.73042	0.74777
Sanchez (2015)	Plagdet	0.98120	**0.88470**	0.87920	0.63040
	Precision	0.99330	0.89990	0.84810	0.97390
	Recall	0.97610	0.86990	0.91280	0.48620
Sanchez (2014)	Plagdet	0.90032	0.88417	**0.88659**	0.56070
	Precision	0.83369	0.91015	0.88465	0.99910
	Recall	0.97853	0.86067	0.88959	0.41274
Glinos (2014)	Plagdet	0.96236	0.80623	0.84722	0.62359
	Precision	0.96445	0.96951	0.96165	0.96451
	Recall	0.96028	0.72478	0.76248	0.48605
Oberr. (2014)	Plagdet	0.91976	0.86775	0.88118	0.36804
	Precision	0.85231	0.90608	0.89977	0.93581
	Recall	0.85231	0.90608	0.89977	0.93581
Shrestha (2014)	Plagdet	0.89174	0.86556	0.84384	0.15550
	Precision	0.80933	0.92335	0.88008	0.90455
	Recall	0.97438	0.83161	0.85318	0.08875
PAN Baseline	Plagdet	0.93404	0.07123	0.10630	0.04462
	Precision	0.88741	0.98101	0.97825	0.91147
	Recall	0.99960	0.04181	0.08804	0.03649

TABLE 3.2: Plagdet Scores for Selected Approaches on the PAN-2014 Test Corpus for each obfuscation type

From the results in Table 3.2 it can be seen that our proposed framework of cascade of approaches gives state-of-the-art results for no-obfuscation and summary obfuscation detection and mid-range results for random obfuscation and translation obfuscation.

This can be attributed to the specific structure of the framework where specific approaches can be used to detect a particular type of obfuscation, provided suitable criteria are available for the presence of particular obfuscation types. In this sense, this framework of approaches is inline with the current trend of using a combination of approaches in contrast to using an individual approach for all obfuscation types. Furthermore, even a lower score for translation obfuscation shows that the framework has the flexibility to produce state-of-the-art scores for some obfuscation type, but modest scores for other types due to its hybrid structure. Infact, given a suitable criterion for identifying translation obfuscation, the current approach can be easily replaced with the best performing approach for detecting translation plagiarism.

No Obfuscation: Upon further analysis, it can be found that for no-obfuscation plagiarism detection, repeated detection cycles for matching string tiles $\geq$ a particular threshold are the primary reason why our approach gave the best performing, almost near perfect result. This can be contrasted with most of the other approaches where only one cycle of detection might have been the reason for detection rates below the perfect score.

Summary Obfuscation: For summary obfuscation detection, our proposed approach was based on the definition of how extractive summaries are produced (i.e., by calculating the difference of the distances between source and suspicious documents' tiles). This approach outperformed all other approaches in the summary obfuscation detection by a wide margin. Further refinement of the parameters for seed size and difference threshold may produce a further improvement in scores.

Random and Translation Obfuscation: For random and translation obfuscation detection, we utilized our earlier PAN-2014 approach (Alvi, Stevenson, and Clough, 2014) of using merging heuristics integrated into the cascade framework but with a different seeding strategy. Compared to other approaches, our approach performed within the range of best approaches in the random obfuscation detection, but scored below the top performing approaches in the translation obfuscation. This score is inline with the earlier performance of this approach in PAN-2014. The use of this approach demonstrates the flexible structure of the cascade block, where this approach could easily be replaced by one of the best performing approaches.

Figures 3.6 and 3.7 gives a comparative performance of our approach vs. selection of approaches on the PAN-2014 corpus for each obfuscation type. It can be observed from these results that our approach outperforms other approaches for no-obfuscation and summary obfuscation.

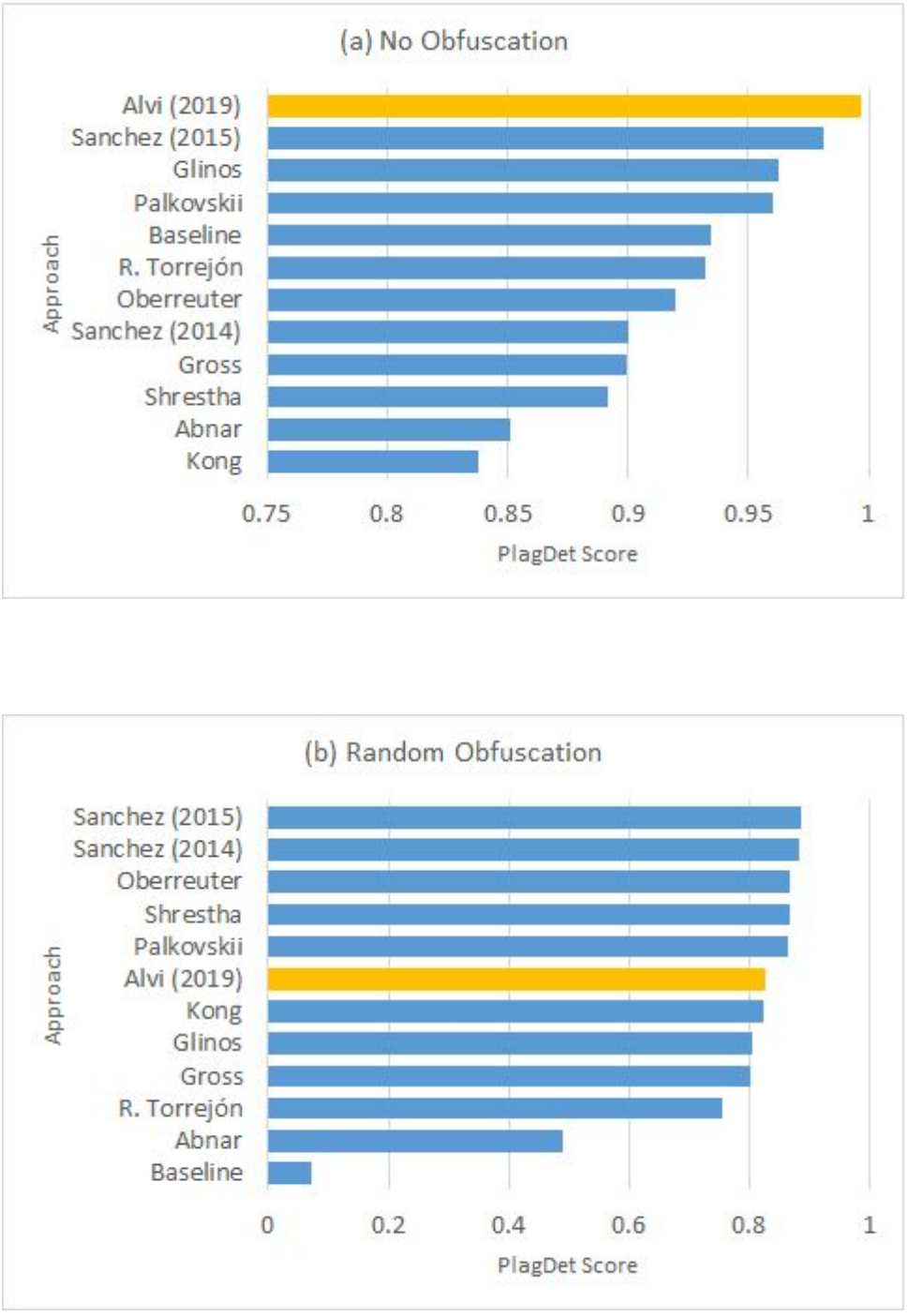
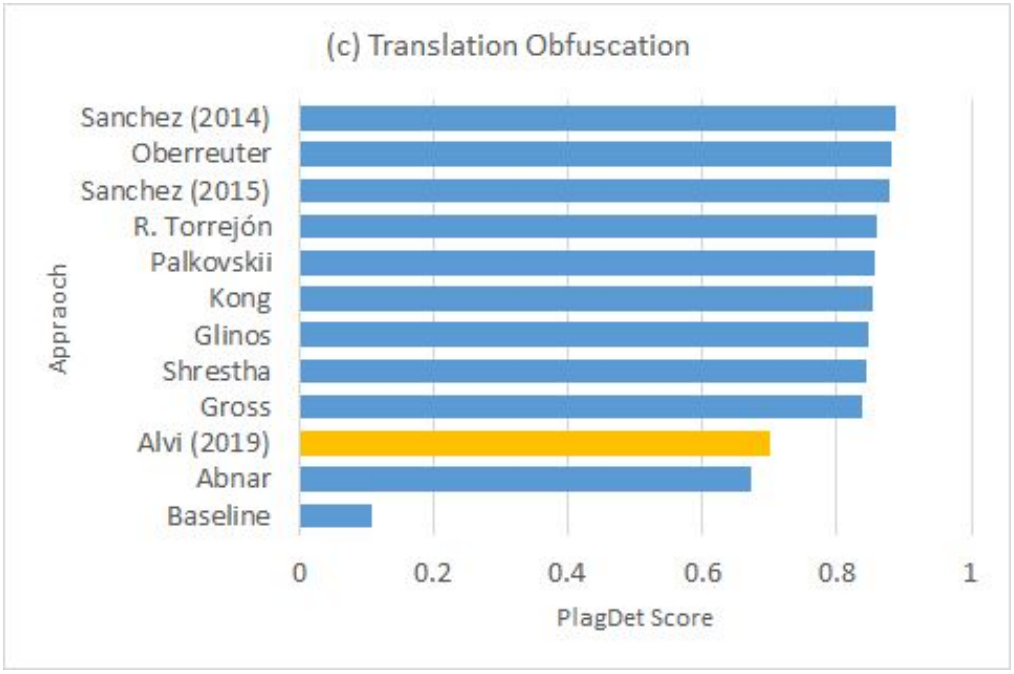


FIGURE 3.6: A Comparison of Plagdet Scores for (a) No Obfuscation and (b) Random Obfuscation



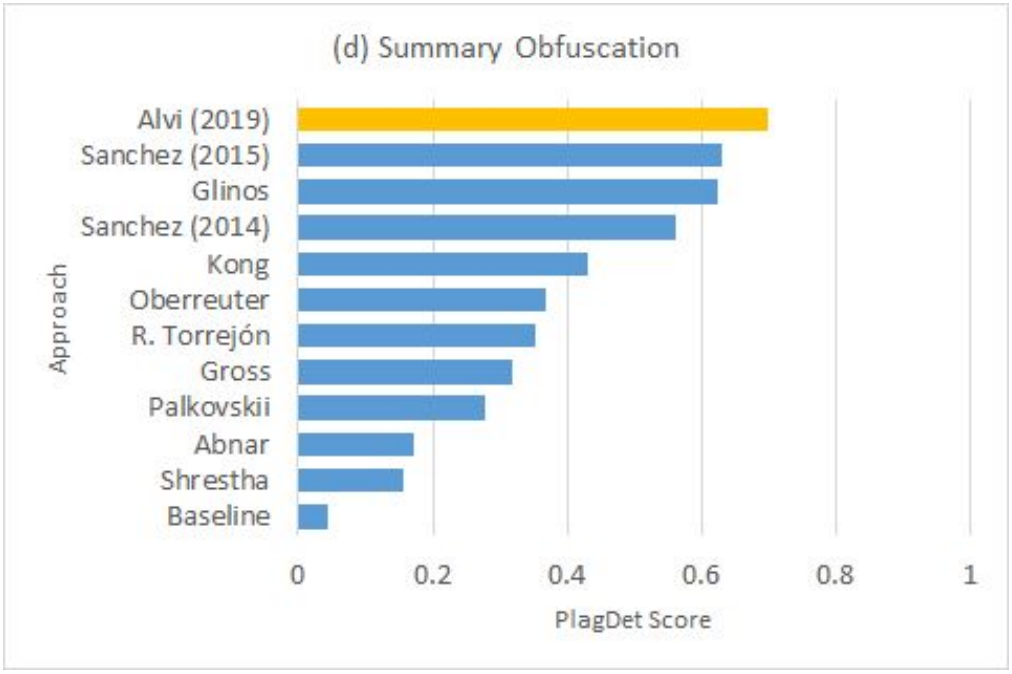


FIGURE 3.7: A Comparison of Plagdet Scores for (c) Translation Obfuscation and (d) Summary Obfuscation

3.5 Conclusion

In this chapter we presented a cascade of approaches for plagiarism detection when the set of document pairs contains plagiarism using various obfuscation types. Our proposed framework has a modular structure thereby having the ability to integrate customized approaches for various obfuscation types. This framework produced state-of-the-art results for no-obfuscation and summary obfuscation types.

The primary contribution of the proposed framework is cascaded detection structure corresponding to various obfuscation types based on decreasing seed size. This is an improvement over several approaches presented in PAN Evaluation Labs (Potthast et al., 2013, 2014) for various obfuscation types which mostly rely on a single or a collection of methods corresponding to various obfuscation types. Another contribution of the proposed framework is the approach presented for detecting summary obfuscation which is based on the standard deviation of the distances corresponding to the seeds in the source and summarised texts.

One limitation of the proposed structure the lack of a criterion distinguishing between translation and random obfuscation types. Another possible venue of improvement could come from investigating further distance metrics for detecting plagiarism in summary obfuscation. This leads to propose future work proposing further refinements of the proposed structure by, (a) finding distinguishing criterion for translation obfuscation, and (b) investigating distance metrics for summary obfuscation.

From a broader perspective, another challenge is whether the proposed structure can be scaled to include approaches corresponding to additional obfuscation types. Such an expansion in the scope of obfuscation types would lead to the creation of a more comprehensive plagiarism detection framework. Given the large number of approaches corresponding to various obfuscation types within the PAN Evaluation labs, this leads us to contemplate about proposing additional obfuscation types. This could then lead to fruitful research towards researching effective methods of plagiarism detection corresponding to additional obfuscation types and ways of integrating them.

In the next chapter we will address expanding the scope of obfuscation types to include novel obfuscation types for plagiarism detection. Using the large number of plagiarism detection approaches in the PAN repository, we examine the performance of detecting plagiarism on these newer types using PAN approaches.

Chapter 4

Generation of Additional Obfuscation Types

In the last chapter we presented a cascade of approaches for plagiarism detection with strategies for dealing with various types of obfuscations within the PAN framework. Research on detecting plagiarism on these obfuscation types, i.e., no (copy-paste), random, translation and summary obfuscation, have produced several approaches (Potthast et al., 2014).

As discussed in the previous chapter, plagiarism detection research on the obfuscation types within the PAN Evaluation Labs (Potthast et al., 2013, 2014) has produced high plagdet scores with a variety of techniques proposed by researchers. Therefore, there is a need to expend research efforts on broadening the scope of obfuscation types that might evade detection using the available plagiarism detection techniques. This gives rise to the following research questions: (a) What are some of the additional obfuscation types that might be employed by a plagiarist to obfuscate copied text, and (b) What is the effectiveness of the currently available plagiarism detection techniques in detecting plagiarism on these novel obfuscation types?

In this chapter we expand the scope of obfuscation types by creating a data source with new obfuscation types using the corpus construction task in PAN-2015. The main research objectives towards construction of this corpus were two-fold: (a) to propose newer obfuscation types as cases of simulated plagiarism using document pairs which describe similar content, but as different versions, and (b) to use the existing collection of plagiarism detection techniques within the PAN framework in order to assess the relative difficulty of detecting plagiarism within these obfuscation types.

This chapter describes the construction of our data source, i.e. 'The Short Stories Corpus' (Alvi, Stevenson, and Clough, 2015). This dataset is based on retold versions of Grimms' Fairy Tales, which were selected as these various versions can be used to simulate paraphrase plagiarism. The proposed dataset consists of four obfuscation types: (a) no-plagiarism, (b) story retelling, (c) synonym replacement, and (d) character (homoglyph) substitution. Two of these types are lexical (no-plagiarism and

character substitution), while the other two are semantic (story retelling and synonym replacement). In this sense, story retelling obfuscation can be considered as a simulated representation of paraphrase plagiarism.

Results of several of the available approaches for plagiarism detection within the PAN framework show that several of the approaches successfully detect plagiarism of the semantic obfuscation types. However, character (homoglyph) substitution has proven to be a simple but very effective technique at obfuscating text reuse, since a large number of approaches were unable to detect *any* plagiarism for this obfuscation type.

The rest of this chapter is organized as follows: Section 4.1 provides a brief introduction to the short stories corpus. Section 4.2 provides a detailed description of each obfuscation type with the methodology of constructing the data for these types. Section 4.3 provides an overview of the various reviews received for this corpus. Section 4.4 provides details of the results of various PAN approaches with respect to this corpus, as well as a discussion on the results. Finally, Section 4.5 concludes this work and sets the direction for the next chapter.

4.1 Introduction

The corpus construction task was initiated in PAN-2015 for text alignment in the plagiarism detection track. The objective of this task was two-fold: (a) to involve the participants in creating a data source with annotated passages involving real and/or artificially generated samples of text reuse or plagiarism with varying obfuscation types, and (b) to evaluate the difficulty of detecting plagiarism within these obfuscation types using the large PAN repository of plagiarism detection approaches.

For the creation of a data resource, we considered the idea of a domain specific resource involving text reuse. Text reuse detection has already been a well researched area in the news domain (Clough et al., 2002). We extended the idea of text reuse in the news domain to textual similarity in retold versions of short stories. This resulted in the creation of our data source using retold versions of Grimms' Fairy Tales, called 'The Short Stories Corpus' (Alvi, Stevenson, and Clough, 2015). The source documents' passages were taken from various translations of Grimms' fairy tales and are available on the Project Gutenberg[1] website.

The corpus consists of 200 document pairs with 50 document pairs each within the following four obfuscation groups,

1. no-plagiarism,

2. (human) story-retelling,

[1]https://www.gutenberg.org/ebooks/2591 [Last Accessed: 29-June-2020]

3. synonym-replacement, and

4. character-substitution.

The no-plagiarism group consists of completely different short stories that may share some genre-specific terms leading to minor textual overlap. The story retelling group describes pairs of story fragments taken from two different retellings by human writers. The third group, synonym replacement, describes story fragment pairs with replacement of words and phrases with their synonym equivalents. Finally, character substitution refers to technical disguise, where letters in words are replaced with their look-alike unicode equivalent characters, or *homoglyphs*.

4.2 Corpus Construction

The corpus is composed of documents from the Grimms' fairy tales as available on Project Gutenberg website. The corpus is small in comparison to other PAN corpora. This is because the number of tales available within the Grimms' collection ranges from a maximum of 200 in some editions to less than 50 within other editions. This availability of tales in Grimms' collection restricts the size of a corpus from 50 to 200 document pairs. In order to have a balanced collection of documents within each group, our corpus consists of 200 document pairs, with 50 pairs for each group. Some statistics related to the passage length within the corpus documents are shown in Table 4.1.

From the Table it can be seen that the average length of passages in each of the three obfuscation types of story retelling, synonym replacement and character substitution are within the range 450–600 characters with the highest passage length in story retelling. For the 'No Plagiarism' group, there is no maximum, minimum or average length of passages since there are no plagiarised passages between the documents.

TABLE 4.1: Statistics of Passage Sizes in the Corpus (characters)
(*Sizes of no plagiarism are none since plagiarised passage sizes are zero)

Passage Length	No Plagiarism*	Story Retelling	Synonym Replacement	Character Substitution
Number of Docs.	50	50	50	50
Maximum Length	none	1160	765	729
Minimum Length	none	285	259	220
Average Length	none	**590**	**497**	**455**

Here, a passage is defined as a contiguous maximal-length sequence of characters (or text) that consists of similar text between two versions. For corpus construction, we selected passages from two versions of a story that correspond to the same events, since different versions of the same story may sometimes differ in details of events.

In the following subsections, we describe each of the obfuscation type groups in more detail:

4.2.1 No plagiarism

The no-plagiarism obfuscation type has been present in both PAN-2013 and PAN-2014 text alignment datasets. We included it in the short stories corpus in order to validate the format of our dataset. In particular, by ensuring a successful (error-free) run of various approaches on the no plagiarism type, we can verify that the format of our corpus is entirely correct.

For the no-plagiarism group, we included short stories that have a different theme but may have some textual overlap. This overlap may come from genre-specific words e.g. characters or places such as *the king*, *queen*, *forest*, etc. or actions such as "*looked at*", "*said to*", etc. However, despite this overlap, the stories selected for the no plagiarism type have a completely different storyline, and therefore the similarity score should be close to zero.

Construction of the No Plagiarism Obfuscation type

For constructing this group, the ferret trigram similarity (Lane, 2011) was computed for the the entire Grimms' collection. We then selected 50 document pairs which had a non-zero trigram similarity, (i.e., some genre-specific textual overlap) but having a completely different theme between them, thereby implying no plagiarism. Figure 4.1 shows a comparison of two such documents. It can be observed that despite some matching terms (such as *"answered the old woman"*, *"went into the"*), which can possibly serve as seeds for a given plagiarism detection approach, overall there is no plagiarism between the two documents.

4.2.2 Story Retelling

Fairy Tales are oral traditions that have been retold for generations and are still favourites. Most fairy tales have a good number of versions, providing a rich collection of various retellings. Unlike some modern short stories, these are in the public domain and so freely available with no copyright restrictions. However, some retellings may contain archaic language not in use today.

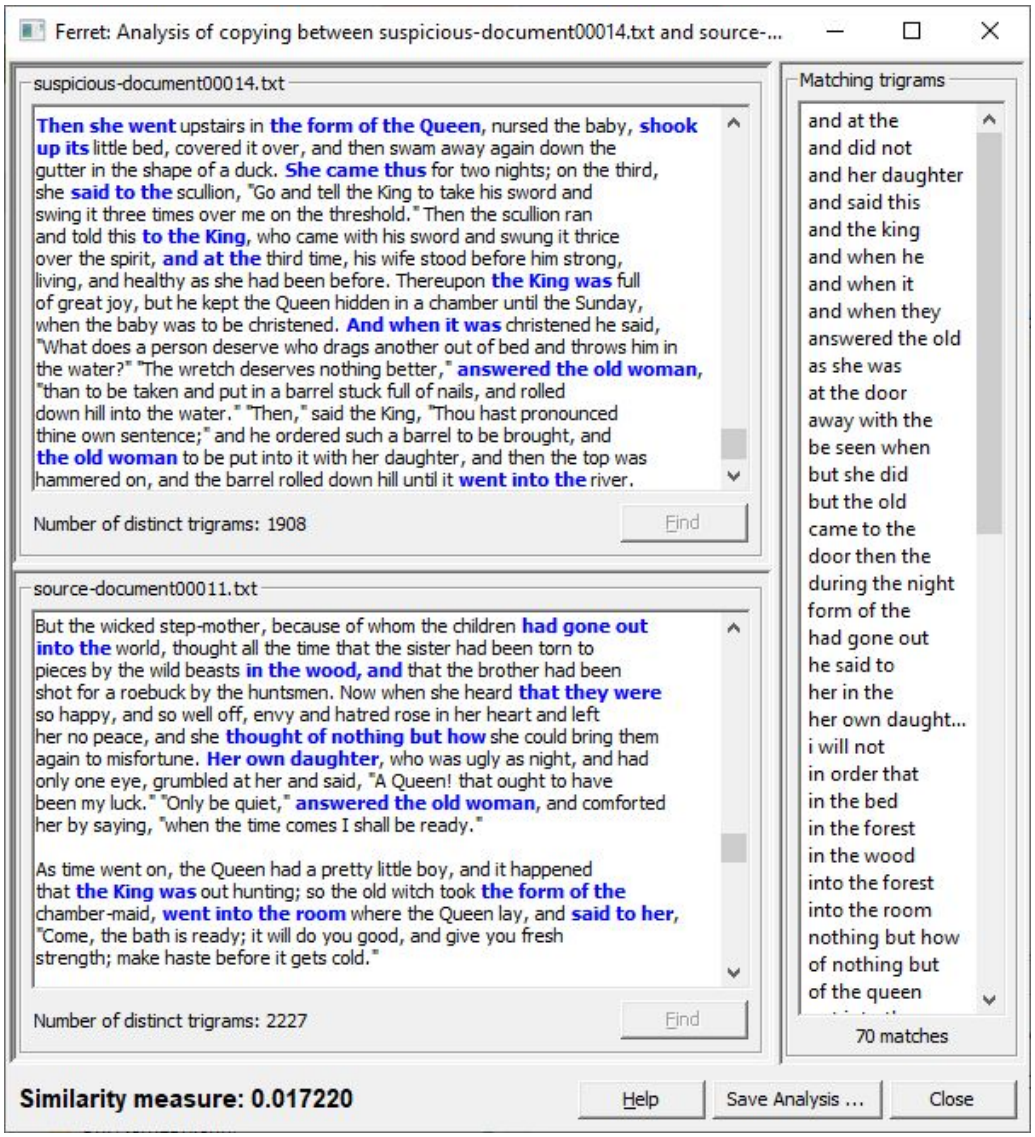


FIGURE 4.1: Ferret Trigram Similarity of Suspicious Document 00014 with Source Document 00011

Story retelling is defined as *"a new, and often updated or retranslated, version of a story."*[2]. A retold story may borrow various elements, for example characters and parts of the storyline from the original version. Consequently, a story retelling may reuse text and language from the original story thus effectively being a paraphrased version of the original.

Considering the relationship between story retelling and paraphrasing we can consider retold versions of various stories as simulated cases of paraphrase plagiarism. This is because a story retelling involves rephrasing text using various text rewrite operations as well as addition or removal of certain elements. This position is supported by the remark stated by Clough et al. (2002), *"Of course, reusing language is as old as the retelling of stories..."*. This remark provides a firm indication of

[2]http://dictionary.reference.com/browse/retelling [Last Accessed: 24-May-2019]

Retold Story Version 1

It happened that ***the wedding of the King's eldest son was to be celebrated***, so the poor woman went up and placed herself by the door of the hall to look on. When all the candles were lit, and people, each more beautiful than the other, entered, and all was full of pomp and splendour, ***she thought of her lot with a sad heart*** , and cursed the pride and haughtiness which had humbled her and brought her to so great poverty.

The smell of the delicious dishes which were being taken in and out reached her, and now and then the servants threw her a few morsels of them: these ***she put in her jars to take home***.

Retold Story Version 2

She had not been there long before she heard that ***the king's eldest son was passing by, going to be married;*** and she went to one of the windows and looked out. Everything was ready, and all the pomp and brightness of the court was there. Then ***she bitterly grieved*** for the pride and folly which had brought her so low. And the servants gave her some of the richmeats, which ***she put into her basket to take home***.

FIGURE 4.2: Example of Textual Similarity in Story Retellings

text reuse in story retellings.

Therefore, we consider various retellings of fairy tales as a rich resource of paraphrased texts that can be considered as simulated versions of paraphrase plagiarism. In case of Grimms' fairy tales, it is important to mention here that the original Grimms' tales are in the German language, with various retellings including translations in the English language.

In Figure 4.2 we show fragments from two retellings of the story 'King Thrushbeard' (also called 'King Grisly Beard'). Here we give a correspondence between sentence fragments found between the two retellings.

1. (Fragment 1) The wedding of the King's eldest son was to be celebrated.
   (Fragment 2) The king's eldest son was passing by, going to be married.

2. (Fragment 1) She thought of her lot with a sad heart.
   (Fragment 2) She bitterly grieved.

3. (Fragment 1) She put in her jars to take home.
   (Fragment 2) She put into her basket to take home.

We see that in pair 1, fragment 2 corresponds to a paraphrase of fragment 1 with some rearrangement of words. In pair 2, fragment 2 corresponds to a summarization of fragment 1. For pair 3, a substitution of 'jars' with 'basket' corresponds to a substitution of one type of container with another, effectively a same polarity substitution.

Construction of the Story Retelling Obfuscation type

For the construction of this obfuscation type, we chose passages from two different retellings of the same story by two different authors. These passages represented the same event or idea, and we ensured that passage lengths were generally comparable, i.e., did not correspond to a summarization. These were then embedded into other unrelated stories in order to provide similarity of context. We also ensured that the embedding stories (i.e., the outer fairy tales containing the passages) do not have textual similarity with each other.

4.2.3 Synonym Replacement

The third group in our corpus is synonym replacement. This refers to replacement of words and phrases with synonymous words and equivalents. Synonym substitution is one of the most common operations in plagiarism as discussed in Section 1.1. Therefore, this obfuscation type can be considered as a special case of story retelling where words and phrases are replaced with synonymous terms. However, we do not rearrange words and phrases, thereby leaving the contexts of words and phrases unchanged. For this obfuscation type, we created a genre-specific list of synonymous terms of approximately 900 unique content words in the corpus. In addition to these, we also removed some articles (a, an, the), and replaced alternate occurrences of some pronouns with proper nouns. Below we give an example of synonym replacement:

1. (Fragment 1) The King, who had a bad heart, and was angry...
2. (Fragment 2) The monarch, who had a worse heart, and was enraged...

In the above pair of sentence fragments, the following word substitutions have been made: "king ↔ monarch", "bad ↔ worse" and "angry ↔ enraged".

Construction of the Synonym Replacement Obfuscation Type

Given a passage from a source document, selected words and phrases from the source passage were replaced with genre-specific synonymous words and phrases. These passages were then embedded into unrelated stories/documents as done for the story retelling type. Table 4.2 shows an example of suspicious and source documents (side-by-side) with synonym replacements. For example, the following replacements can be observed in the document pairs: "lived ↔ dwelled", "brothers ↔ siblings", "innocent ↔ not guilty", etc.

TABLE 4.2: Example of Synonym Substitution

Source Text	Suspicious Text
Now there **lived** in the **country** two **brothers**, sons of a **poor man**, who **declared** themselves willing to undertake the **hazardous enterprise**; the **elder**, who was **crafty** and **shrewd**, out of **pride**; the younger, who was **innocent** and simple, from a kind heart.	Now there **dwelled** in the **kingdom** two **siblings**, sons of a **impoverished human**, who **announced** themselves willing to undertake **dangerous adventure**; the **older**, who was **astute** and **cunning**, out of **arrogance**; younger, who was **not guilty** and simple, from kind heart.

4.2.4 Character Substitution

Substitution of characters with their unicode equivalents in order to exploit the weakness of a plagiarism detection approach is called technical disguise (Meuschke and Gipp, 2013). In particular, the substitution of an ASCII character with a lookalike character from a Latin, non-ASCII character set is called homoglyph substitution. Table 4.3 shows the correspondence between an ASCII letter and its unicode cyrillic equivalents (Gillam, Marinuzzi, and Ioannou, 2010).

TABLE 4.3: Letters 'a' and 'e' with their Cyrillic Equivalents

Ansi Character	Unicode Value	Unicode Equivalent	Unicode Value
a	92	a (Cyrillic)	U + 0430
e	97	e (Cyrillic)	U + 0435

Most word *n*-gram based approaches might fail to detect homoglyph substitution obfuscation since a unit of similarity in these approaches is a word. In the short stories corpus, we used a replacement of two of the most frequently occurring letters 'a' and 'e' with their Cyrillic equivalents. This replacement makes most words containing 'a' or 'e' incomparable with their ASCII counterparts, however words such as 'in', 'willing', 'country' are still recognizable. We opted for a lower degree of obfuscation by leaving some words intact in order to leave some space for detecting seeds. However, a higher degree of homoglyph substitution is also possible which might make the entire text incomparable, thereby rendering plagiarism detection techniques useless. This scheme is followed in the dataset by Palkovskii and Belov (2015), which was a part of PAN-2015. The following example shows our replacement of homoglyphs:

1. (Sentence 1) Now there lived in the country two brothers, sons of a poor man, who declared themselves willing to undertake the hazardous enterprise.
2. (Sentence 2) Now **there** lived in **the** country two brothers, sons of **a** poor m**a**n, who d**e**clared themsel**v**e**s** willing to und**e**rta**k**e th**e** h**a**z**a**rdous **e**nterpris**e**.

Construction of the Character Substitution Obfuscation Type

Given a source passage, we replaced the given characters ('a' and 'e') with their Cyrillic equivalents. In order for the a given plagiarism detection approach not to detect any other type of similarity, the suspicious document comprised only the converted passage, while the source document consisted of the original passage in addition to the text of the story. Because of this, the document sizes in the source and suspicious documents for this type of obfuscation are disproportionate, i.e., source documents are much longer than the suspicious documents.

4.3 Reviews of the Corpus

To ensure the quality of the submitted corpora, PAN-2015 also included a track for peer review of the submitted corpora, i.e. participants reviewed each others' submitted corpora. Furthermore, Potthast et al. (2015) also provided an indepth review of our corpus. The following are the salient features of these reviews.

1. **Domain-Specific Dataset**: Reviewers observed that the dataset represents the domain of short stories. While being from a single domain provides similarity of topic and consistency, it also includes archaic language which might not be in use today (Potthast et al., 2015). Nevertheless, there is a possibly strong topical relation between documents describing the same fairy tale.

2. **Provision of Context in the Embedding Document**: Since the methodology of corpus construction was to embed plagiarised passages into other, unrelated fairy tales, this method would blend in the plagiarised text with documents of a similar genre providing similarity of context. With this strategy topic drift analysis may only be partially successful in detecting plagiarism (Potthast et al., 2015).

3. **Usage of Manual Translations**: A salient feature of our corpus is the usage of manual translations of Grimms' fairy tales which is equivalent to utilising a pseudo-parallel corpus (Potthast et al., 2015). A pseudo-parallel corpus is a collection of aligned "*sentences in one language with their corresponding automatic translations*" (Zhang, Jiajun and Zong, Chengqing, 2016). In case of Grimms' fairy tales, a collection of aligned text fragments from various translations can

be considered as a pseudo-parallel corpus. This usage of independent translations is an interesting challenge for text alignment algorithms.

4. **UTF Character Substitution as a Novel Obfuscation Type**: Based on the obfuscation types present in previous years of PAN, the introduction of UTF character substitution is a new addition to the types of obfuscation. Potthast et al. (2015) remark that, UTF character substitution "*makes it more difficult, though not impossible, for text alignment algorithms to match words at a lexical level.*"

5. **Loss of Semantic Relatedness and Overall Opinion**: Franco-Salvador et al. (2015) remark that automatic replacement of synonyms in the synonym replacement obfuscation type might result in the loss of semantic relatedness in some cases, since the replaced text might not represent the same nuance as the original text. However, in their view, "*the overall opinion about this corpus is positive.*"

In general, the reviewers have appreciated several features within the corpus including the use of manual translations, selection of similar topics, and introduction of UTF character substitution. Furthermore, as shown in the next section, results of the plagiarism detection in the corpus using various approaches in the PAN repository also testify to the usefulness of this corpus towards highlighting the strengths and weaknesses of various plagiarism detection approaches.

4.4 Results and Discussion

In this section we provide the results of detecting plagiarism on various obfuscation types within the Short Stories Corpus using detection approaches available within the PAN-repository. The objective of these results is to present the suitability of using the short stories corpus as benchmark to assess the effectiveness of plagiarism detection approaches. These results also highlight the relative effectiveness of obfuscation strategies in hiding plagiarism by evading detection. For a further insight into the character substitution strategy, we also present the results for these approaches on the PAN-2015 dataset by Palkovskii and Belov, which included a high degree of character substitution as its only obfuscation strategy.

Figure 4.3 displays the plagdet scores, while Figure 4.4 shows the precision and recall scores for the 7 best performing approaches on each of the three different types of obfuscation (story retelling, synonym replacement and character substitution), as well as on the dataset by Palkovskii and Belov. In cases where more than one version of a successful approach was available, we considered the latest version only.

4.4.1 Story Retelling

For story retelling in Figure 4.3 (a), we observe that a number of PAN approaches successfully detect plagiarism for retold versions of the original passages. In particular, the approach by Oberreuter performs the best within these approaches with a plagdet score of 0.735. Other approaches by Kong and Sanchez-Perez (Sanchez-Perez, Sidorov, and Gelbukh, 2014) also perform well in detecting plagiarism with a plagdet score above 0.70. Overall, the average plagdet score of all of these approaches 0.663.

In terms of precision and recall, we observe from Figure 4.4 (a) that while the precision is high for almost all of the approaches, the recall is generally lower, however it is near 0.60 for the best ones of the approaches. From these results, we can infer that the best performing approaches within the PAN repository are generally successful in detecting plagiarism in text that has been paraphrased.

4.4.2 Synonym Replacement

For the case of synonym replacement obfuscation, Figure 4.3 (b) and Figure 4.4 (b) show that the plagdet scores, as well as precision and recall are much higher as compared to the scores for story retelling. Furthermore, we observe that a number of approaches detect plagiarism close to perfection with a plagdet score greater than 0.90. This is inline with the nature of these obfuscations since synonym replacement is confined to substitution of synonyms with equivalent words and expressions. Here also, the best performing approach is that of Oberreuter with a plagdet score of 0.966. The average plagdet score for these approaches was 0.883.

4.4.3 Character Substitution

For character substitution, we have included results from two corpora: one from our own corpus (Figure 4.3 (c) and the other from Figure 4.3 (d) by Palkovskii and Belov). Character substitution in the short stories corpus involved replacement of only two characters ‘a’ and ‘e’, which is a lower degree of obfuscation, while in the corpus by Palkovskii and Belov there was a much higher replacement of characters in words, with possibly every word being obfuscated by homoglyphs.

From the results in Figure 4.3 and Figure 4.4 we observe that most of the approaches scored a low plagdet score in our corpus with the only exceptions being the approaches by Glinos, Oberreuter and Palkovskii. In particular, the approach by Glinos involved a short seed length and sequence alignment expansion, which led to a good plagdet score in the approach by Glinos, since obfuscation by homoglyphs did not affect all of the words.

FIGURE 4.3: Plagdet Scores for (a) Story Retelling, (b) Synonym Replacement, and Character Substitution (c) (low) and (d) (high)

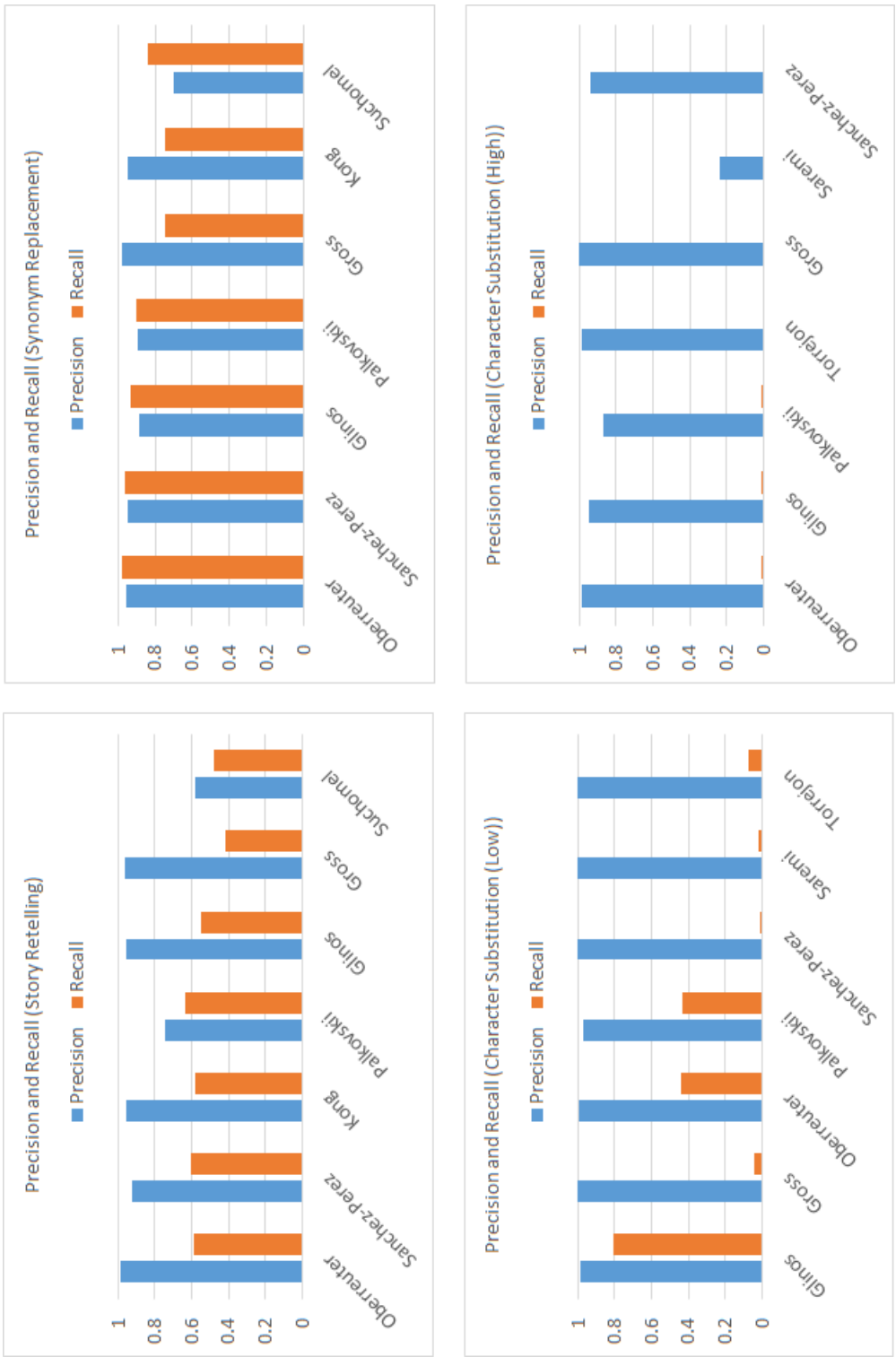


FIGURE 4.4: Precision and Recall Scores for (a) Story Retelling, (b) Synonym Replacement, and Character Substitution (c) (low) and (d) (high)

However, as the number of characters substituted is increased, to the point where every word in the corpus is obfuscated by homoglyphs, all of the approaches perform very poorly. This can be seen from Figure 4.3 (d) where all of the approaches score a nearly 0.0 plagdet score. These results can also be corroborated from Figure 4.3 and Figure 4.4 where it can be observed that the recall values are generally low in case of Figure 4.4 (low obfuscation) and almost 0.0 (high obfuscation).

From these results, it can be observed that out of all the obfuscation types, character substitution has shown to be a simple but highly effective technique for hiding plagiarism since all of the approaches are simply unable to detect *any* plagiarism. This point is not only true for PAN approaches, but it can be corroborated by a using a number of commercial plagiarism detectors, where it has been found that these detectors fail to detect any plagiarism if text is obfuscated by homoglyphs (Gillam, Marinuzzi, and Ioannou, 2010).

4.5 Conclusion

In this chapter, we proposed additional obfuscation types by creating a new data source for plagiarism detection which was tested against a large number of plagiarism detection approaches from the PAN repository. Our corpus included novel obfuscation types, with two of these being semantic and the other two being syntactic. The quality of the corpus was verified using the peer review process. From the results, we observed that PAN approaches generally performed well on the semantic types, i.e., story retelling and synonym replacement, suggesting that plagiarism detection approaches within the PAN repository can be successfully applied to paraphrase plagiarism. However, character (homoglyph) substitution proved to be a simple, yet very effective technique in hiding plagiarism as almost all of the approaches were completely unable to detect any plagiarism.

The main research contributions of this chapter are twofold which address the research questions presented in the introductory section of the chapter, i.e., (a) the creation of data source which includes novel obfuscation types simulating additional types of obfuscation, and (b) the finding that character substitution is an obfuscation type which effectively evades plagiarism detection as compared to the proposed obfuscation types.

In the next chapter we will explore character substitution further and propose two different approaches for detecting plagiarism in text that has been obfuscated with homoglyphs.

Chapter 5

Plagiarism Detection in Texts Obfuscated by Homoglyphs

In the last chapter it was shown that UTF character substitution (or homoglyph substitution) is an effective form of technical disguise used for obfuscating plagiarised text. This was evident by the tests of plagiarism detection approaches within the PAN repository on texts obfuscated by homoglyph substitution. Furthermore, this observation is also confirmed by tests of several plagiarism detection systems on homoglyph obfuscated texts (Weber-Wulff, 2014). This effectiveness of homoglyph substitution in evading plagiarism detection methods is due to the change in the computational representation of plagiarised text. Because of this change in computational representation, plagiarism detection methods that are based on matching individual characters (and words) are unable to correctly match identical letters (and words), resulting in a zero similarity score.

In this chapter we explore various aspects of the homoglyph substitution obfuscation by first examining the unicode list of confusable characters and the dataset by Palkovskii and Belov, 2015, which is based on homoglyph substitution. We then propose two different strategies to detect plagiarism in texts obfuscated by homoglyphs: (a) by substituting ASCII characters corresponding to homoglyphs from the unicode list of confusable characters, and (b) by approximate word matching using the normalized hamming distance between words. Experiments on the dataset by Palkovskii and Belov (2015) show that the list based approach performs better than using hamming distances. However, the hamming distance based approach might be useful in documents having a large number of foreign characters as source text.

The organization of this chapter is as follows: Section 5.1 provides an introduction to the notion of 'disguised plagiarism'. Section 5.2 examines the unicode list of confusable characters and the dataset in use. Section 5.3 provides details of the proposed approaches. Section 5.4 presents the results of the two approaches with a relative comparison of their performance. Finally Section 5.5 concludes this chapter and sets the direction for the next chapter.

5.1 Introduction

The notion of 'Disguised Plagiarism' refers to a class of methods used for intentionally hiding text that has been copied (Meuschke and Gipp, 2013). In particular, *technical disguise* refers to a form of disguised plagiarism, wherein obfuscation techniques are used in order to evade the detection of plagiarised text by changing the computational representation of text. This change may be in the form of a change of script or the use of graphical symbols in place of character representation of text.

An important method for technically disguising text is to substitute characters visually identical to other characters in some other script (Gillam, Marinuzzi, and Ioannou, 2010). A *homoglyph* is a character that is visually identical to another character in some other script. For example, the Latin character 'p' (Unicode U+160) and the Cyrillic 'p' (Unicode U+0440) have identical glyphs but distinct Unicode values, making the words 'paypal' and 'paypal' appear identical to a human evaluator, but undetectable to an automated plagiarism detection system that has not been designed to deal with such changes. Using the substitution of homoglyphs in lieu of the characters of copied text can result in plagiarism detection approaches being unable to compare identical words and phrases. In tests of several leading plagiarism detection systems most systems were unable to detect similarities between source and plagiarised texts obfuscated using homoglyphs (Kakkonen and Mozgovoy, 2010).

In addition to text obfuscation for plagiarism, homoglyphs have also been used in IDN (Internationalized Domain Name) homograph attacks used to direct users towards alternative websites. With such an attack, users could be directed towards the website 'paypal.com' which is a Cyrillic substituted version of the Latin 'paypal.com'. Some of the current approaches to deal with IDN homoglyph attacks include:

1. Coloring-based strategies that distinguish homoglyphs by assigning various colors to foreign script characters (Wenyin, Fu, and Deng, 2008);

2. Unicode character similarity list (UC-SimList) (Fu, Deng, and Wenyin, 2006) to detect homoglyphs in URLs; and

3. Punycode (Costello, 2003) which is designed to convert internationalized (Unicode) domain names into ASCII characters. Using Punycode, non-ASCII characters are converted into a combination of ASCII letters, digits and hyphens. For example, the string "Goo´gle" is converted to the ASCII 'xn–oole-ksbc', while 'façade' is represented in punycode as ASCII 'xn–faade-zra'. An important application of punycode is detection and avoidance of phishing attacks by directing users away from identical looking but suspicious websites.

Some of these approaches might not be useful for plagiarism detection e.g. Punycode results in loss of information, and coloring requires visual inspection. However, the idea of using a list of Unicode equivalents for detecting IDN homograph attacks can be utilized for plagiarism detection in homoglyph obfuscated texts.

The research questions that arise from the stated discussion are as follows: (a) What are some of the approaches that can be used to detect plagiarism when text has been obfuscated using homoglyph substitution, and (b) what are the advantages, disadvantages and use case scenarios for each of these proposed approaches.

In this chapter we present two alternate approaches for detecting plagiarism in text that has been obfuscated using homoglyphs:

1. by using the Unicode list of 'confusables' to find and replace homoglyphs with visually identical ASCII letters; and

2. by using a measure of similarity based on normalized hamming distance to match homoglyph obfuscated words with source words.

Our work shows while both approaches are successful in detecting plagiarism as compared to using no approach for homoglyph obfuscation at all, the list-based approach performs slightly better. In particular, the list-based approach scores higher in terms of plagdet score, while having less computational overhead.

5.2 Resources

In this section we list the resources used towards our approach for plagiarism detection in homoglyph obfuscated texts.

5.2.1 The Unicode List of Confusable Characters

A straightforward approach for plagiarism detection in homoglyph obfuscated texts is to use a list of homoglyph-alphabet pairs. Several lists of homoglyph-alphabet pairs are freely available (e.g., `homoglyphs.net`). These lists consist of a sequence of latin characters in a table or list format with possible look-alike characters in various scripts. For example, the list available at `homoglyphs.net` consists of a sequence of 93 latin characters with a maximum of 6 homoglyphs in various scripts for each character giving a total of 558 entries.

One of the most comprehensive lists of homoglyph-alphabet pairs is the Unicode list of *confusable* characters. This list has been released by the Unicode consortium, which is a list of visually similar character pairs that include homoglyphs and their corresponding Latin letters `http://unicode.org/reports/tr36/confusables.txt`. We use the current version (12.0.0) of the list of confusables which contains contains

6296 pairs of confusable characters. Figure 5.1 shows a partial list of letters similar to the letter 'p' taken from this list.

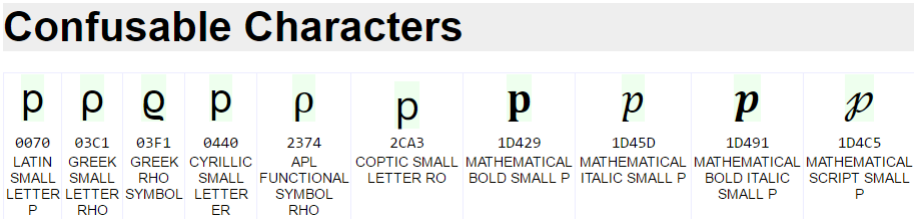


FIGURE 5.1: Visually Confusable characters for 'p' from the Unicode List of Confusables

5.2.2 Evaluation Dataset

We use the PAN-2015 evaluation lab dataset submission by Palkovskii and Belov (2015) which is based on the PAN-2013 training dataset with characters in the suspicious documents replaced with homoglyphs. This dataset consists of 5185 document pairs divided into five categories of 'no plagiarism', 'no obfuscation', 'translation', 'random' and 'summary' obfuscation. Results of plagiarism detection with this dataset in Section 4.5 showed a plagdet score of 0.0 for almost all of the plagiarism detection approaches in the PAN-repository.

Empirically, we have found that this dataset replaces all occurrences of 'A', 'a', 'E', 'e', 'I', 'i', 'O', 'o', with their Cyrillic equivalents. Due to this substitution almost all of the words in the suspicious documents are not comparable to the words in the source documents. For example, consider the following two sentences from the corpus. Except for the word 'just', none of the words in the two sentences match with each other.

1. Inside the classroom, you obtain assignments, write papers, conduct research, take tests, and exchange information just like in a traditional class.

2. **I**ns**i**d**e** th**e** cl**a**ssr**oo**m, y**o**u **o**bt**a**in **a**ssignm**e**nts, wr**i**t**e** p**a**p**e**rs, c**o**nduct r**e**s**ea**rch, t**a**k**e** t**e**sts, **a**nd **e**xch**a**ng**e** **i**nf**o**rm**a**ti**o**n just l**i**k**e** **i**n **a** tr**a**diti**o**n**a**l cl**a**ss.

Figure 5.2 shows source and plagiarised text using the Corpus Viewer software. It can be observed that the highlighted text in the two documents appears the same to a human eye, although the text in the right window has been replaced with homoglyphs.

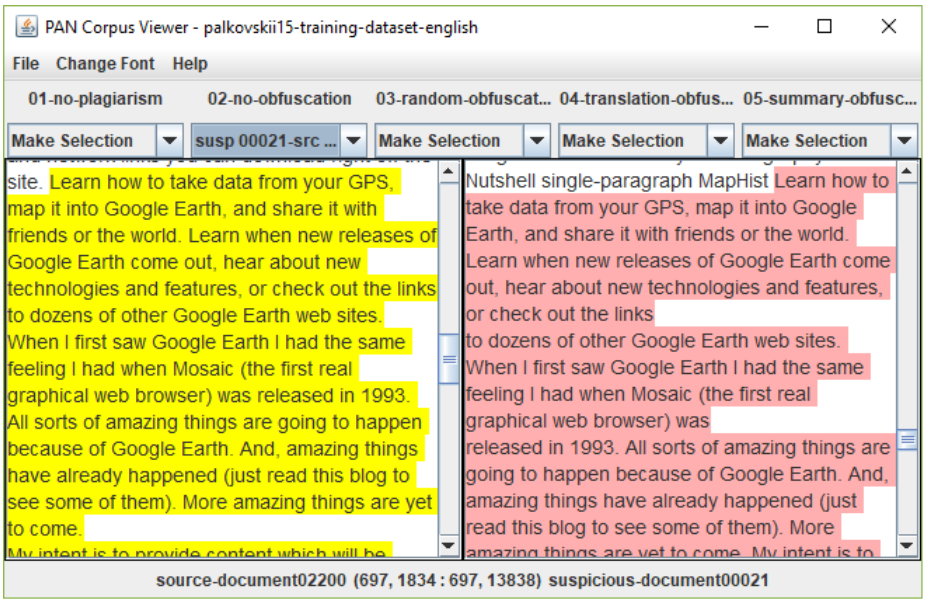


FIGURE 5.2: View of Source and Plagiarised Text Obfuscated using Homoglyphs

5.3 Approaches

In this section we describe our proposed approaches used for detecting plagiarism in homoglyph obfuscated texts.

5.3.1 Using the List of Unicode Confusables

In our first approach (shown in Figure 5.3), we find and replace every non-ASCII character in the suspicious documents with the corresponding visually matching character from the list of confusable characters. This process replaces homoglyphs in the text of the suspicious documents with visually similar ASCII characters. The resulting suspicious documents can then be compared with the source documents for similarity.

This approach can be appended to any existing plagiarism detection technique as a preprocessing module. For the choice of plagiarism detection technique, we use our PAN-2014 hashing based approach as described in Section 3.3.3. However, we use word trigram similarity as the seeding strategy instead of using character n-grams. This is achieved by an exact matching of word n-grams of size 3 between the source and suspicious documents. The rest of the merging and filtering phases remain the same as described in Sections 3.3.4 and 3.3.5. This change of seeding is done in order for this approach to be comparable with our second approach.

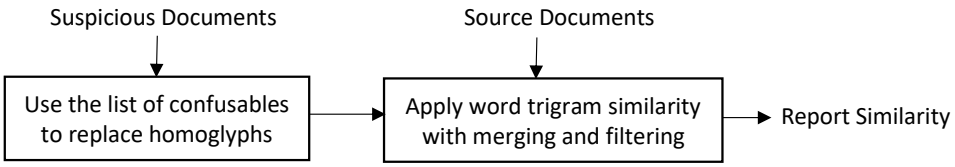


FIGURE 5.3: Block Diagram for Plagiarism Detection using the List of Confusables

5.3.2 Normalized Hamming Distance between words

Hamming Distance (when applied to strings of characters) detects the number of substitutions (replacements) from one string to another by finding the number of positions where the two strings differ (Navarro, 2001). We use the normalized version of hamming distance, where hamming distance is divided by the length of the larger string. This is denoted as a similarity score (sim_h) computed using normalized hamming distance, defined between two words w_1, w_2 of equal length as:

$$sim_h(w_1, w_2) = 1 - \frac{\texttt{Number of substitutions}(w_1, w_2)}{\texttt{length}(w_1)} \tag{5.1}$$

Compared to other approximate string similarity measures (such as word edit distance), normalized hamming distance has the advantage of significantly reducing the number of false positives generated. This is because hamming distance is undefined for strings of unequal length, (we consider $sim_h = 0$ in this case), whereas these strings might be marked as similar using alternative string similarity techniques, such as character skip gram matching.

For example, sim_h**(play, plаys) = 0, while** sim_h**(play, plаy) = 0.75**.

Figure 5.4 shows a matrix with normalized hamming distance similarity values for part of the sentence in the Section 5.3.1. Here, the Cyrillic letters (homoglyphs) are represented as underlined. It can be observed that most of the words not lying in the vicinity of the main diagonal have a sim_h value of 0, however the word 'a' (on the main diagonal) also has a sim_h value of 0. For this example, a sim_h value ≥ 0.50 would suffice for matching words.

In our second approach, normalized hamming distance similarity (sim_h) is used to compare each word in the suspicious document with the words in the source document. If a pair of words have a value of sim_h greater than or equal to a particular threshold, we consider them as similar. The threshold value depends on the extent of homoglyph substitution in the dataset. For example, if most of the letters in each word have been replaced by homoglyphs, then a lower threshold value will be required to match these words.

	just	like	in	a	traditional	class
just	1.00	0.00	0.00	0.00	0.00	0.00
like	0.00	0.50	0.00	0.00	0.00	0.00
in	0.00	0.00	0.50	0.00	0.00	0.00
a	0.00	0.00	0.00	0.00	0.00	0.00
traditional	0.00	0.00	0.00	0.00	0.55	0.00
class	0.00	0.00	0.00	0.00	0.00	0.80

FIGURE 5.4: Homoglyph Matrix with sim_h values

Using this procedure for approximate matching of words instead of exact matching, we apply word trigram similarity with merging and filtering (as used in the list-based approach) to find the plagdet score between the source and suspicious documents. We conduct our experiments on the PAN-2015 dataset (Palkovskii and Belov, 2015) used in the list-based approach. Regarding the threshold value of sim_h for matching words in our experiments, we do not pre-select a value for this threshold. Instead we calculate plagdet scores for the entire dataset for a range of values of sim_h as shown in Figure 5.4.

5.4 Results and Discussion

In this section we provide the results of the two approaches followed by a discussion on the comparative advantages and disadvantages of the two approaches.

5.4.1 Substitution using the Unicode List of Confusables

Table 5.1 shows the results of plagiarism detection in terms of Precision, Recall and Plagdet scores. In this Table, granularity scores are not shown, but used for the computation of plagdet scores. It can be seen that except for summary obfuscation, plagdet scores for all other categories including that for the entire dataset are moderately high ($\geq$ 0.60). This can be compared with the performance of most of the PAN approaches from the PAN repository in Chapter 4 where where the reported plagdet scores were mostly 0.0. This leads us to conclude that the addition of a module for dealing with homoglyph substitution is a small, but significant step for detecting plagiarism in homoglyph obfuscated texts.

Observations

During the homoglyph replacement phase using the list-based approach, we observed that a number of replacements were also made for non-Latin characters in

TABLE 5.1: Plagdet Scores by Replacing Homoglyphs

Obfuscation Type	Precison	Recall	Plagdet
Entire Dataset	0.772	0.727	**0.670**
No Obfuscation	0.663	0.988	**0.717**
Random Obfuscation	0.953	0.667	**0.707**
Translation Obfuscation	0.781	0.643	**0.632**
Summary Obfuscation	0.826	0.107	**0.142**

suspicious documents which were not intended as homoglyphs in source documents. For example, the currency symbol '¢' was replaced by a 'c', as these are also considered as confusable characters. However, in case of the present dataset the number of such replacements was small enough not to cause any significant change to the overall result.

This observation suggests that the approach of using a list of homoglyph-alphabet pairs to replace characters may not work well when the source text contains a large number of foreign characters, since these might be converted to ASCII characters in the substitution phase. This problem can be addressed by searching through the source and suspicious documents to distinguish homoglyphs from true source non-Latin characters, at the cost of increased computation time. Another possible solution would be to map all non-ASCII characters to ASCII characters in both the source and suspicious documents before carrying out plagiarism detection.

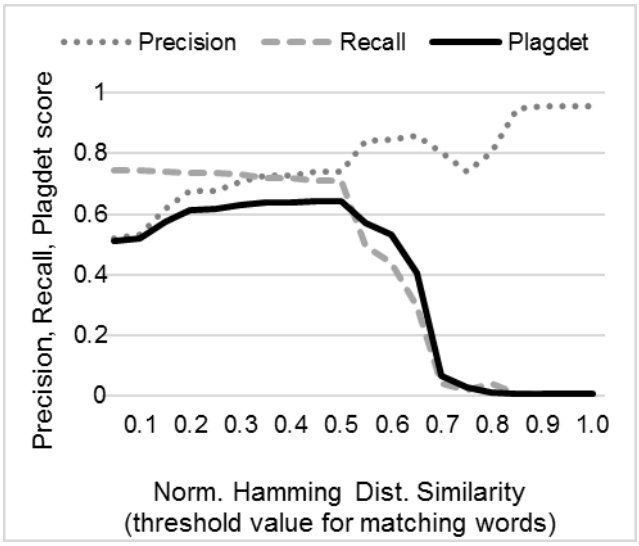


FIGURE 5.5: Precision, Recall and Plagdet Scores for the complete dataset using Normalized Hamming Distance Similarity

5.4.2 Normalized Hamming Distance

Figure 5.5 shows precision, recall and plagdet scores for various values of sim_h using normalized hamming distance similarity. It can be seen that a threshold value of $sim_h \approx 0.45$ is giving the highest plagdet score of 0.644. The plagdet score rapidly decreases after $sim_h = 0.5$ since higher threshold values increase the number of true matches being rejected. Table 5.2 gives plagdet scores for each category of plagiarism in the dataset for a threshold value of 0.45. Similar to Table 5.1, we observe that except for summary obfuscation, most of these values are moderately high (≥ 0.6). These scores suggest a better performance for the list-based approach.

Figure 5.6 gives the precision, recall and plagdet scores for each obfuscation type vs. the threshold value. Here, we observe that the ideal value of the threshold for each obfuscation type is ≈ 0.45 except for no obfuscation which is ≈ 0.60. The primary reason for this change is an increase in precision after $sim_h = 0.50$ for the no obfuscation type. This leads us to the observation that the ideal threshold value using normalized hamming distance might be different for each obfuscation type.

TABLE 5.2: Plagdet Scores using the Normalized Hamming Distance for $sim_h = 0.45$

Obfuscation Type	Precison	Recall	Plagdet
Entire Dataset	0.739	0.711	**0.644**
No Obfuscation	0.604	0.989	**0.662**
Random Obfuscation	0.943	0.636	**0.688**
Translation Obfuscation	0.790	0.627	**0.626**
Summary Obfuscation	0.846	0.142	**0.142**

Comparison of the two approaches

On comparing the two approaches, i.e., the list-based approach and the normalized hamming distance based approach, we find the list-based approach producing slightly higher plagiarism detection scores in all the obfuscation types in Table 5.1 as compared to Table 5.2. This is primarily due to (a) normalized hamming distance being unable to match some words (for example consider the pair (a, `а`) in Figure 5.4, and (b) possible mismatches of words due to approximate string matching. In contrast, for the list-based approach, there is no requirement of a threshold value or computational overhead for computing sim_h. In this sense, we consider substitution using the unicode list of confusables to be a better approach for the given dataset.

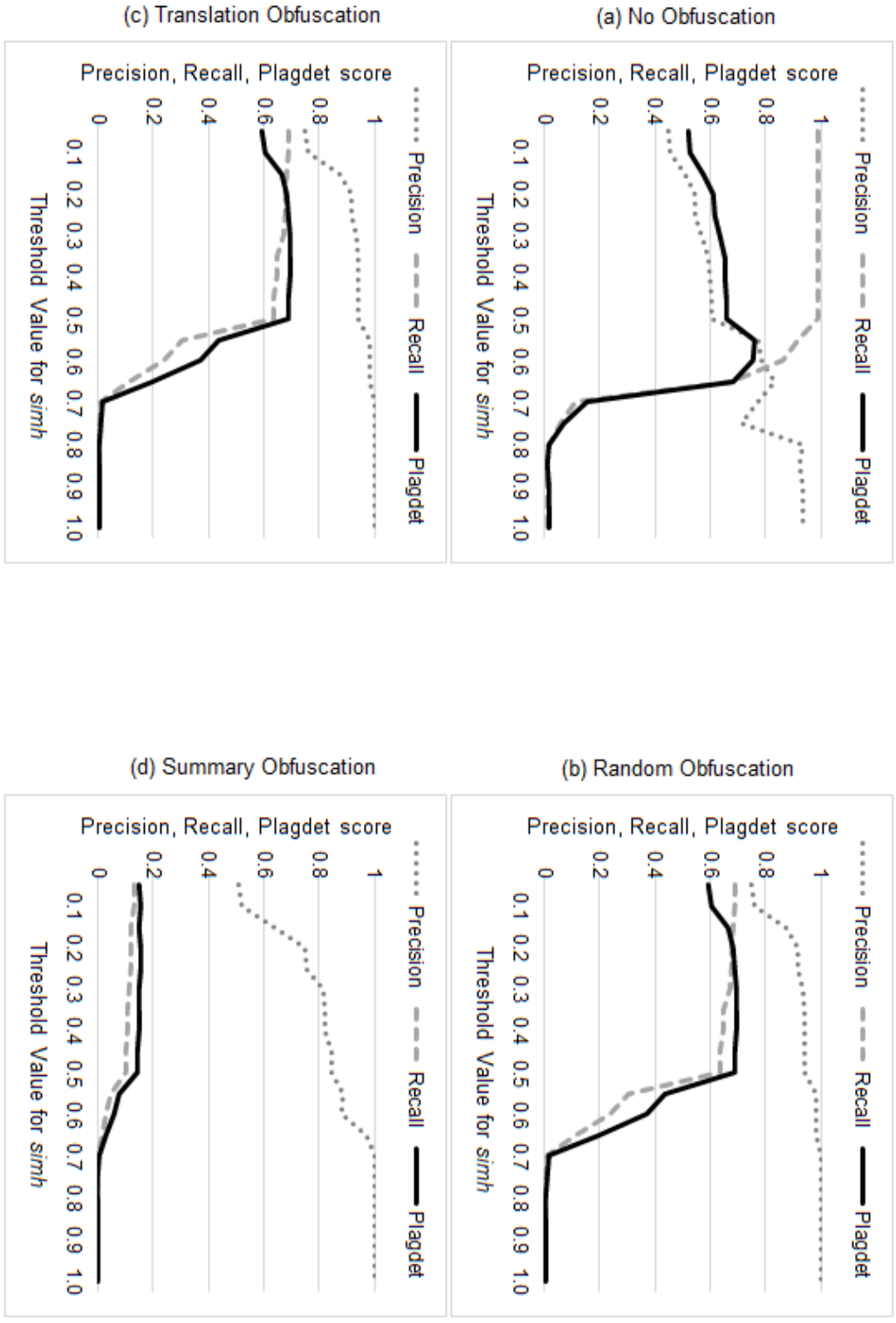


FIGURE 5.6: Precision, Recall and Plagdet Scores for each obfuscation type for various values of sim_h

On the other side, the normalized hamming distance based approach can be useful when the source documents contain a large number of foreign (non-ASCII) characters, replaced by homoglyphs which are also foreign characters. Under this situation mapping all characters in both the source and suspicious documents to ASCII characters using the list-based approach might result in loss of information leading to lower plagdet scores.

5.5 Conclusions and Future Work

The development of techniques for automated plagiarism detection continues to be an active area of research. In this chapter we presented two approaches for plagiarism detection in homoglyph obfuscated texts: the first approach utilizes the Unicode list of confusables to replace homoglyphs with visually identical letters; the second approach uses a similarity score computed using normalized hamming distance. Performance of both approaches showed better scores for the list-based approach with no requirements for threshold values. However, the list-based approach has the drawback of considering true non-ASCII characters as homoglyphs which might reduce its effectiveness in the presence of a large number of foreign (non-ASCII) characters.

A possible variation of character substitution can be considered by inserting characters within text that are not visible in the printable version. Heather (2010) considers a variety of possibilities for generating a pdf version of such a document. In addition, insertion of non-printing characters in obfuscated text can be considered as another example of hiding plagiarised text using character substitution. Both these types of obfuscation can be addressed at the tokenization stage by the detection and removal of such characters, as these characters generally do not represent copied text.

In the next chapter we shift our focus from plagiarism detection to identifying paraphrase types in text that has already been detected as plagiarised. This identification of paraphrase types is intended to assist a human evaluator in deciding whether a given pair of texts can be considered as plagiarised or not. In this sense, paraphrase type identification marks the next phase in plagiarism detection research.

Chapter 6

Same Polarity Substitution Detection

With the advancement of research in plagiarism detection, researchers have identified several forms of plagiarism as outlined in Chapter 2 (Section 2.2). Paraphrase plagiarism is one such form of plagiarism wherein paraphrasing is used by applying various text editing operations to obfuscate plagiarised text. These operations include, (but are not limited to) synonym substitution, insertion and deletion of text and word reordering. In the context of plagiarism, these textual transformations have been identified as paraphrase types, grouped together as paraphrase typologies as outlined in Section 2.6.

Current and past research on plagiarism detection has been focused on the reporting of plagdet (similarity) scores and the alignment of textual matches between source and suspicious documents (Section 2.4). This trend can be also be observed in commercial plagiarism detection systems whereby a similarity score is reported along with textual matches between source and suspicious documents. Furthermore, based on these similarity reports, the decision regarding the occurrence of plagiarism is left to a human evaluator since reported matches might not correspond to actual instances of plagiarism.

Barrón-Cedeño et al. (2013) have proposed the identification of paraphrase types in plagiarised text as the next direction of research in plagiarism detection. They have also proposed a paraphrase typology comprising of 20 paraphrase types, with same polarity substitution being identified as the most frequent paraphrase type in the P4P (Paraphrase for Plagiarism) corpus. Research on methods identifying paraphrase types in plagiarised text has the potential to provide additional valuable information to a human evaluator in making an informed decision about the actual occurrence of plagiarism.

In the context of identifying paraphrase types in paraphrased, plagiarised text, the following research questions arise: (a) what are some of the methods that can be used to identify same polarity substitutions in paraphrased, plagiarised text, and

(b) how do these methods perform in comparison to existing measures used for detecting paraphrase plagiarism.

In this chapter we develop methods to identify the most common paraphrase type used in plagiarism, i.e., same polarity (synonym) substitution. Our methods are based on the matching of contexts and applying word embedding similarity scores for the detection of same polarity substitutions. We use a three staged approach which involves (a) preprocessing, (b) detection of (i) contextual and (ii) non-contextual same polarity substitutions and (c) filtering. We use the Smith Waterman Algorithm (Smith and Waterman, 1981) for detecting contextual same polarity substitutions and ConceptNet Numberbatch pretrained word embeddings (Speer, Chin, and Havasi, 2017) for detecting non-contextual same polarity substitutions. Since the problem of detecting same polarity substitutions has not been addressed in plagiarism detection research, we compare our methods with existing similarity measures used for paraphrase plagiarism detection, showing improved results.

The organization of this chapter is as follows: In Section 6.1 we provide an introduction to the problem of detecting same polarity substitutions. Section 6.2 describes the datasets in use and provides details of the measurement parameters. Section 6.3 elucidates the proposed approach with description of each phase in detail. Section 6.4 provides results of our approaches and a comparison of our approach with existing measures used for paraphrase plagiarism detection. Finally Section 6.5 concludes this chapter with a summary of our approach and directions for further research.

6.1 Introduction

As stated in Section 1.1, plagiarism detection systems have been developed in order to detect and prevent plagiarism. These plagiarism detection systems calculate a similarity score, and provide a report displaying matched parts of text between the source and target documents. However, the final decision on whether a reported similarity score represents a genuine case of plagiarism rests with a human evaluator (McKeever, 2006). This is because, "*it requires a careful consideration of these annotated matches by a person to determine which, if any, constitute plagiarism*" (Mphahlele and McKenna, 2019, p. 1084).

Paraphrase plagiarism (Barrón-Cedeño et al., 2013; Carmona et al., 2018) can be defined as obfuscation of copied text using lexical and semantic transformations, such as synonym substitution, word reordering and rephrasing. These modifications change the surface form of a text, but preserve its overall meaning, thereby making it difficult for computers (and humans) to identify and prove the occurrence of plagiarism.

As discussed in Section 2.6.4, Barrón-Cedeño et al. (2013) have identified a number of different paraphrase types used in the context of plagiarism. These paraphrase types refer to text rewrite operations that a plagiarist might undertake in order to obfuscate copied text. In a study based on simulated cases of plagiarism, Barrón-Cedeño et al. (2013) have found that 'same polarity substitution', i.e., 'the substitution of synonymous words and phrases' forms the largest proportion of paraphrase types in plagiarised text. These findings have been corroborated in Bhagat and Hovy (2013) and Sun and Yang (2015), who have also reported that synonym substitution forms a large proportion of rewrite operations in paraphrased texts.

In this chapter we develop methods to identify same polarity substitutions in paraphrased, plagiarised text. The primary motivation for this research is to develop methods that may assist a human evaluator in making an informed decision regarding the occurrence of plagiarism, by highlighting the substituted text from within the reported plagiarised text. Our proposed methodology uses a three stage approach which involves: (a) preprocessing, (b) detection of same polarity substitutions through (i) matching word contexts, and (ii) using word embedding similarity measures, and (c) filtering. We use a variant of the Smith Waterman Algorithm for plagiarism detection to find near identical context matches that serve as anchor points for detecting same polarity substitutions. Secondly, we use word cosine similarity scores from ConceptNet Numberbatch pretrained word embeddings (Speer, Chin, and Havasi, 2017) to detect non-contextual cases of same polarity substitutions. Results from these two phases are merged and duplicates and other paraphrase types are removed during the filtering phase. The F_1 scores from our combined approach are 0.584 for P4P and 0.565 for the MSRP-A corpus, which are much higher than methods using either contexts or similarity measures including WordNet based similarity measures used in current works for paraphrase plagiarism detection.

From a wider perspective of plagiarism detection, a proposed solution to identifying paraphrase types can be considered as a possible extension to the textual alignment module of an extrinsic plagiarism detection system (Section 2.3). This is because after the textual alignment phase of an extrinsic plagiarism detection system, when sections of suspicious documents have been aligned with the source document, the process of paraphrase type identification can be initiated. This process is explained in Figure 6.1 where the output of an extrinsic plagiarism detection system as suspicious sections is presented as input with the source document to a paraphrase type identification module. The final output of this process are segments of paraphrase types identified within suspicious sections. In this sense methods detecting paraphrase types can enhance an extrinsic plagiarism detection system by providing an added functionality.

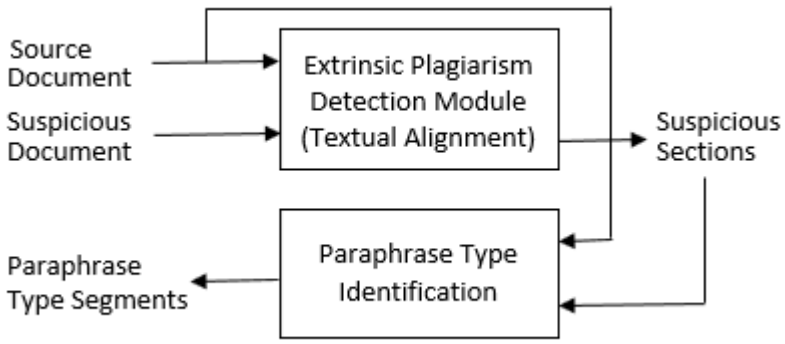


FIGURE 6.1: Block Diagram showing Paraphrase Type Identification added to Extrinsic Plagiarism Detection (Textual Alignment) Module

6.1.1 Problem Description

In the research problem under consideration, our goal is the detection of same polarity substitutions from a pair of text snippets. Each pair of text snippets comprises of 1–3 sentences and consists of a source and a plagiarised text snippet. Same Polarity Substitutions are defined as textual substitutions having nearly the same meaning and primarily consist of "*synonymy, general/specific substitutions, or exact/approximate alternations*" (Barrón-Cedeño et al., 2013). Let us consider the following partial snippet pair (17312–13) from the P4P corpus:

1. The wording of this resolution is very suggestive and it shouldn't be criticized very much.

2. The language of this resolution is gravely suggestive, and cannot be too closely criticised.

In the partial snippet pair above, also shown in Figure 6.2, we identify annotated same polarity substitutions as well as other paraphrase types in a pair of source and paraphrased snippets. It can be observed that in the given snippet pair there are three same polarity substitutions, while two are other paraphrase types. Furthermore, there are three single word substitutions and two phrasal substitutions.

6.2 Datasets and Measurement Parameters

In this section we provide a description of the datasets in use along with details of relevant measurement parameters.

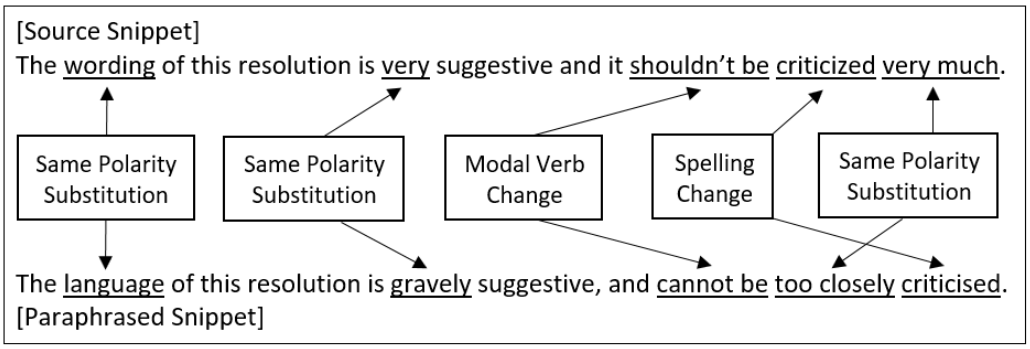


FIGURE 6.2: Paraphrase Types in Source and Paraphrased Snippets (P4P Corpus, 17312–313)

6.2.1 Datasets Used

We use the P4P (Paraphrase for Plagiarism) corpus (Barrón-Cedeño et al., 2013) as the primary source of data for identifying same polarity substitutions. The P4P corpus is a collection of simulated plagiarism cases taken from the PAN-2010 corpus (Potthast et al., 2009). These pairs form 6% of the manually simulated cases of plagiarism generated through Amazon Mechanical Turk by human participants, and therefore form the most challenging part of the PAN-2010 corpus. An analysis of various approaches for the PAN-2010 plagiarism detection approaches on several clusters of the P4P corpus demonstrated that F_1 scores from even the best performing approaches did not exceed 0.20 (Barrón-Cedeño et al., 2013), with several approaches not producing any F_1 scores at all.

Apart from the P4P corpus, we also use the MSRP-A corpus as a supplementary source of data for the identification of paraphrase types. The MSRP-A corpus is derived from the Microsoft Research Paraphrase Corpus (MSRP) (Dolan, Quirk, and Brockett, 2004) and annotated for the occurrence of various paraphrase types. Although the MSRP-A corpus is not based on cases of simulated paraphrase plagiarism, it does represent paraphrasing of sentences based on news sources and therefore we use it in order to complement the range of our results on the P4P corpus.

Non-contiguous Substitutions

The P4P corpus consists of a total of 5071 same polarity substitutions[1] out of which 145 are *non-contiguous*. The MSRP-A corpus consists of a total of 5485 same polarity substitutions out of which 142 are *non-contiguous*. Here, a non-contiguous substitution means that a single contiguous set of words in the source snippet may be

[1]Updated README file of the English annotated version, downloaded from http://clic.ub.edu/corpus/en/paraphrases-download-en [Last Accessed 13-July-2020]

mapped to a non-contiguous set of words in the target snippet or vice versa as a same polarity substitution.

In the partial snippet pair shown below from the P4P corpus, we provide an example of a non-contiguous substitution. It can be observed that the word "*desirable*" in the source snippet is mapped to two words "*Dear...dearer*" in the target snippet:

1. Peace is much *desirable* than war...
2. *Dear* as war may be, a dishonorable peace will prove much *dearer*...

Non-contiguous same polarity substitutions result in a 1-to-n or n-to-1 mappings with gaps in between. Because of possibly large gaps in the substitution it is difficult to identify the context from the annotation. Therefore, we have removed these non-contiguous substitutions from the datasets. The resulting corpora have 4926 same polarity substitutions for the P4P corpus and 5343 same polarity substitutions for the MSRP-A corpus, respectively.

6.2.2 Measurement Parameters

We use information retrieval measures of precision, recall and F_1 score to gauge the effectiveness of our approaches. The methodology to measure precision and recall for paraphrase type identification is similar to the one used in (Barrón-Cedeño et al., 2013), which in turn is based on measuring plagdet scores derived from PAN plagiarism detection series.

Using this methodology, we consider each instance of a same polarity substitution annotation as a four-tuple $s = (s_{start}, s_{size}, t_{start}, t_{size})$, where

- s_{start} = starting index of the same polarity substitution in the source snippet,
- s_{size} = length of the same polarity substitution in the source snippet,
- t_{start} = starting index of the same polarity substitution in the target snippet,
- t_{size} = length of the same polarity substitution in the target snippet,

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

A *match* between an annotation $s = (s_{start}, s_{size}, t_{start}, t_{size})$ and a detection $r = (s'_{start}, s'_{size}, t'_{start}, t'_{size})$ is the number of overlapping positions of characters represented as $s \cap r$. For the entire dataset, we calculate precision by dividing the size of each match by the size of corresponding detection; for recall this quantity is divided by the size of the corresponding annotation. These individual quantities are then summed up and further normalized by dividing by the number of instances for the respective measures (number of detections for precision, number of annotations for recall). This gives us the macro-averaged precision and recall.

Mathematically, precision and recall are given as:

$$precision = \frac{1}{|R|} \sum_{r \epsilon R} \frac{|\bigcup_{r \epsilon R}(s \cap r)|}{|r|}, \tag{6.1}$$

$$recall = \frac{1}{|S|} \sum_{s \epsilon S} \frac{|\bigcup_{s \epsilon S}(s \cap r)|}{|s|} \tag{6.2}$$

The F_1 *score*, which is the harmonic mean of both precision and recall is used for finding the combined effect of precision and recall as follows:

$$F_1 \; score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{6.3}$$

The above computed F_1 score is ***macro***-averaged where each same polarity substitution annotation is given an equal weight irrespective of its size in terms of the number of characters. In contrast, ***micro***-averaged precision, recall and F_1 score can be found by scaling each annotation by its size. This can be achieved by considering the size of each annotation and detection in the average calculation process. A micro-averaged F_1 score is influenced heavily by the size of the annotations and the detections since larger sized annotations (and corresponding detections) have a higher weight in the F_1 score. In contrast in a macro-averaged F_1 score, each annotation contributes equally regardless of its size. Throughout the rest of the work, precision, recall and F_1 scores refer to macro averages, unless specified otherwise.

6.3 Proposed Approach

We use a three-stage approach for the detection of same polarity substitutions which involves preprocessing, detection of same polarity substitutions and filtering. Each of these phases is briefly described below.

1. *Preprocessing*: This involves case-folding, punctuation removal, identifying abbreviations, sentence segmentation and matching.

2. *Detection of same polarity Substitutions*: This phase involves detecting same polarity substitutions using contextual matches and pretrained word embedding similarity measures.

3. *Filtering*: This is performed to filter out other paraphrase types that may have been detected as same polarity substitutions, such as spelling changes and modal verb changes.

Figure 6.3 shows the overall block diagram for our three-phase approach with a workflow through each of the three phases shown vertically.

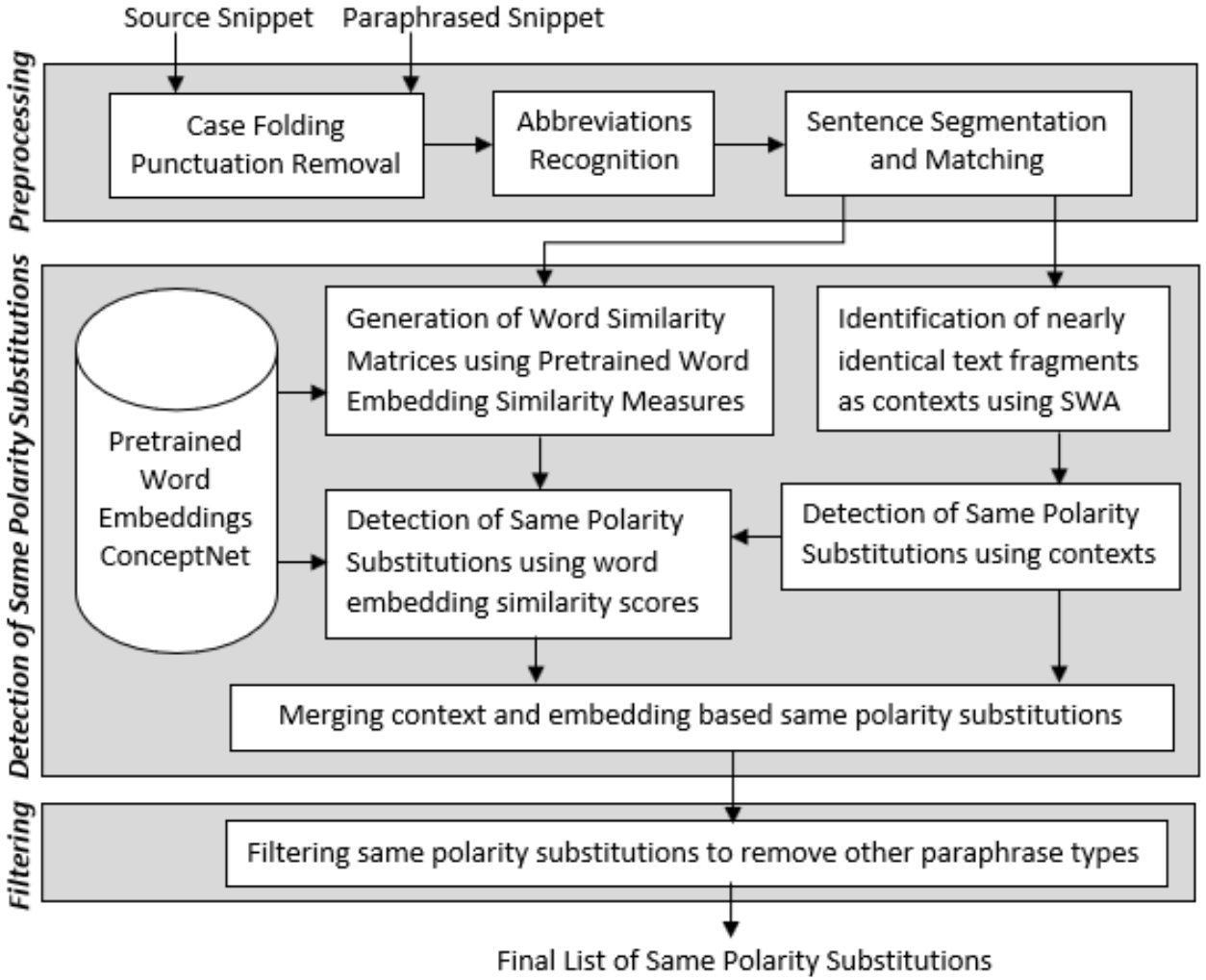


FIGURE 6.3: Block Diagram of the Three Stage System to detect Same Polarity Substitutions

6.3.1 Preprocessing

Preprocessing is primarily carried out to achieve two tasks:

1. removal of punctuation markers and non-alphabet characters, and
2. identifying and matching corresponding sentences in the source and paraphrased snippets.

In particular, the objective of the second step in preprocessing is to split snippets consisting of multiple sentences into their constituent sentences. This is followed by mapping the split sentences of the source snippet to those of the paraphrased snippet in order to reduce the pair size.

Preprocessing is initiated by case folding and replacing all punctuation and non-alphabetic characters with a space, except for the period or dot ('.'). Since the dot ('.') serves both as an end-of-sentence marker and abbreviation token, sentence splitting is achieved by first identifying abbreviations. This is done by using regular expressions for short abbreviations e.g. (Mr., U. K.), and using a lookup table for longer abbreviations (e.g. "Messrs.").

Next, snippets are split into sentences using the end-of-sentence marker ('.'). Matching of split sentences is done for only those snippet pairs that have an equal number of split sentences in order to achieve a one-to-one mapping. We use sentence cosine similarity for matching sentences by considering each sentence as a vector of words and create a matrix of sentence similarity values. More formally, let S_i be the i^{th} split sentence in the source snippet and T_j be the j^{th} split sentence in the paraphrased snippet. Then the entries of the sentence similarity matrix are given by:

$$M[i,j] = sim(\vec{S_i}, \vec{T_j}) = \frac{\vec{Si} \cdot \vec{Tj}}{|\vec{Si}| \cdot |\vec{Tj}|} \tag{6.4}$$

where $\vec{Si}$ and $\vec{Tj}$ represent the vector form of the sentences S_i and T_j in terms of words, respectively.

With this matrix constructed, the task of finding a 1-1 mapping between sentences is similar to the problem of finding a maximal matching in edge weighted bipartite graphs, also known as the linear assignment problem (Burkard and Cela, 1999). Bipartite graph matching has been used for matching words, such as in (Bhagwani, Satapathy, and Karnick, 2012). We use maximal weight bipartite graph matching for sentences as shown in Figure 6.4.

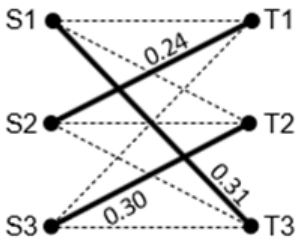
Figure 6.4 (a) shows constituent sentences of the snippet pair 16488-89 from the P4P corpus while Table 6.4 (b) displays a matrix of word cosine similarity values. It can be seen that sentence S1 matches T3, S2 matches T1 and S3 matches T2, which is also shown by the bipartite graph in Figure 6.4 (c).

Sentences in Snippet 16488	Sentences in Snippet 16489
(S1) All art is imitation of nature. (S2) One does not need to recognize a tangible object to be moved by its artistic representation. (S3) Here by virtue of humanity's vestures, lies its appeal.	(T1) In order to move us, it needs no reference to any recognised original. (T2) It is there in virtue of the vesture of humanity in which it is clothed, and makes its appeal at once and directly. (T3) It is usual to speak of all the fine arts as imitative arts.

(a) Constituent Sentences

sim	T1	T2	T3
S1	0.00	0.29	**0.31**
S2	**0.24**	0.04	0.12
S3	0.00	**0.30**	0.08

(b) Matrix of Pairwise Sentence Similarity



(c) Maximal Matching between Sentences

FIGURE 6.4: (a) Constituent Sentence Pairs in Snippets 16488-89, (b) Matrix of Pairwise Sentence Similarity Values, (c) Bipartite Graph showing Maximal Matching

Using the above procedure we successfully split and matched 305 snippet pairs in the P4P corpus and 153 snippet pairs in the MSRP-A corpus. These figures do not include single sentence snippets. It has to be kept in consideration that most of the snippets in the MSRP-A corpus are single sentence snippets as opposed to the P4P corpus, which is reflected in the lower number of split sentence pairs shown for the MSRP-A corpus.

6.3.2 Detection of same polarity Substitutions

After preprocessing, the detection of same polarity substitutions proceeds in two parallel ways, using: (1) matching of contexts, and (b) word embedding similarity scores.

Context-Based Same-Polarity Substitutions:

For the context-based method, our criterion of considering two text fragments as same polarity substitutions is based on the distributional hypothesis, which states that *"words are similar if their contexts are similar"* (Freitag et al., 2005).

We use the Smith Waterman Algorithm (Smith and Waterman, 1981) to detect identical text fragments between the source and paraphrased snippets. However, our definition of 'identical' is extended to 'nearly-identical' fragments by considering articles ('a' ≈ 'an'), pronoun forms with/without apostrophe (he ≈ he's) as well as simple verb forms (e.g. 'sit' ≈ 'sits'), as nearly identical words. This addition of nearly identical word forms increases the number of contexts, and consequently the correctly detected substitutions. Text markers indicating the beginning and end of the snippets are also inserted as additional contexts.

The Smith Waterman Algorithm can be customized to assign scores for matches and gaps depending on the specific requirements of the application. Our objective here is to detect same polarity substitutions which we consider as regions of unmatched text bounded by matched identical text fragments. Therefore, we construct the scoring scheme for the Smith Waterman Algorithm such that the cost of a match is much higher than the cost of a mismatch or gap penalty. In particular, for the scoring equation below, we use the following parameters: for a match, `sim(a, b) = 100 (match)`, `-1 (mismatch)` and for the gap penalty, `gap = -1`, stated in the following equation:

$$M[i,j] = max \begin{cases} M[i-1,j-1] + \texttt{sim(a, b)}, \\ M[i,j-1] + \texttt{gap}, \\ M[i-1,j] + \texttt{gap}, \\ 0, \quad otherwise. \end{cases} \tag{6.5}$$

The rationale of selecting an arbitrarily high score for a match (*sim* = 100 as compared to a *mismatch* and *gap* = -1) is to ensure that the calculated entries of the scoring matrix are all positive. This ensures that any traceback of the scoring matrix does not stall at a value of zero if it encounters a large same polarity substitution, but it allows alignment for the entire length of the snippet pairs.

We begin by constructing a scoring matrix of size $(m + 1) \times (n + 1)$, where m and n are sizes of the source and the paraphrased snippets including the beginning and end markers respectively. Figure 6.5 shows the scoring matrix for the partial snippet pair 17732-33 from the P4P corpus. The following same polarity substitutions can be deduced from this matrix:

1. 'Even with' ↔ 'Under',
2. 'instruction' ↔ 'teaching',
3. 'spent little time' ↔ 'was not long in'.

	Paraphrased Snippet	BEGIN	Under	such	teaching	she	was	not	long	in	taking
Source Snippet	0	0	0	0	0	0	0	0	0	0	0
BEGIN	0	100	99	98	97	96	95	94	93	92	91
Even	0	99	99	98	97	96	95	94	93	92	91
with	0	98	98	98	97	96	95	94	93	92	91
such	0	97	97	198	197	196	195	194	193	192	191
instruction	0	96	96	197	197	196	195	194	193	192	191
she	0	95	95	196	196	297	296	295	294	293	292
spent	0	94	94	195	195	296	296	295	294	293	292
little	0	93	93	194	194	295	295	295	294	293	292
time	0	92	92	193	193	294	294	294	294	293	292
taking	0	91	91	192	192	293	293	293	293	293	393

Ident	Correctly matched identical fragments/words
SPS	Detected Same Polarity Substitution

FIGURE 6.5: Scoring (Similarity) Matrix Constructed Using the Smith Waterman Algorithm for Snippet Pair 17732-33 in the P4P Corpus

Compared to the use of other algorithms for matching identical contexts, (such as the Greedy String Tiling (Wise, 1995)), the Smith Waterman Algorithm effectively deals with problems arising due to misalignment of identical contexts. This point can be illustrated using the following example.

In the partial fragment pair (17614-15, P4P corpus) shown in Figure 6.6, the phrase 'wished to' can be mismatched due to its later occurrence in the paraphrased snippet. The correct alignment here should be 'wished' ↔ 'determined'. Using the Smith Waterman Algorithm with our proposed scoring scheme effectively removes incorrectly matched identical fragments as shown in Figure 6.7.

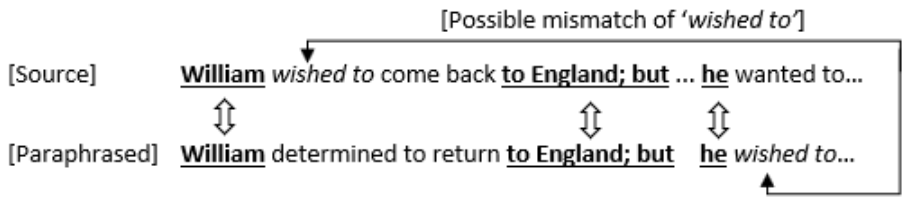


FIGURE 6.6: Possible Mismatch of Identical Text Fragments

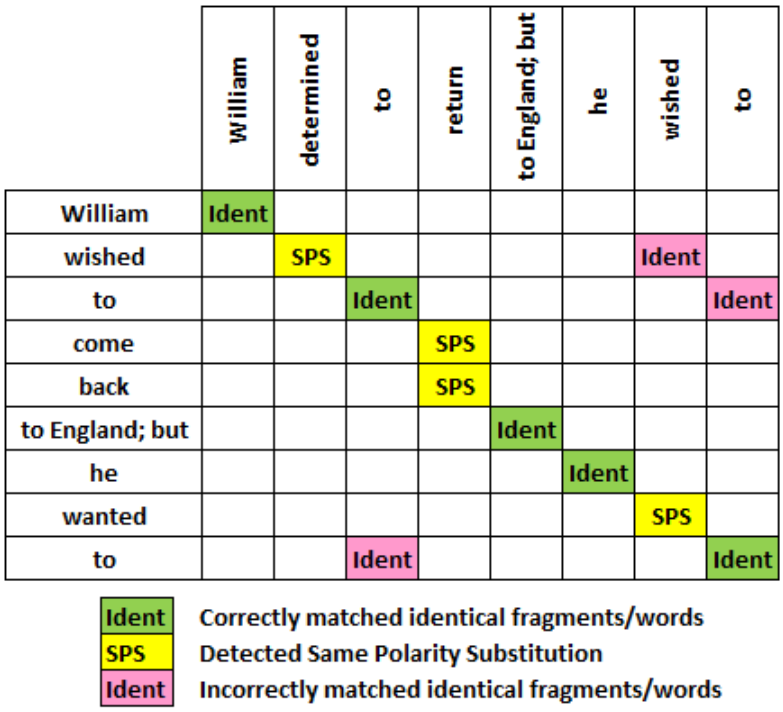


FIGURE 6.7: Word Similarity Matrix for Snippet Pair 17614-15 in the P4P Corpus

Word Embedding Based Same-Polarity Substitutions:

Our second method of extracting same polarity substitutions relies on using pretrained word embedding based similarity. We use the ConceptNet Numberbatch pretrained word embeddings (Speer, Chin, and Havasi, 2017) for detecting same polarity substitutions. There are two ways in which we utilise pretrained word embedding similarity scores: (a) for the detection of non-contextual same polarity substitutions, i.e., substitutions for which contexts may not match, and (b) for bifurcating detected multiword substitutions.

Since the size of the vector matrix in ConceptNet Numberbatch is much larger than the P4P Corpus, we filtered the vector matrix to include only those words and short phrases that are present within the P4P and MSRP-A corpora. For ConceptNet Numberbatch, this reduced the size from 4 million words to around 10,923 unique words and phrases. A similar process was carried out for the MSRP-A corpus which reduced the size of the vector matrix to 11,870 words and phrases.

For detecting *non-contextual same polarity substitutions*, we generate a word similarity matrix, however the entries of this matrix are now filled with pairwise word cosine similarity values. Word cosine similarity for two word vectors $\vec{w}_s$ and $\vec{w}_y$ is defined as:

$$sim(\vec{w}_s, \vec{w}_y) = \frac{\sum_{i=1}^{n} \vec{w_{si}} \cdot \vec{w_{yi}}}{|\vec{w_{si}}| \cdot |\vec{w_{yi}}|} \tag{6.6}$$

Word pairs which have similarity values above a certain threshold are considered as same polarity substitutions. However, we exclude stopwords from this criterion as word embedding similarity values of stopword pairs can be quite high, but these might be unlikely same polarity substitutions. Therefore, we consider only content (non-stopword) pairs for this phase.

Figure 6.8 shows a word similarity matrix for the snippet pair (16952-53) from the P4P corpus. We highlight the highest similarity in each row and column subject to the constraint that it is not a stopword with similarity value ≥ 0.35 (threshold). Identical word values are also marked in this Figure.

	While	the	prognosis	is	favourable	on	the	whole
The	0.39	1.00	-0.10	0.47	0.01	0.50	1.00	0.33
outlook	0.02	0.00	0.50	0.00	0.20	0.00	0.00	0.09
overall	0.12	0.16	0.16	0.07	0.09	0.02	0.16	0.39
seems	0.23	0.14	0.06	0.54	0.00	0.09	0.14	0.11
to	0.26	0.47	-0.10	0.30	0.06	0.37	0.47	0.14
be	0.11	0.18	0.01	0.60	0.03	0.19	0.18	0.03
good	0.03	0.02	0.05	0.15	0.41	0.02	0.02	0.11

Ident	Matched identical fragments/words
SPS	Correctly Detected Same Polarity Substitution
Ident	Discarded high value Same Polarity Substitutions (Stopwords)

FIGURE 6.8: Word Similarity Matrix (with similarity values) for the Snippet Pair 16952-53

The following non-contextual same polarity substitutions are derived from this snippet pair:

1. outlook ↔ prognosis
2. good ↔ favourable
3. overall ↔ whole

Our second method of using word embedding similarity scores is through *bifurcation of multiple word substitutions* into mappings of individual words, where possible. This is especially useful, when multiword substitutions are detected through

context matching. This type of bifurcation has been proposed by Brockett (2007) shown in Figure 6.9.

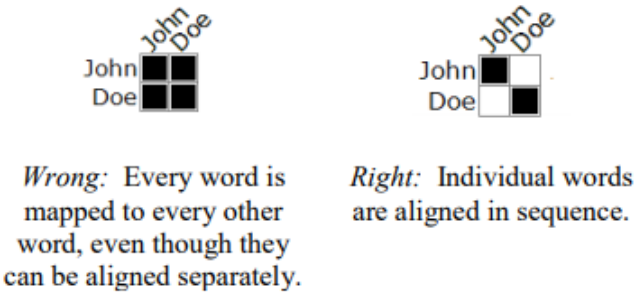


FIGURE 6.9: Annotation Guidelines showing Bifurcation for the RTE-2006 Dataset (Brockett, 2007)

Bifurcation of substitutions detected by contextual matching is useful qualitatively, since it helps in making an accurate mapping of the constituent words. Furthermore, from a quantitative perspective also, it contributes towards the precision, recall and F_1 score by increasing the correctly detected same polarity substitutions.

We perform bifurcation, i.e., mapping of content words only in a multiword substitution based on their word embedding similarity scores. Figure 6.10 shows a bifurcation of the expressions 'torrid persuasiveness' vs. 'passionate eloquence' in snippet pair 17530-31.

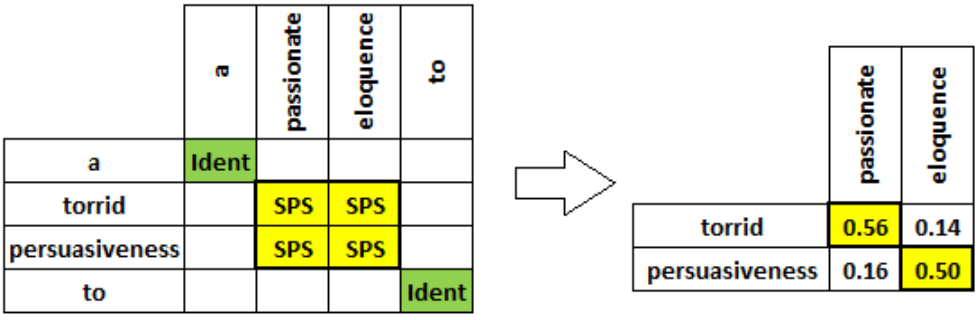


FIGURE 6.10: Bifurcation of Same Polarity Substitution based on Word Embedding Similarity Score (Snippet Pair 17530-31, P4P Corpus)

6.3.3 Merging Same-Polarity Substitutions

Both contextual and non-contextual same polarity substitutions detected using the previous two methods are now merged to form a single set of same polarity substitutions. A variety of techniques are used for this merging process as stated next.

Duplicates are removed if there is a total overlap between a context-based substitution and an embeddings based substitution. In case of a partial overlap, i.e., a match of source words only but different words from the paraphrased snippet (or vice versa), the context-based substitution is given preference. Embedding based substitutions that are a subset of a context-based substitution (i.e., a word pair match within a phrase pair) are used towards bifurcation as discussed above.

6.3.4 Filtering

In this final stage of our pipeline, some of the detected substitutions found via contexts or embeddings might be of some other type. For example, the substitution 'criticized' $\leftrightarrow$ 'criticised' is annotated as a spelling change (Figure 6.2), although the ConceptNet Numberbatch word embedding similarity score of the pair ('criticized', 'criticised') is 0.76.

In this stage, we remove primarily three paraphrase types from the detected same polarity substitutions, i.e., spelling changes, inflectional changes and modal verb changes. These changes are mostly at the granularity of individual words and not phrases. Furthermore, some miscellaneous detected substitutions such as the ones containing numerical information in the form of digits as well as abbreviations are also filtered. These filtered types are described in more detail as follows.

1. All word pairs such that the detected substitution from the paraphrased snippet pair ends in an 's' or an 'es' appended to the source detection are removed from the set of detected same polarity substitutions. (e.g., mango, mangoes were removed). These correspond to an inflectional change.

2. Detected same polarity substitution pairs with spelling errors are removed, such as (appeareance, appearance). Furthermore, US/British English spelling variants are also removed, e.g. (labour, labour).

3. Modal verbs such as ('should', 'could') or ('can', 'may') are also removed as these are classified as modal verb paraphrase type.

4. Miscellaneous detected types such as the ones containing numerical information (3, three), abbreviations (that is, i.e.) and slang forms of pronouns such as (you've, uve) are also removed.

This step of filtering completes our approach for detecting same polarity substitutions.

6.4 Results and Discussion

Our combined approach of preprocessing, detection of same polarity substitutions using matching of contexts, word embedding similarity scores and filtering results in the F_1 scores of 0.584 for the P4P corpus and 0.565 for the MSRP-A corpus respectively. A detailed breakdown of the scores in terms of precision and recall along with various other approaches is shown in Table 6.1 for the P4P corpus and Table 6.2 for the MSRP-A corpus. Since the problem of paraphrase type identification in general and same polarity substitution detection in particular has not yet been addressed in plagiarism detection literature, we use two different baselines to compare our work with.

1. **Context-Based Substitutions:** Context-based approaches have been used in earlier works for domain specific synonym detection. In an earlier work by Grigonyte et al. (2010), domain specific synonymous pairs were identified in sentence pairs using paraphrase casts. Their stated approach uses the Smith Waterman Algorithm with one synonymous pair reported per sentence pair. For comparison purposes, we consider our context-based approach using the Smith Waterman Algorithm with the detection of multiple synonym pairs, as detecting a single synonymous pair for each snippet pair might lead to low scores. In addition, we also compare our approach by applying the Greedy String Tiling Algorithm (Wise, 1995) to detect identical string tiles, which are used as contexts for detecting same polarity substitutions.

2. **WordNet and Word Embedding Similarity Based Measures:** We use the cosine word similarity measure from three pretrained word embeddings, i.e., ConceptNet Numberbatch (Speer, Chin, and Havasi, 2017), FastText (Mikolov et al., 2018) and GloVe (Pennington, Socher, and Manning, 2014) for comparison with our proposed approach. From a plagiarism detection perspective, Kanjirangat and Gupta (2018) and Carmona et al. (2018) use the WUP WordNet Similarity (Wu and Palmer, 1994) in their respective proposed semantic-syntactic similarity measures for sentences used for paraphrase plagiarism detection. A comparison of our approach with the application of WUP WordNet similarity measure is also included.

P4P Corpus		Precision	Recall	F_1 score
	Combined Approach	**0.509**	**0.684**	**0.584**
Context Based Approaches	SW Algorithm	0.484	0.656	0.557
	Greedy String Tiling	0.516	0.573	0.543
Word Embedding (Cosine) Similarity Based Measures	Concept Net	0.504	0.376	0.426
	FastText	0.411	0.403	0.400
	GLoVe	0.325	0.310	0.312
WordNet Based	WUP Similarity	0.109	0.104	0.119

TABLE 6.1: Precision, Recall and F_1 scores for the P4P corpus using various approaches

MSRP-A Corpus		Precision	Recall	F_1 score
	Combined Approach	**0.483**	**0.681**	**0.565**
Context Based Approaches	SW Algorithm	0.471	0.663	0.550
	Greedy String Tiling	0.497	0.621	0.552
Word Embedding (Cosine) Similarity Based Measures	Concept Net	0.440	0.238	0.304
	FastText	0.366	0.277	0.311
	GLoVe	0.292	0.243	0.251
WordNet Based	WUP Similarity	0.083	0.077	0.076

TABLE 6.2: Precision, Recall and F_1 scores for the MSRP-A corpus using various approaches

Tables 6.1 and 6.2 show the Precision, Recall and F_1 scores for various approaches, which have been classified into context-based, word embedding similarity based and WordNet based approaches. It can be observed that the combined approach of using contexts, ConceptNet Numberbatch word embeddings and filtering outperforms all other approaches, including the use of other word embeddings, i.e. FastText, GLoVe, as well as WUP Similarity using WordNet. Furthermore, it can be observed that the context-based approaches perform better than word embeddings and WordNet based similarity measures.

Several reasons can be attributed to this point which are outlined as follows:

1. **Phrasal Same Polarity Substitutions**: A large number of same polarity substitutions (approximately 40% of the entire substitutions for the P4P corpus) are phrasal or multiword substitutions. Word Similarity based measures may only partially assist in the detection of a phrasal substitution. This is because most word similarity measures calculate similarity between individual words, and therefore may contribute partially (e.g., bifurcation) to the detection of the entire phrasal substitution. The following partial snippet pair (P4P Corpus, 16604-65) highlights this point:

   (a) ... and a desire for attention especially...

   (b) ... and bold ostentation especially...

   In this snippet pair the phrases 'desire for attention' $\leftrightarrow$ 'bold ostentation' form a same polarity substitution. Using word similarity scores by applying word embeddings or WordNet, we find that the similarity scores between individual words for these two phrases are low. However, by analyzing the contexts, the same polarity substitution 'a desire for attention' $\leftrightarrow$ 'bold ostentation' can be readily detected.

2. **Stopwords** A considerable number of same polarity substitutions (approximately 10% for the P4P Corpus) are between stopwords which may include pronouns and prepositions. Detecting such same polarity substitutions is considerably simpler using matching of contexts (if possible) as compared to using word similarity scores, since word similarity scores may lead to low precision by detecting a large number of unannotated pairs. The following partial snippet pair (P4P Corpus, Snippet 17020-21) illustrates this point:

   (a) ...Feed themselves with any material...

   (b) ...Nourish themselves on any material...

   In the above example, the same polarity substitution 'with' $\leftrightarrow$ 'on' is between stopwords, which are prepositions in this case.

3. **Coreferences and Pronouns**: Same Polarity Substitutions which replace a proper noun with a pronoun may require methods such as coreference resolution for detection. However, use of context matching can considerably simplify detection of this type of same polarity substitution, as illustrated by the following example (MSRP-A Corpus, Snippet 29020-29021):

   (a) He said the foodservice pie business doesn't fit the company's long-term growth strategy.

   (b) "The foodservice pie business does not fit our long-term growth strategy".

In the above example, the phrase "the company ↔ our" is a same polarity substitution. Compared to the use of contexts, word similarity scores are not helpful in detecting this type of same polarity substitution since the word pair (our, company) has overall low similarity values.

6.4.1 Observations

From Tables 6.1 and 6.2 we also find that within the results of word embeddings based approaches and the WUP Similarity metric, ConceptNet Numberbatch pretrained word embeddings give the highest F_1 scores. This result reinforces the usefulness of ConceptNet Numberbatch as a state-of-the-art pretrained word embedding, providing a useful tool for the detection of non-contextual (as well as confirming contextual) content word same polarity substitutions.

Finally, the recall results of our combined approach for both the P4P (and the MSRP-A corpus) are slightly above 0.68, suggesting that slightly more than two-thirds of the same polarity substitutions have been detected. This result can be explained by the point that the task of detecting same polarity substitutions within the P4P corpus can be considered as challenging. This point can be partially corroborated by the fact that of the several state-of-the-art approaches, none could score more than an F_1 score of 0.20 in *detecting* plagiarism cases of the P4P corpus (Barrón-Cedeño et al., 2013).

6.5 Conclusions and Future Work

In this chapter, we applied the use of contexts, word embedding similarity and filtering for the detection of same polarity substitutions in plagiarised (paraphrased) text. We found that the use of context matching and word embeddings are the most effective tools for the detection of same polarity substitutions, the most commonly occurring type of paraphrase plagiarism. Furthermore, within pretrained word embeddings, ConceptNet Numberbatch has shown to be the most effective word embedding for detecting paraphrase types. From a plagiarism detection perspective, this work has the potential to contribute towards detecting paraphrase types in plagiarised text, thereby assisting a human evaluator in determining whether plagiarism has occurred in reported cases of textual similarity.

Further improvement of the current results can be achieved by exploring the relationship between a context and associated substitution. This could be achieved using more intensive Natural Language Processing methods; however, the efficiency of such operations has to kept in consideration.

Chapter 7

Conclusions and Future Work

Plagiarism and its detection have seen several advances in the past several years and the field has evolved rapidly. These advances have resulted in not only a large collection of detection techniques for various types of plagiarism, but also the availability of a number of plagiarism detection systems, both in the academic and the scientific domain. The underlying motivation for this interest is the rapid rise in the incidences of plagiarism due to the widespread availability of information technology and the ease with which content can be copied.

This thesis focused on several aspects within plagiarism detection by advancing the field on several fronts. From a plagiarism detection perspective, we proposed a cascade of detection approaches for detecting several types of plagiarism which integrate customized sub-approaches corresponding to each obfuscation type. We also extended the scope of obfuscation types by creating a corpus using retold versions of short stories. For obfuscation types pertaining to technical disguise (such as homoglyph substitution) which resulted in a clear evasion of plagiarism detection, we proposed two techniques with a relevant comparison. From a paraphrase type identification perspective, we proposed contextual and non-contextual methods to detect same polarity substitutions using the Smith Waterman Algorithm and ConceptNet Numberbatch pretrained word embeddings.

Research conducted in this thesis has the potential to add further functionality to the current generation of plagiarism detection systems. It has been well established that plagiarism detection systems have benefited from the current and past research in plagiarism detection and currently "*production ready systems visually present results to the users and should be able to identify duly quoted text.*" (Foltỳnek, Meuschke, and Gipp, 2019). The overall research aim of this thesis was to provide research contributions that enhance the functionality of currently available plagiarism detection systems.

This chapter is organized as follows: Section 7.1 summarizes the key findings of the thesis highlighting the various contributions towards each research objective. Section 7.2 provides directions towards future research for plagiarism detection. Section 7.3 concludes this thesis.

7.1 Findings of the Thesis

The key findings of the thesis are described in more detail as follows:

1. A composite, cascading framework for plagiarism detection corresponding to various obfuscation types can be designed if discerning criteria between various obfuscation types can be identified. This design is modular in nature in the sense that it adapts to an inclusion of customized sub-approaches for detecting plagiarism of each obfuscation type. Using this framework we successfully integrated two approaches that correspond to state-of-the-art scores for the PAN-2014 plagiarism detection corpus in the no-obfuscation and summary obfuscation categories.

2. Summary obfuscation, which has proven to be the most challenging obfuscation type in PAN-2013 and PAN-2014 text alignment corpora, can be detected by considering the distances between matching text fragments in the source and suspicious documents respectively. Since summary by definition is a condensed form of a document, we expect a much shorter distance between matching text fragments in the plagiarised summary as opposed to in the main document. This approach of considering difference of these distances produced best performing scores for summary obfuscation detection, effectively outperforming all other approaches.

3. Technical disguise in the form of homoglyph substitution can prove to be a very effective technique in hiding plagiarised text and evading detection using plagiarism detection techniques. However, two simple approaches, i.e., (a) using the Unicode list of confusable characters to substitute homoglyphs, and (b) approximate word matching using normalized hamming distance, can prove to be effective in detecting plagiarism that has been obfuscated using homoglyph substitution.

4. Same Polarity Substitutions can be detected using both contextual methods, as well as non-contextual methods such as word embedding similarity measures. In general, contextual methods result in the detection of a much larger proportion of same polarity substitutions, with non-contextual methods contributing towards a smaller, but significant, type of same polarity substitution. For contextual methods, the Smith Waterman Algorithm is an effective technique, while ConceptNet Numberbatch word embeddings produce improved detection results as compared to other pretrained word embeddings for the detection of non-contextual same polarity substitutions.

7.1.1 Impact of Research

Plagiarism detection systems have utilized current and past research by implementing plagiarism detection methods. Several research contributions have been presented in this thesis as presented in Section 7.1. In particular, the proposed method for detecting summary obfuscation has the potential for opening a new line of research. This could be particularly impactful as plagiarism detection methods for summary obfuscation have produced lower plagdet scores compared to other obfuscation types. In addition, research methods presented on detecting same polarity substitutions in paraphrased, plagiarized text have the potential to enhance the current generation of plagiarism detection systems and benefit the academic and scientific communities in general. Currently, the available plagiarism detection systems produce a similarity score, identify textual matches and present results to the user in a visual format. This functionality can be expanded to include identification of paraphrase types, specially same polarity substitutions, as well as identification of an extended range of obfuscation types.

7.2 Future Work

This thesis addressed monolingual extrinsic plagiarism detection from several viewpoints, making small but effective contributions towards each of its research objectives. For future research, not only these research areas can be expanded further, but new research problems could also be explored, stated as follows.

1. From a plagiarism detection perspective, the approach of using an integrated framework could be automated further using machine learning approaches based on supervised and unsupervised learning such as regression and support vector machines (Foltỳnek, Meuschke, and Gipp, 2019). Furthermore, there is also a need to expand the scope of the problem as well from the current set of obfuscation types (available within the PAN datasets) to an integrated dataset having a wide variety of obfuscation types. Application of advanced machine learning methods such as deep learning (LeCun, Bengio, and Hinton, 2015) could be used not only for classifying the correct obfuscation type, but also for identifying the plagiarised segments of text from within the identified obfuscation types.

2. Experiments with several combinations of distance measures could be carried out for detecting summary obfuscation. Although we applied standard deviation (which is a statistical measure used to calculate deviation from the mean), its application produced improved results over the existing detection results for summary obfuscation, since this agrees with the basic methodology of how

summaries are produced. This could be extended further to include extended distance measures and observe the improvement as well as limitations of this approach.

3. For technical disguise plagiarism, our research focused on homoglyph substitutions only. However, this is a much broader area of research with research efforts required on plagiarism detection when other types of technical tricks have been employed. For example, replacement of an entire block of text in the source document with its modified image in the suspicious document could be a form of technical disguise that is worth exploring.

4. Paraphrase Type Identification has the potential to enhance current batch of plagiarism detection systems. Although we focused on detecting same polarity substitutions, this research could be extended to include several paraphrase types. Furthermore, with the availability of datasets that have a large and varied number of paraphrase types (such as opposite polarity substitutions), word embeddings on such plagiarism datasets could be generated to form a repository of pretrained word embeddings suitable for paraphrase type identification. This direction of work could be further expanded to include advanced language models such as the BERT model (Devlin et al., 2019).

7.3 Conclusion

Plagiarism detection continues to be an active area of research. We expect that research contributions from this thesis will lead to better and more improved plagiarism detection systems and incorporation of paraphrase type identification within plagiarism detection systems. Furthermore, it will also motivate research aimed towards improved plagiarism detection systems both in the academic and scientific domains.

References

Agirre, Eneko et al. (2016). "SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 497–511. URL: https://www.aclweb.org/anthology/S16-1081.

Allen, Carl and Timothy M. Hospedales (2019). "Analogies Explained: Towards Understanding Word Embeddings". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, California, USA*, pp. 223–231. URL: http://homepages.inf.ed.ac.uk/thospeda/papers/allen2019analogies.pdf.

Alvi, Faisal, Mark Stevenson, and Paul Clough (2014). "Hashing and Merging Heuristics for Text Reuse Detection – Notebook for PAN at CLEF 2014". In: *Working Notes for CLEF 2014 Conference, Sheffield, United Kingdom*, pp. 939–946. URL: http://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-AlviEt2014.pdf.

Alvi, Faisal, Mark Stevenson, and Paul Clough (2015). "The Short Stories Corpus – Notebook for PAN at CLEF 2015". In: *Working Notes for CLEF 2015 Conference, Toulouse, France*. URL: http://ceur-ws.org/Vol-1391/90-CR.pdf.

Alvi, Faisal, Mark Stevenson, and Paul Clough (2017). "Plagiarism Detection in Texts Obfuscated with Homoglyphs". In: *European Conference on Information Retrieval, Aberdeen, United Kingdom*. Springer International Publishing, pp. 669–675. URL: https://link.springer.com/chapter/10.1007/978-3-319-56608-5_64.

Alzahrani, Salha and Naomie Salim (2010). "Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection-Lab Report for PAN at CLEF 2010". In: *CLEF 2010 LABs and Workshops, Notebook Papers, Italy*. URL: http://www.clef-initiative.eu/documents/71612/86374/CLEF2010wn-PAN-AlzahraniEt2010.pdf.

Alzahrani, Salha M., Naomie Salim, and Ajith Abraham (2012). "Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 42.2, pp. 133–149. URL: https://ieeexplore.ieee.org/document/5766764.

Apostolico, Alberto and Concettina Guerra (1987). "The Longest Common Subsequence Problem Revisited". In: *Algorithmica* 2.1-4, pp. 315–336. URL: https://link.springer.com/article/10.1007/BF01840365.

Baeza-Yates, Ricardo and Berthier Ribeiro-Neto (2011). "Modern Information Retrieval: The Concepts and Technology behind Search (Second Edition)". In: *Addision Wesley* 84.

Bailey, Jonathan (2017). *The Top 10 Plagiarism Stories of 2016*. accessed 20 June 2020. URL: http://www.ithenticate.com/plagiarism-detection-blog/the-top-10-plagiarism-stories-of-2016.

Bär, Daniel, Torsten Zesch, and Iryna Gurevych (2012). "Text Reuse Detection using a Composition of Text Similarity Measures". In: *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pp. 167–184. URL: https://www.aclweb.org/anthology/C12-1011/.

Bär, Daniel et al. (2012). "UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures". In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, pp. 435–440. URL: https://www.aclweb.org/anthology/S12-1059/.

Barrón-Cedeño, Alberto (2012). "On the Mono- and Cross-Language Detection of Text Re-use and Plagiarism". PhD thesis. Universitat Polytecnica De Valencia. URL: https://riunet.upv.es/bitstream/handle/10251/16012/tesisUPV3833.pdf.

Barrón-Cedeño, Alberto and Paolo Rosso (2009). "On Automatic Plagiarism Detection Based on n-Grams Comparison". In: *Advances in Information Retrieval, 31th European Conference on IR Research, ECIR 2009, France*, pp. 696–700. URL: https://link.springer.com/chapter/10.1007/978-3-642-00958-7_69.

Barrón-Cedeño, Alberto, Paolo Rosso, and José-Miguel Benedí (2009). "Reducing the Plagiarism Detection Search Space on the Basis of the Kullback-Leibler Distance". In: *Computational Linguistics and Intelligent Text Processing, 10th International Conference, CICLing 2009, Mexico*, pp. 523–534. URL: https://link.springer.com/chapter/10.1007/978-3-642-00382-0_42.

Barrón-Cedeño, Alberto et al. (2013). "Plagiarism meets Paraphrasing: Insights for the Next Generation in Automatic Plagiarism Detection". In: *Computational Linguistics* 39.4, pp. 917–947. URL: https://www.mitpressjournals.org/doi/10.1162/COLI_a_00153.

Barzilay, Regina, Kathleen R. McKeown, and Michael Elhadad (1999). "Information Fusion in the Context of Multi-Document Summarization". In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, pp. 550–557. URL: https://www.aclweb.org/anthology/P99-1071/.

Baždarić, Ksenija et al. (2012). "Prevalence of Plagiarism in Recent Submissions to the Croatian Medical Journal". In: *Science and engineering ethics* 18.2, pp. 223–239. URL: https://link.springer.com/article/10.1007/s11948-011-9347-2.

Bendersky, Michael and W. Bruce Croft (2009). "Finding Text Reuse on the Web". In: *Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain*. Pp. 262–271. URL: https://dl.acm.org/doi/abs/10.1145/1498759.1498835.

Bhagat, Rahul and Eduard H. Hovy (2013). "What Is a Paraphrase?" In: *Computational Linguistics* 39.3, pp. 463–472. URL: https://www.mitpressjournals.org/doi/full/10.1162/COLI_a_00166.

Bhagwani, Sumit, Shrutiranjan Satapathy, and Harish Karnick (2012). "sranjans : Semantic Textual Similarity using Maximal Weighted Bipartite Graph Matching". In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, pp. 579–585. URL: https://www.aclweb.org/anthology/S12-1085.pdf.

Bodenreider, Olivier (Jan. 2004). "The Unified Medical Language System (UMLS): Integrating Biomedical Terminology". In: *Nucleic Acids Research* 32.1, pp. D267–D270. ISSN: 0305-1048. URL: https://doi.org/10.1093/nar/gkh061.

Brockett, Chris (2007). *Aligning the RTE 2006 corpus*. Tech. rep. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2007-77.pdf.

Brown, Alan S. and Dana R. Murphy (1989). "Cryptomnesia: Delineating Inadvertent Plagiarism." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 15.3, pp. 432–442. URL: https://psycnet.apa.org/record/1989-24862-001.

Burkard, Rainer E. and Eranda Cela (1999). "Linear Assignment Problems and Extensions". In: *Handbook of Combinatorial Optimization*. Springer International Publishing, pp. 75–149.

Callison-Burch, Chris (2008). "Syntactic Constraints on Paraphrases Extracted from Parallel Corpora". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 196–205. URL: https://dl.acm.org/doi/10.5555/1613715.1613743.

Cambria, Erik and Bebo White (2014). "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]". In: *Computational Intelligence Magazine, IEEE* 9.2, pp. 48–57. URL: http://www.krchowdhary.com/ai/ai14/lects/nlp-research-com-intlg-ieee.pdf.

Carmona, Miguel Ángel Álvarez et al. (2018). "Semantically-Informed Distance and Similarity Measures for Paraphrase Plagiarism Identification". In: *Journal of Intelligent and Fuzzy Systems* 34.5, pp. 2983–2990. URL: https://doi.org/10.3233/JIFS-169483.

Cederberg, Scott and Dominic Widdows (2003). "Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction". In: *Proceedings of the Seventh Conference on Natural language Learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 111–118. URL: https://www.aclweb.org/anthology/W03-0415/.

Ceska, Zdenek and Chris Fox (2009). "The Influence of Text Pre-processing on Plagiarism Detection". In: *Recent Advances in Natural Language Processing, RANLP 2009, 14-16 September, 2009, Borovets, Bulgaria*, pp. 55–59. URL: https://www.aclweb.org/anthology/R09-1011/.

Ceska, Zdenek, Michal Toman, and Karel Jezek (2008). "Multilingual Plagiarism Detection". In: *Artificial Intelligence: Methodology, Systems, and Applications*. Springer International Publishing, pp. 83–92. URL: https://link.springer.com/chapter/10.1007/978-3-540-85776-1_8.

Charles, Walter G. (2000). "Contextual Correlates of Meaning". In: *Applied Psycholinguistics* 21.4, 505–524. URL: https://www.cambridge.org/core/journals/applied-psycholinguistics/article/contextual-correlates-of-meaning/E75E86E7B814A5F93C3A41BF42E37F9B#.

Chen, Chien-Ying, Jen-Yuan Yeh, and Hao-Ren Ke (2010). "Plagiarism Detection using ROUGE and WordNet". In: *Journal of Computing* 2.3, pp. 34–44. URL: https://pdfs.semanticscholar.org/0680/5a47af2011c827ed5fdd993c6cb6122de2ad.pdf.

Chong, Miranda (2013). "A Study on Plagiarism Detection and Plagiarism Direction Identification using Natural Language Processing Techniques". PhD thesis. University of Wolverhampton. URL: http://clg.wlv.ac.uk/papers/chong-thesis.pdf.

Chong, Miranda and Lucia Specia (2011). "Lexical Generalisation for Word-level Matching in Plagiarism Detection." In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing 2011*, pp. 704–709. URL: https://www.aclweb.org/anthology/R11-1102/.

Chong, Miranda, Lucia Specia, and Ruslan Mitkov (2010). "Using Natural Language Processing for Automatic Detection of Plagiarism". In: *Proceedings of the 4th International Plagiarism Conference (IPC 2010), Newcastle, UK*. URL: https://pdfs.semanticscholar.org/636d/4c0b0fe6919abe6eb546907d28ed39bf56e6.pdf.

Chow, Tommy W. S. and M. K. M. Rahman (2009). "Multilayer Som with tree-structured Data for Efficient Document Retrieval and Plagiarism Detection". In: *IEEE Transactions on Neural Networks* 20.9, pp. 1385–1402. URL: https://dl.acm.org/doi/10.1109/TNN.2009.2023394.

Clough, Paul (2000). "Plagiarism in Natural and Programming Languages: An Overview of Current Tools and Technologies". In: *Research Memoranda: CS-00-05, Department of Computer Science, University of Sheffield, UK*, pp. 1–31. URL: https://pdfs.semanticscholar.org/8db6/7141b9e7a2a80ced6550e4c0da2d958c3b86.pdf.

Clough, Paul (2001). "Measuring Text Reuse and Document Derivation". Postgraduate Transfer Report. Department of Computer Science, University of Sheffield. URL: "http://ir.shef.ac.uk/cloughie/papers/transfer.pdf".

Clough, Paul (2003a). "Measuring Text Reuse". PhD thesis. University of Sheffield. URL: https://ir.shef.ac.uk/cloughie/papers/thesis.pdf.

Clough, Paul (2003b). "Old and New Challenges in Automatic Plagiarism Detection". In: *National Plagiarism Advisory Service*. University of Sheffield. URL: https://ir.shef.ac.uk/cloughie/papers/pas_plagiarism.pdf.

Clough, Paul and Mark Stevenson (2011). "Developing a Corpus of Plagiarised Short Answers". In: *Language Resources and Evaluation* 45.1, pp. 5–24. URL: https://link.springer.com/article/10.1007/s10579-009-9112-1.

Clough, Paul et al. (2002). "METER: Measuring Text Reuse". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 152–159. URL: https://www.aclweb.org/anthology/P02-1020.pdf.

Cormen, Thomas H. et al. (2009). *Introduction to Algorithms*. MIT press.

Costello, A (2003). "RFC3492-Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)". In: *Network Working Group*. Online: Accessed: 2016-10-19. URL: {http://www.ietf.org/rfc/rfc3492.txt}.

Dale, Robert (2010). "Classical Approaches to Natural Language Processing". In: *Handbook of Natural Language Processing*. Ed. by Nitin Indurkhya and Fred J. Damerau. Boca Raton: CRC Press, pp. 3–8.

Derose, Steven J. (1990). "Stochastic Methods for Resolution of Grammatical Category Ambiguity in Inflected and Uninflected Languages". UMI Order No: GAX90-02217. PhD thesis. Providence, RI, USA: Brown University.

Devlin, Jacob et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. URL: https://www.aclweb.org/anthology/N19-1423.pdf.

Dias, Paulo C. and Ana Sofia C. Bastos (2014). "Plagiarism Phenomenon in European Countries: Results from GENIUS Project". In: *Procedia-Social and Behavioral Sciences* 116, pp. 2526–2531. URL: https://www.sciencedirect.com/science/article/pii/S1877042814006223.

Dice, Lee R. (1945). "Measures of the Amount of Ecologic Association between Species". In: *Journal of Ecology* 26, pp. 297–302. URL: https://www.jstor.org/stable/1932409.

Dolan, Bill, Chris Quirk, and Chris Brockett (2004). "Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources". In: *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, p. 350. URL: https://www.aclweb.org/anthology/C04-1051/.

Dorr, Bonnie J. et al. (2004). "Semantic Annotation and Lexico-Syntactic Paraphrase". In: *Proceedings of the Workshop on Building Lexical Resources from Semantically Annotated Corpora (LREC-2004). Portugal*. URL: https://pdfs.semanticscholar.org/5140/d30bcd4270510b6abc0d464af53938f03761.pdf.

Dreher, Heinz (2007). "Automatic Conceptual Analysis for Plagiarism Detection". In: *Issues in Informing Science and Information Technology* 4, pp. 601–628. URL: https://espace.curtin.edu.au/handle/20.500.11937/33407.

Eisa, Taiseer Abdalla Elfadil, Naomie Salim, and Salha Alzahrani (2015). "Existing Plagiarism Detection Techniques: A Systematic Mapping of the Scholarly Literature". In: *Online Information Review* 39.3, pp. 383–400. URL: https://doi.org/10.1108/OIR-12-2014-0315.

Elhadi, Mohamed and Amjad Al-Tobi (2008). "Use of Text Syntactical Structures in Detection of Document Duplicates". In: *Third International Conference on Digital Information Management. ICDIM 2008*, pp. 520–525. URL: https://ieeexplore.ieee.org/document/4746719.

Ewing, Helen et al. (2019). "Student and Faculty Perceptions of Plagiarism in Health Sciences Education". In: *Journal of Further and Higher Education* 43.1, pp. 79–88. URL: https://www.tandfonline.com/doi/full/10.1080/0309877X.2017.1356913.

Foltýnek, Tomáš, Norman Meuschke, and Bela Gipp (2019). "Academic Plagiarism Detection: A Systematic Literature Review". In: *ACM Computing Surveys (CSUR)* 52.6, pp. 112.1–112.42. URL: https://dl.acm.org/doi/10.1145/3345317.

Franco-Salvador, Marc et al. (2015). "PAN 2015 Shared Task on Plagiarism Detection: Evaluation of Corpora for Text Alignment – Notebook for PAN at CLEF 2015". In: *Working Notes for CLEF 2015 Conference, Toulouse, France*. URL: http://ceur-ws.org/Vol-1391/inv-pap12-CR.pdf.

Franco-Salvador, Marc et al. (2016). "Cross-Language Plagiarism Detection over Continuous Space and Knowledge Graph-Based Representations of Language". In: *Knowledge Based Systems* 111.C, 87–99. URL: https://doi.org/10.1016/j.knosys.2016.08.004.

Freitag, Dayne et al. (2005). "New Experiments in Distributional Representations of Synonymy". In: *Proceedings of the Ninth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pp. 25–32. URL: http://dl.acm.org/citation.cfm?id=1706543.1706548.

Fu, Anthony Y., Xiaotie Deng, and Liu Wenyin (2006). "REGAP: A Tool for Unicode-Based Web Identity Fraud Detection". In: *Journal of Digital Forensic Practice* 1.2, pp. 83–97. URL: https://www.cs.cityu.edu.hk/~liuwy/publications/REGAP-J\%20of\%20Digital\%20Forensic\%20Practice.pdf.

Fujita, Atsushi (2005). "Automatic Generation of Syntactically Well-formed and Semantically Appropriate Paraphrases". PhD thesis. URL: https://library.naist.jp/mylimedio/dllimedio/showpdf2.cgi/DLPDFR003612_P1-99.

Gillam, Lee, John Marinuzzi, and Paris Ioannou (2010). "Turnitoff-Defeating Plagiarism Detection Systems". In: *Proceedings of the 11th Higher Education Academy - ICS Annual Conference. Higher Education Academy*. URL: http://epubs.surrey.ac.uk/790662/2/HEA-ICS_turnitoff.pdf.

Gipp, Bela (2014). "Citation-based Plagiarism Detection". In: *Citation-based Plagiarism Detection*. Springer International Publishing, pp. 57–88. URL: https://www.springer.com/gp/book/9783658063931.

Glinos, Demetrios G. (2014). "Discovering Similar Passages within Large Text Documents". In: *Proceedings of the 5th International Conference of the CLEF Initiative, CLEF 2014, Sheffield, United Kingdom*, pp. 98–109. URL: https://link.springer.com/chapter/10.1007/978-3-319-11382-1_10.

Goddard, Cliff and Andrea C Schalley (2010). "Semantic Analysis". In: *Handbook of Natural Language Processing*. Ed. by Nitin Indurkhya and Fred J. Damerau. Boca Raton: CRC Press, pp. 93–121.

Gorbenko, Anna and Vladimir Popov (2012). "The Longest Common Subsequence Problem". In: *Applied Mathematical Sciences* 6.116, pp. 5781–5787. URL: http://m-hikari.com/asb/asb2012/asb5-8-2012/popovASB5-8-2012-1.pdf.

Grigonyte, Gintare et al. (2010). "Paraphrase Alignment for Synonym Evidence Discovery". In: *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pp. 403–411. URL: http://aclweb.org/anthology/C10-1046.

Grman, Jan and Rudolf Ravas (2011). "Improved Implementation for Finding Text Similarities in Large Collections of Data - Notebook for PAN at CLEF 2011". In: *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship,*

and Social Software Misuse. URL: http://www.clef-initiative.eu/documents/71612/86377/CLEF2011wn-PAN-GrmanEt2011.pdf.

Guégan, Marie and Nicolas Hernandez (2006). "Recognizing Textual Parallelisms with Edit Distance and Similarity Degree". In: *11th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Trento, Italy*. URL: https://www.aclweb.org/anthology/E06-1036.pdf.

Gülich, Elisabeth (2003). "Conversational Techniques used in Transferring Knowledge between Medical Experts and Non-experts". In: *Discourse Studies* 5.2, pp. 235–263. URL: https://journals.sagepub.com/doi/10.1177/1461445603005002005.

Gustafson, Nathaniel, Maria Soledad Pera, and Yiu-Kai Ng (2008). "Nowhere to Hide: Finding Plagiarized Documents based on Sentence Similarity". In: *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, pp. 690–696. URL: https://ieeexplore.ieee.org/document/4740531.

Guthrie, David et al. (2006). "A Closer Look at Skip-gram Modelling". In: *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006)*, pp. 1–4. URL: https://www.aclweb.org/anthology/L06-1210/.

HaCohen-Kerner, Yaakov and Aharon Tayeb (2017). "Rapid Detection of Similar Peer-Reviewed Scientific Papers via Constant Number of Randomized Fingerprints". In: *Information Processing and Management* 53.1, 70–86. URL: https://www.sciencedirect.com/science/article/abs/pii/S0306457316302254.

Hearst, Marti A. (1992). "Automatic Acquisition of Hyponyms from Large Text Corpora". In: *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, pp. 539–545. URL: https://www.aclweb.org/anthology/C92-2082/.

Hearst, Marti A. (1998). "Automated Discovery of WordNet Relations". In: *WordNet: an electronic lexical database*, pp. 131–153. URL: http://people.ischool.berkeley.edu/~hearst/papers/wordnet98.pdf.

Heather, James (2010). "Turnitoff: Identifying and Fixing a Hole in Current Plagiarism Detection Software". In: *Assessment & Evaluation in Higher Education* 35.6, pp. 647–660. URL: https://srhe.tandfonline.com/doi/full/10.1080/02602938.2010.486471.

Hoad, Timothy C. and Justin Zobel (2003). "Methods for Identifying Versioned and Plagiarized Documents". In: *Journal of the American Society for Information Science and Technology* 54.3, pp. 203–215. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.10170.

Huang, Shudong et al. (2002). *Multiple-translation Chinese Corpus*. Linguistic Data Consortium, University of Pennsylvania.

Irving, Robert W. (2004). *Plagiarism and Collusion Detection using the Smith-Waterman Algorithm*. Tech. rep. TR-2004-164. University of Glasgow.

Iyengar, Rishi (2015). *200 South Korean Professors Charged in Massive Plagiarism Scam*. accessed 20 June 2020. URL: https://www.time.com/4126853/south-korea-professors-books-plagiarism-/.

Jaccard, Paul (1912). "The Distribution of the Flora of the Alpine Zone". In: *New Phytologist* 11, pp. 37–50. URL: https://www.scienceopen.com/document?vid=f920d164-bc91-4ad0-b03b-6d81e6cbbbb9.

Jackson, Peter and Isabelle Moulinier (2007). *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*. Vol. 5. John Benjamins Publishing. URL: https://benjamins.com/catalog/nlp.5.

Jing, Hongyan and Kathleen R. McKeown (1999). "The Decomposition of Human-written Summary Sentences". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 129–136. URL: http://www.cs.columbia.edu/nlp/papers/1999/jing_mckeown_99.pdf.

Joy, Mike et al. (2009). "A Taxonomy of Plagiarism in Computer Science". In: 1st International Conference on Education and New Learning Technologies. Barcelona, Spain: IATED, pp. 3372–3379.

Jurafsky, Daniel and James H. Martin (2008). *Speech and Language Processing (2nd Edition)*. Prentice Hall.

Kakkonen, Tuomo and Maxim Mozgovoy (2010). "Hermetic and Web Plagiarism Detection Systems for Student Essays — An Evaluation of the State-of-the-art". In: *Journal of Educational Computing Research* 42.2, pp. 135–159. URL: https://journals.sagepub.com/doi/pdf/10.2190/EC.42.2.a.

Kanjirangat, Vani and Deepa Gupta (2016). "Study on Extrinsic Text Plagiarism Detection Techniques and Tools." In: *Journal of Engineering Science & Technology Review* 9.5, pp. 150–164. URL: http://www.jestr.org/downloads/Volume9Issue4/fulltext23942016.pdf.

Kanjirangat, Vani and Deepa Gupta (2018). "Unmasking Text Plagiarism using Syntactic - Semantic based Natural Language Processing Techniques: Comparisons, Analysis and Challenges". In: *Information Processing & Management* 54.3, pp. 408–432. URL: http://sciencedirect.com/science/article/pii/S0306457317300547.

Karp, Richard M and Michael O Rabin (1987). "Efficient Randomized Pattern-Matching Algorithms". In: *IBM Journal of Research and Development* 31.2, pp. 249–260. URL: https://ieeexplore.ieee.org/abstract/document/5390135.

Kauffman, Yashu and Michael F. Young (2015). "Digital Plagiarism: An Experimental Study of the Effect of Instructional Goals and Copy-and-Paste Affordance". In:

Computers & Education 83, pp. 44–56. URL: http://sciencedirect.com/science/article/abs/pii/S0360131514002930.

Kullback, Solomon and Richard A. Leibler (1951). "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1, pp. 79–86. URL: https://www.jstor.org/stable/2236703?seq=1.

Landauer, Thomas K. (2006). *Latent Semantic Analysis*. Wiley Online Library.

Lane, Peter (2011). *UH Ferret : Implementation of a Copy-Detection Tool, Software*. https://uhra.herts.ac.uk/handle/2299/12041. [Online; accessed 28-May-2019].

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep Learning". In: *Nature* 521.7553, pp. 436–444. URL: http://nature.com/articles/nature14539.

Leung, Chi-Hong and Yuen-Yan Chan (2007). "A Natural Language Processing Approach to Automatic Plagiarism Detection". In: *Proceedings of the 8th ACM SIGITE conference on Information technology education*. ACM, pp. 213–218. URL: https://dl.acm.org/doi/10.1145/1324302.1324348.

Levenshtein, Vladimir I. (1966). "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals". In: *Soviet Physics Doklady*. Vol. 10, pp. 707–710. URL: https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf.

Liddell, Jean (2003). "A Comprehensive Definition of Plagiarism". In: *Community & Junior College Libraries* 11.3, pp. 43–52. URL: https://doi.org/10.1300/J107v11n03_07.

Lin, Chin-Yew (2004). "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Proceedings of the Workshop on: Text Summarization Branches Out (WAS 2004)*. URL: https://www.aclweb.org/anthology/W04-1013/.

Liu, Bing (2012). "Sentiment Analysis and Opinion Mining". In: *Synthesis Lectures on Human Language Technologies* 5.1, pp. 1–167. URL: https://www.morganclaypool.com/doi/abs/10.2200/S00416ED1V01Y201204HLT016.

Ljunglöf, Peter and Mats Wirén (2010). "Syntactic Parsing. (book chapter)". In: *Handbook of Natural Language Processing*. Ed. by Nitin Indurkhya and Fred J. Damerau. Boca Raton: CRC Press, pp. 59–91.

Mani, Inderjeet (2001). *Automatic Summarization*. Vol. 3. John Benjamins Publishing.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. Vol. 1. Cambridge University Press, Cambridge.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz (1993). "Building a Large Annotated Corpus of English: The Penn Treebank". In: *Computational linguistics* 19.2, pp. 313–330. URL: https://www.aclweb.org/anthology/J93-2004/.

Martin, Brian (1994). "Plagiarism: A Misplaced Emphasis". In: *Journal of Information Ethics* 3.2, pp. 36–47. URL: https://documents.uow.edu.au/~bmartin/pubs/94jie.html.

Maurer, Hermann A., Frank Kappe, and Bilal Zaka (2006). "Plagiarism-A Survey". In: *Journal of Universal Computer Science* 12.8, pp. 1050–1084. URL: https://pdfs.semanticscholar.org/2093/02b33726db59303707186707bd080e68fe59.pdf.

McKeever, Lucy (2006). "Online Plagiarism Detection Services — Saviour or Scourge?" In: *Assessment & Evaluation in Higher Education* 31.2, pp. 155–165. URL: https://www.tandfonline.com/doi/full/10.1080/02602930500262460.

Meuschke, Norman and Bela Gipp (2013). "State-of-the-Art in Detecting Academic Plagiarism". In: *International Journal for Educational Integrity* 9.1, pp. 50–71. URL: https://www.gipp.com/wp-content/papercite-data/pdf/meuschke13.pdf.

Meuschke, Norman, Bela Gipp, and Corinna Breitinger (2012). "CitePlag: A Citation-based Plagiarism Detection System Prototype". In: *Proceedings of the 5th International Plagiarism Conference, Newcastle upon Tyne, UK*. URL: https://pdfs.semanticscholar.org/9b78/e970a66f5015cff120c915bcffea63c3b851.pdf.

Mikolov, Tomas et al. (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems*, pp. 3111–3119. URL: https://dl.acm.org/doi/10.5555/2999792.2999959.

Mikolov, Tomas et al. (2018). "Advances in Pre-Training Distributed Word Representations". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. URL: https://www.aclweb.org/anthology/L18-1008/.

Miller, George A (1995). "WordNet: A Lexical Database for English". In: *Communications of the ACM* 38.11, pp. 39–41. URL: https://dl.acm.org/doi/10.1145/219717.219748.

Moraliyski, Rumen Valentinov (2013). "Discovery of Noun Semantic Relations based on Sentential Context Analysis". PhD thesis. University of Beira Interior, Portugal. URL: https://dias.users.greyc.fr/publications/thesis-Rumen.pdf.

Mozgovoy, Maxim, Tuomo Kakkonen, and Erkki Sutinen (2007). "Using Natural Language Parsers in Plagiarism Detection". In: *Speech and Language Technology in Education (SLaTE2007) Farmington, PA, USA*. URL: http://web-ext.u-aizu.ac.jp/~mozgovoy/homepage/papers/MKS07.pdf.

Mphahlele, Amanda and Sioux McKenna (2019). "The Use of Turnitin in the Higher Education Sector: Decoding the Myth". In: *Assessment & Evaluation in Higher Education*, pp. 1–11. URL: https://srhe.tandfonline.com/doi/full/10.1080/02602938.2019.1573971.

Navarro, Gonzalo (2001). "A Guided Tour to Approximate String Matching". In: *ACM Computing Surveys (CSUR)* 33.1, pp. 31–88. URL: https://dl.acm.org/doi/abs/10.1145/375360.375365.

Nawab, Rao Muhammad Adeel (2012). "Mono-lingual Paraphrased Text Reuse and Plagiarism Detection". PhD thesis. University of Sheffield. URL: http://etheses.whiterose.ac.uk/2785/.

Nawab, Rao Muhammad Adeel, Mark Stevenson, and Paul Clough (2011). "External Plagiarism Detection using Information Retrieval and Sequence Alignment-Notebook for PAN at CLEF 2011." In: *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*. URL: http://ceur-ws.org/Vol-1177/CLEF2011wn-PAN-NawabEt2011.pdf.

Needleman, Saul B. and Christian D. Wunsch (1970). "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins". In: *Journal of Molecular Biology* 48.3, pp. 443–453. URL: https://www.sciencedirect.com/science/article/pii/0022283670900574.

Nivre, Joakim (2006). *Inductive Dependency Parsing*. Springer International Publishing.

Oberreuter, Gabriel et al. (2011). "Approaches for Intrinsic and External Plagiarism Detection - Notebook for PAN at CLEF 2011". In: *CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands*. URL: http://ceur-ws.org/Vol-1177/CLEF2011wn-PAN-OberreuterEt2011.pdf.

Ohshima, Hiroaki and Katsumi Tanaka (2009). "Real Time Extraction of Related Terms by Bi-directional Lexico-Syntactic Patterns from the Web". In: *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*. ACM, pp. 441–449. URL: https://dl.acm.org/doi/10.1145/1516241.1516318.

Palkovskii, Yurii and Alexei Belov (2014). "Developing High-Resolution Universal Multi-type n-gram Plagiarism Detector". In: *Working Notes Papers of the CLEF 2014 Evaluation Labs*, pp. 984–989. URL: http://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-PalkovskiiEt2014.pdf.

Palkovskii, Yurii and Alexei Belov (2015). *Submission to the 7th International Competition on Plagiarism Detection*. http://www.uni-weimar.de/medien/webis/corpora/corpus-pan-labs-09-today/pan-15/pan15-data/pan15-text-alignment-training-dataset-palkovskii15-english-2015-05-24.zip. [Online; accessed 28-May-2019].

Palmer, David D. (2010). "Text Preprocessing. (book chapter) ". In: *Handbook of Natural Language Processing*. Ed. by Nitin Indurkhya and Fred J. Damerau. Boca Raton: CRC Press, pp. 9–30.

Parker, Kim, Amanda Lenhart, and Kathleen Moore (2011). *The Digital Revolution and Higher Education: College President, Public differ on the value of Online Learning*. Tech. rep. Pew Research Center, Social and Demographic Trends, Washington DC. URL: https://www.pewresearch.org/internet/2011/08/28/the-digital-revolution-and-higher-education/.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). "GloVe: Global Vectors for Word Representation." In: *Proceedings of the 2014 Empirical*

Methods in Natural Language Processing (EMNLP). Vol. 14, pp. 1532–1543. URL: https://www.aclweb.org/anthology/D14-1162.pdf.

Porter, Martin F. (1980). "An Algorithm for Suffix Stripping". In: *Program (Automated Library and Information Systems)* 14.3, pp. 130–137. URL: https://www.emerald.com/insight/content/doi/10.1108/eb046814/full/html.

Porter, Martin F. (2001). *Snowball: A Language for Stemming Algorithms*. URL: http://snowball.tartarus.org/texts/introduction.html.

Potthast, Martin (2011). "Technologies for Reusing Text from the Web". PhD thesis. Bauhaus-Universitat Weimar. URL: https://webis.de/downloads/publications/papers/potthast_2011b.pdf.

Potthast, Martin et al. (2009). "Overview of the 1st International Competition on Plagiarism Detection". In: *3rd PAN Workshop Uncovering Plagiarism, Authorship and Social Software Misuse*, pp. 1–9. URL: http://ceur-ws.org/Vol-502/paper1.pdf.

Potthast, Martin et al. (2010). "An Evaluation Framework for Plagiarism Detection". In: *Conference on Computational Linguistics (COLING) 2010: Posters*, pp. 997–1005.

Potthast, Martin et al. (2013). "Overview of the 5th International Competition on Plagiarism Detection". In: *Working Notes for CLEF 2013 Conference, Valencia, Spain*. ISBN: 978-88-904810-3-1. URL: http://ceur-ws.org/Vol-1179/CLEF2013wn-PAN-PotthastEt2013.pdf.

Potthast, Martin et al. (2014). "Overview of the 6th International Competition on Plagiarism Detection". In: *Working Notes for CLEF 2014 Conference, Sheffield, UK*. URL: http://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-PotthastEt2014.pdf.

Potthast, Martin et al. (2015). "Towards Data Submissions for Shared Tasks: First Experiences for the Task of Text Alignment". In: *Working Notes Papers of the CLEF 2015 Evaluation Labs, Toulouse, France*. URL: http://ceur-ws.org/Vol-1391/inv-pap11-CR.pdf.

Prechelt, Lutz, Guido Malpohl, and Michael Philippsen (2000). *JPlag: Finding Plagiarism among a Set of Programs*. Tech. rep. Fakultat fur Informatik Universitat Karlsruhe. URL: http://page.mi.fu-berlin.de/prechelt/Biblio/jplagTR.pdf.

Pupovac, Vanja, Lidija Bilic-Zulle, and Mladen Petrovecki (2008). "On Academic Plagiarism in Europe. An Analytical Approach based on Four Studies". In: *Digithum (The Humanities in the Digital Age)* 10, pp. 13–19. URL: https://pdfs.semanticscholar.org/154b/ec43956cfbd82b8bf6ffef1391b07f9ae431.pdf.

Qi, Ye et al. (2018). "When and Why Are Pre-Trained Word Embeddings Useful for Neural Machine Translation?" In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 529–535. URL: https://www.aclweb.org/anthology/N18-2084.pdf.

Rajaraman, Anand and Jeffrey David Ullman (2011). *Mining of Massive Datasets*. Cambridge University Press. URL: http://www.mmds.org/.

Rani, Shama and Jaiteg Singh (2018). "Enhancing Levenshtein's Edit Distance Algorithm for Evaluating Document Similarity". In: *Computing, Analytics and Networks*. Ed. by Rajnish Sharma, Archana Mantri, and Sumeet Dua. Springer Singapore, pp. 72–80. URL: https://link.springer.com/chapter/10.1007/978-981-13-0755-3_6.

Rayson, Paul and Roger Garside (1998). "The Claws Web Tagger". In: *International Computer Archive of Modern English (ICAME) Journal* 22, pp. 121–123. URL: http://ucrel.lancs.ac.uk/people/paul/publications/rg98.pdf.

Romanello, Matteo, Aurélien Berra, and Alexandra Trachsel (2014). "Rethinking Text Reuse as Digital Classicists". In: *Digital Humanities Conference*. URL: http://dharchive.org/paper/DH2014/Panel-106.xml.

Rubenstein, Herbert and John B. Goodenough (Oct. 1965). "Contextual Correlates of Synonymy". In: *Commun. ACM* 8.10, pp. 627–633. ISSN: 0001-0782. URL: http://doi.acm.org/10.1145/365628.365657.

Ruder, Sebastian, Ivan Vulić, and Anders Søgaard (2019). "A Survey of Cross-Lingual Word Embedding Models". In: *Journal of Artificial Intelligence Research* 65, pp. 569–631. URL: https://www.jair.org/index.php/jair/article/view/11640.

Salton, Gerard, Anita Wong, and Chung-Shu Yang (1975). "A Vector Space Model for Automatic Indexing". In: *Communications of the ACM* 18.11, pp. 613–620. URL: https://dl.acm.org/doi/10.1145/361219.361220.

Sanchez-Perez, Miguel A., Grigori Sidorov, and Alexander Gelbukh (2014). "A Winning Approach to Text Alignment for Text Reuse Detection at PAN 2014 – Notebook for PAN at CLEF 2014". In: *Working Notes for CLEF 2014 Conference, Sheffield, UK, September*. URL: http://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-SanchezPerezEt2014.pdf.

Sanchez-Perez, Miguel A., Grigori Sidorov, and Alexander Gelbukh (2015). "Dynamically Adjustable Approach through Obfuscation Type Recognition – Notebook for PAN at CLEF 2015". In: *Working Notes for CLEF 2015 Conference, Toulouse, France*. URL: http://ceur-ws.org/Vol-1391/92-CR.pdf.

Sankoff, David and Joseph B. Kruskal (1983). "Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison". In: *Reading: Addison-Wesley Publication, 1983* 1.

Schleimer, Saul, Daniel S. Wilkerson, and Alex Aiken (2003). "Winnowing: Local Algorithms for Document Fingerprinting". In: *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*. ACM, pp. 76–85. URL: https://dl.acm.org/doi/10.1145/872757.872770.

Schütze, Hinrich and Jan O. Pedersen (1995). *Information Retrieval Based on Word Senses*.

Seo, Jangwon and W. Bruce Croft (2008). "Local Text Reuse Detection". In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 571–578. URL: https://dl.acm.org/doi/10.1145/1390334.1390432.

Smith, Temple F. and Michael S. Waterman (1981). "Identification of Common Molecular Subsequences". In: *Journal of Molecular Biology* 147.1, pp. 195–197. URL: https://dornsife.usc.edu/assets/sites/516/docs/papers/msw_papers/msw-042.pdf.

Somers, Harold L. (1999). "Review Article: Example-based Machine Translation". In: *Machine Translation* 14.2, pp. 113–157. URL: https://link.springer.com/article/10.1023/A:1008109312730.

Speer, Robert, Joshua Chin, and Catherine Havasi (2017). "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge." In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 4444–4451. URL: https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/download/14972/14051.

Stamatatos, Efstathios (2009). "Intrinsic Plagiarism Detection using Character n-gram Profiles". In: *Proceedings of the SEPLN Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse, PAN'09*. URL: http://ceur-ws.org/Vol-502/paper8.pdf.

Stamatatos, Efstathios (2011). "Plagiarism Detection using Stopword n-grams". In: *Journal of the American Society for Information Science and Technology* 62.12, pp. 2512–2527. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/asi.21630.

Stein, Benno, Nedim Lipka, and Peter Prettenhofer (2011). "Intrinsic Plagiarism Analysis". In: *Language Resources and Evaluation* 45.1, pp. 63–82. URL: https://link.springer.com/article/10.1007/s10579-010-9115-y.

Su, Zhan et al. (2008). "Plagiarism Detection using the Levenshtein Distance and Smith-Waterman Algorithm". In: *3rd International Conference on Innovative Computing Information and Control, 2008. ICICIC'08.* IEEE, p. 569. URL: https://ieeexplore.ieee.org/abstract/document/4603758.

Sun, Yu-Chih and Fang-Ying Yang (2015). "Uncovering Published Authors' Text-borrowing Practices: Paraphrasing Strategies, Sources, and Self-plagiarism". In: *Journal of English for Academic Purposes*. URL: https://www.sciencedirect.com/science/article/pii/S1475158515300035.

Tesnière, Lucien (1959). *Eléments de syntaxe structurale*. Librairie C. Klincksieck.

Tschuggnall, Michael and Günther Specht (2013). "Detecting Plagiarism in Text Documents through Grammar-Analysis of Authors". In: *Datenbanksysteme für Business, Technologie und Web (BTW), 15. Fachtagung des GI-Fachbereichs "Datenbanken*

und Informationssysteme" (DBIS), 11.-15.3.2013 in Magdeburg, Germany. Proceedings, pp. 241–259. URL: https://dl.gi.de/handle/20.500.12116/17324?locale-attribute=en.

Vania, Clara and Mirna Adriani (2010). "Automatic External Plagiarism Detection Using Passage Similarities - Lab Report for PAN at CLEF 2010". In: *CLEF 2010 LABs and Workshops, Notebook Papers, 22-23 September 2010, Padua, Italy*. URL: http://ceur-ws.org/Vol-1176/CLEF2010wn-PAN-VaniaEt2010.pdf.

Vila, Marta, M Antònia Martí, and Horacio Rodríguez (2010). "Paraphrase Concept and Typology. A Linguistically Based and Computationally Oriented Approach". In: *Procesamiento del Lenguaje Natural* 46, pp. 83–90. URL: https://pdfs.semanticscholar.org/6be5/c18d0919c1afdd3e9cb24f019fe2af7d2e14.pdf.

Vila, Marta, M. Antònia Marti Martí, Horacio Rodríguez, et al. (2014). "Is This a Paraphrase? What kind? Paraphrase Boundaries and Typology". In: *Open Journal of Modern Linguistics* 4.01, p. 205. URL: https://pdfs.semanticscholar.org/f229/65d1fcad60f4c7aec1988f4ac65b2e6f1bde.pdf.

Wagner, Robert A. and Michael J. Fischer (1974). "The String-to-String Correction Problem". In: *Journal of the ACM* 21.1, pp. 168–173. URL: https://dl.acm.org/doi/10.1145/321796.321811.

Weber-Wulff, Debora (2014). *False Feathers: A Perspective on Academic Plagiarism*. Springer Science & Business.

Wenyin, Liu, Anthony Y. Fu, and Xiaotie Deng (2008). "Exposing Homograph Obfuscation Intentions by Coloring Unicode Strings". In: *Asia-Pacific Web Conf.* Springer International Publishing, pp. 275–286. URL: https://link.springer.com/chapter/10.1007/978-3-540-78849-2_29.

Wise, Michael J. (1993). *Running Karp-Rabin Matching and Greedy String Tiling*. Tech. rep. Basser Department of Computer Science, University of Sydney. URL: https://www.semanticscholar.org/paper/Running-Karp-Rabin-Matching-and-Greedy-String-Wise/ab3d05fb967bcbcc70b71ebcfed8b24b5219aedc.

Wise, Michael J. (1995). "Neweyes: A System for Comparing Biological Sequences Using the Running Karp-Rabin Greedy String-Tiling Algorithm". In: *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology, Cambridge, United Kingdom*, pp. 393–401. URL: https://pdfs.semanticscholar.org/25a4/38dc2215f379cdf425a0897eb5d58a66f888.pdf.

Witten, Ian H, Alistair Moffat, and Timothy C Bell (1999). *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann.

Wu, Zhibiao and Martha Palmer (1994). "Verbs Semantics and Lexical Selection". In: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 133–138. URL: https://www.aclweb.org/anthology/P94-1019.pdf.

Yerra, Rajiv and Yiu-Kai Ng (2005). "A Sentence-Based Copy Detection Approach for Web Documents". English. In: *Fuzzy Systems and Knowledge Discovery*. Vol. 3613. Lecture Notes in Computer Science. Springer International Publishing, pp. 557–570. URL: https://link.springer.com/chapter/10.1007/11539506_70.

Zadeh, Lotfi A. (1965). "Fuzzy sets". In: *Information and Control* 8.3, pp. 338 –353. ISSN: 0019-9958.

Zechner, Mario et al. (2009). "External and Intrinsic Plagiarism Detection using Vector Space Models". In: *Proceedings of the SEPLN Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse, PAN'09*. URL: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-502/pan09-proceedings.pdf#page=55.

Zhang, Jiajun and Zong, Chengqing (2016). "Exploiting Source-side Monolingual Data in Neural Machine Translation". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1535–1545. URL: https://www.aclweb.org/anthology/D16-1160.

Zu Eissen, Sven Meyer and Benno Stein (2006). "Intrinsic Plagiarism Detection". In: *Advances in Information Retrieval*. Springer International Publishing, pp. 565–569. URL: https://link.springer.com/chapter/10.1007/11735106_66.