# Mono-lingual Paraphrased Text Reuse and Plagiarism Detection

**Rao Muhammad Adeel Nawab**

Supervisors: Dr. Mark Stevenson and Dr. Paul D. Clough

Department of Computer Science
University of Sheffield

A thesis submitted for the degree of
*Doctor of Philosophy*

12-09-2012

# Dedication

I would like to dedicate this thesis to my loving parents.

# Acknowledgements

Thanks to Allah who is the source of all the knowledge in this world, and imparts as much as He wishes to any one He finds suitable. I would like to thank my supervisors, Mark Stevenson and Paul D. Clough, who have supported me throughout my work on this dissertation with their patience and knowledge. Without their guidance, motivation and support this dissertation would not have been completed. One could not wish for more kind, accessible and friendlier supervisors than them.

I would also like to acknowledge the efforts of Jayne Woodward (my disability advisor) who assisted me during my research for these three years. My special thanks to Khurram Shahzad who helped me a lot throughout my work and gave very useful suggestions regarding it.

I offer my sincere gratitude to my beloved Sheikh (Hazrat Muhammad Abdul Qadir Owaise Shb DB) for his special attention and prayers. Without his support, motivation and guidance it would not have been possible to successfully complete this thesis.

I am thankful to my parents for their support, prayers, love and care throughout my life and they have played a vital role in achieving this milestone. My wife has always been a wonderful being to me and extended her whole-hearted support especially during my PhD studies, which I could not have completed without her. To my daughter Siham; you have played your part very well as your naughtiness always made me relaxed. I extend my thanks to my sisters, brothers, in-laws, relatives and friends for their continuous support and prayers.

Also many thanks to my elders and friends; Abdul Jabbar Khan, Dr. Saleem, Sheikh Muaz, Muhammad Umair, Tariq Najmi, Rashed Ahmed Khan, M. Safeer, Khan Rizwan, Raja Rizwan, Ghufran-ul-Haq, Ali Khawaja, Rizwan Iqbal, Shakeel Sultan, Qamar-uz-Zaman, Sheikh Hayat, M. Tasif, Sheikh Farhan, Usman Ghani Khan, Muzammil Shahbaz, Abdul Qadir, Ansar Mahmood.

Finally, I am thankful to COMSATS Institute of Information Technology, Lahore, Pakistan for funding this work under the Faculty Development Program.

Rao Muhammad Adeel Nawab
12 September, 2012.

# Abstract

Text reuse is the process of creating new documents using existing ones. Among the different types of text reuse, plagiarism (the unacknowledged reuse of text) is a widespread problem. Easy access to online information has made it easier to plagiarise and in recent years cases of plagiarism have increased. Consequently, plagiarism and its detection is receiving attention within the research community.

The main goal of this research is to develop algorithms for detecting mono-lingual text reuse, with a particular emphasis on mono-lingual extrinsic plagiarism detection. In this type of reuse both the source and reused texts are in the same language and the aim is to identify the source document that has been reused. Special attention is given to cases of text reuse created by paraphrasing the source document because detecting them is an open challenge.

This thesis focuses on two connected problems related to the detection of mono-lingual text reuse. The first is **candidate document selection**, the process of comparing a document against a collection to identify a small set of "candidate documents" which include the source document(s). The second problem is **pairwise document comparison**, the process of comparing a pair of documents to determine whether one has reused the other.

An IR-based framework is proposed for candidate document selection. To deal with cases of text reuse in which the source has been paraphrased, query expansion is incorporated into the IR-based framework. Different lexical resources for query expansion are explored. Evaluation is carried out using a variety of benchmark corpora and a new evaluation corpus created. Results showed that the proposed IR-based approach outperforms a state-of-the-art approach and is more robust in detecting verbatim (word to word copy) and modified copies of texts. Query expansion is found to be useful in detecting cases of reuse when the source document has been heavily paraphrased.

A system is also developed to carry out a pairwise comparison of documents. To deal with cases when the source text has been paraphrased, an n-gram overlap approach is extended with modified n-grams created by substituting words with synonyms and deleting words in an n-gram. Various lexical resources are used for substituting words with synonyms. A range of benchmark corpora are used for evaluation. Results showed that the modified n-gram approach improves performance when the source text has been paraphrased to create the reused text.

The thesis explores the problems of candidate document retrieval and pairwise document comparison for mono-lingual text reuse detection. It shows how techniques can be developed for these problems and that they can be extended to identify cases of text reuse when the source document has been paraphrased.

The following publications resulted from the research carried out for this thesis: [Nawab et al., 2010], [Nawab et al., 2011], [Nawab et al., 2012b] and [Nawab et al., 2012a].

# Declaration

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been acknowledged.

**Rao Muhammad Adeel Nawab**

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Text reuse is the process of creating new documents using the text from existing ones [Clough and Gaizauskas, 2009]. In some situations, it is a standard practice (e.g. text reuse in journalism), while in others it is unacceptable (e.g. plagiarism). The amount of text that is reused can vary from phrases, sentences, paragraphs to the entire document. In some situations, an entire document is reused to create a new document (e.g. Wikipedia[1] revisions). However, it is more likely that portions of text from a document will be reused.

Large scale electronic document collections (e.g. the internet) are now readily available, making it easy to reuse text and difficult to detect the source(s). Today, the process of reusing text (particularly for plagiarism) works as follows [Potthast, 2011]: Relevant (or source) article(s) are identified from the Web, then portion(s) of text are copied from the relevant article(s). Finally, the copied text is either rewritten or reused verbatim.

Plagiarism is a well-known type of text reuse. It is generally thought of as the unattributed reuse of a piece of text [Martin, 1994]. It is acknowledged as a significant problem in higher education and has been reported to be on the increase [Judge, 2008;

[1] http://www.wikipedia.org/

McCabe, 2005; Park, 2003]. For example, Sheard et al. [2002] reported a summary of three different surveys in which between 88% and 91.7% of students admitted that they were involved in cheating or academic dishonestly at least once. Consequently, plagiarism and its detection have recently received significant attention [Boisvert and Irwin, 2006; McCabe et al., 2006] and automated systems are now routinely used by higher education institutions to identify plagiarism in students' work.

## 1.1 Types of Plagiarism Detection

In recent years, the computational study of plagiarism detection has become a popular research area because it is difficult to manually analyse large electronic document collections and identify the source(s) of plagiarism. The problem of plagiarism detection is often divided into two tasks, both of which begin with a document suspected to contain plagiarism (the 'suspicious document') [Stein et al., 2007]: (1) intrinsic plagiarism detection - checking that the entire document (or all the passages) were written by one single author and (2) extrinsic plagiarism detection - searching for the source(s) (or original text(s)) that were reused to create the suspicious document.

In case of intrinsic plagiarism detection, the focus is on identifying portion(s) of text whose writing style significantly differs from the remaining text in the suspicious document, which means that the entire document is not written by one single author and contains text written by other author(s). Stylometric features are used to capture an author's writing style including average sentence or word length, function words, most frequent words, counting the use of punctuation, part of speech tag, spelling mistakes [Stamatatos, 2009].

Extrinsic plagiarism detection mainly involves comparison of the suspicious document with potential source documents. The task is complex because plagiarism can occur at different "levels" [Martin, 1994] including word-to-word plagiarism (phrases or passages are exactly copied without quotations and/or acknowledging the source(s)), paraphrasing plagiarism (words are modified but source can still be detected) and plagiarism of ideas (the idea of the original text is reused without dependence on the words or form of the source).

The task of extrinsic plagiarism detection can be further categorised into: (1) mono-lingual extrinsic plagiarism detection and (2) cross-lingual extrinsic plagiarism detection. In the former case, both the plagiarised and source texts are in the same language, while in the latter case, the plagiarised text is in one language and the source is in another. In cross-lingual plagiarism, the source text can be translated either automatically or manually. The translated text can be further modified or reused verbatim.

In extrinsic plagiarism detection, the suspicious document should ideally be exhaustively compared with all the available source documents to identify source(s) of plagiarised text. However, this is not practical in large document collections like the Web. To avoid this the suspicious document is compared with a small set of "candidate documents", which are assumed to contain all the sources of plagiarised text. The set of "candidate documents" should be carefully chosen from the document collection because any source document missed at this stage will not be identified in the later stages of processing.

## 1.2 Thesis Focus

The main aim of this thesis is to explore the problem of mono-lingual text reuse detection, with a particular focus on mono-lingual extrinsic plagiarism detection. Special attention is paid to the paraphrased cases of text reuse because they are hard to detect and an open challenge [Barrón-Cedeño, 2012; Maurer et al., 2006; Potthast et al., 2010b, 2011].

Plagiarists often try to disguise their behaviour by altering the text in some way (obfuscation), for example, by paraphrasing, summarising or inserting/deleting portion(s) of text [Campbell, 1990; Johns and Myers, 1990; Keck, 2006]. However, many previous approaches to plagiarism detection have been limited to the detection of verbatim copies of documents. Results showed that it is often straightforward to detect this type of plagiarism [Clough and Stevenson, 2011; Lane et al., 2006; Shivakumar and Garcia-Molina, 1995]. Previous studies have also shown that it is difficult to detect plagiarism when the original text has been paraphrased [Maurer et al., 2006; Potthast et al., 2010b, 2011]. For example, Maurer et al. [2006] paraphrased a passage with an Anti-Anti Plagiarism System[1] - a simple automatic tool for word replacement. The paraphrased passage was analysed by two well-known commercial plagiarism detection services and both failed to detect plagiarism. In addition, the best system [Kasprzak and Brandejs, 2010] in the 2nd International Competition on Plagiarism Detection [Potthast et al., 2010b] achieved a recall of more than 0.99 and precision of 0.95 when detecting verbatim (exact copy) plagiarism. However, none of the systems which took part in the competition achieved a recall of more than 0.28 for manually paraphrased (simulated) cases of plagiarism (the precision score varied). These results indicate that detecting plagiarism when the original text has been paraphrased is an open challenge.

In mono-lingual extrinsic plagiarism detection, the problem of detecting plagiarism created with paraphrasing has not been thoroughly investigated. Most approaches are limited to the detection of synonym replacement using WordNet (for example, see Ceska and Fox [2009]; Chen et al. [2010]) or carry out syntactic analysis to detect word reordering (for example, see Mozgovoy et al. [2007]; Uzuner et al. [2005]). The aim of this research is to develop algorithms which can identify the source(s) of text reuse (and plagiarism) particularly when the original text has been paraphrased.

This thesis explores two connected problems related to mono-lingual text reuse detection. The first is **candidate document selection**, the retrieval of potential

[1] `http://sourceforge.net/projects/aaps/` Last visited: 12-07-2012

source documents from large document collections. Candidate document selection has been shown to improve the speed and efficiency of text reuse detection systems [Barrón-Cedeño et al., 2009]. In addition, it could also be useful for semi-automatic approaches to text reuse detection by providing a human expert with a set of documents which is small enough to be manually analysed. To quickly reduce the search space for text reuse detection, an Information Retrieval (IR)-based framework is proposed (see Section 3.3). The second problem is **pairwise document comparison**, which aims to compare a pair of documents to determine whether one has reused the other. An exhaustive pairwise comparison of documents is useful in determining the amount of text reused to create the new text. In addition, it could be used to discriminate between different levels of text reuse. For this purpose, an n-gram overlap approach is implemented (see Section 5.2) and augmented with modified n-grams created by substituting words with synonyms and deleting words (see Section 5.3).

## 1.3 Research Goals

The main research goals of this thesis are as follows:

- Develop algorithms and techniques for mono-lingual text reuse detection with a particular emphasis on paraphrased text reuse.
- Develop techniques based on Information Retrieval to identify candidate source documents from large reference collections.
- Evaluate the effect of query expansion[1] for detecting text reuse when the original text has been paraphrased.
- Apply lexical resources to assist in the detection of similarity between documents.

[1] Query expansion is the process of modifying a query by adding related terms. For example, the original query, “car” can be expanded to “car vehicle automobile motorcar” to make an expanded query.

- Develop techniques to compare pairs of documents to identify text reuse particularly when the reused text has been paraphrased.

## 1.4 Contributions

The main contributions of this work are:

1. **Development of an IR-based framework for retrieving candidate documents from large document collections.**

   An IR-based framework is proposed to efficiently retrieve candidate documents from large source collections. The source collection is indexed and a suspicious document is split into queries which are used to retrieve a set of potential source documents. The top $N$ documents are selected for each query and the results of multiple queries merged using a score-based fusion approach (the CombSUM method [Fox and Shaw, 1994]) to generate a final ranked list of source documents. The top $K$ documents in the ranked list generated by CombSUM are marked as potential candidate documents.

2. **Incorporation of query expansion into the IR-based framework to deal with paraphrased cases of text reuse.**

   To detect text reuse when the original text has been altered by a high level of paraphrasing, query expansion is incorporated into the IR-based framework. Content words in the suspicious document are expanded with synonymous words from a thesaurus because lexical substitution is the most commonly used editing operation in mono-lingual paraphrasing and the reused text is a summarised version of the original one [Barrón-Cedeño, 2012].

3. **Exploration of lexical resources for text reuse detection.**

   Three lexical resources are explored for query expansion: (1) WordNet, a general-purpose thesaurus, (2) Paraphrase Lexicon, a corpus-derived thesaurus generated using an automatic paraphrase generation system [Callison-Burch, 2008] and (3) UMLS Metathesaurus, a thesaurus for processing biomedical text. To the best of our knowledge, the Paraphrase Lexicon and UMLS Metathesaurus have not been previously used for text reuse detection.

4. **Evaluation on a variety of benchmark corpora.**

   Four benchmark corpora are used to evaluate the performance of the proposed algorithms: (1) the PAN-PC-10 Corpus, which contains artificial (automatic) and simulated (manual) cases of plagiarism, (2) the MEDLINE Corpus, which is composed of potential cases of plagiarism from academic journal articles in the biomedical domain, (3) the METER Corpus, which contains real examples of text reuse (news stories) in journalism and (4) the Short Answer Corpus, which contains simulated examples of plagiarism as answers to five questions about Computer Science. Evaluation on a range of benchmark corpora with different properties provides a realistic picture of the performance of the proposed approaches.

5. **Development of a corpus for evaluating systems for the candidate document retrieval task.**

   The Short Answer Corpus [Clough and Stevenson, 2011] is too small to be used to evaluate the candidate document retrieval task. To make the task more challenging it is extended by adding documents from the Web. The new corpus is called the Extended Short Answer Corpus.

6. **Development of a system for pairwise document comparison.**

   A system for comparing pairs of documents to detect text reuse is developed. The approach is based on the widely used technique of n-gram comparison. To detect paraphrased text, standard n-grams are augmented with modified n-grams. The document which is suspected to contain reused text is used to create modified n-grams in two ways: (1) substitutions and (2) deletions. In the first case, words in an n-gram are substituted with synonymous words from WordNet, Paraphrase Lexicon and UMLS Metathesaurus to generate modified n-grams. In the second case, words in an n-gram are deleted to create modified n-grams. In addition, n-grams are weighted with probability scores obtained by training a language model.

## 1.5 Main Findings of this Research

The following observations are the main findings of this research:

**Observation 1:** An IR-based approach was found to be effective for the candidate document selection problem and outperformed an existing state-of-the-art approach based on Kullback-Leibler Divergence.

**Observation 2:** It is relatively straightforward to detect text reuse when the original text has been reused verbatim or slightly modified. However, it is more challenging to detect paraphrased cases of text reuse.

**Observation 3:** Knowledge-based query expansion can improve performance at detecting candidate documents containing paraphrased cases of text reuse.

**Observation 4:** Modified n-grams created by substituting and deleting words are useful in detecting text reuse when the original text has been paraphrased.

**Observation 5:** Assigning appropriate weights to n-grams using language model probability scores helps to improve performance.

## 1.6 Thesis Outline

The remainder of this thesis consists of the following five chapters:

**Chapter 2** Literature Review

This chapter describes state-of-the-art approaches for mono-lingual extrinsic plagiarism detection. Following that an overview of the International Competitions on Plagiarism Detection is presented. Four benchmark corpora which can be used to evaluate the performance of text reuse detection systems are described: (1) PAN-PC-10 Corpus (2) MEDLINE Corpus (3) METER Corpus and (4) Short Answer Corpus. These corpora contain artificial (automatically created), simulated (manually created) and real examples of text reuse. Finally, measures commonly used to evaluate the performance of plagiarism detection systems are discussed.

**Chapter 3** IR-Based Framework for Candidate Document Retrieval

This chapter presents an IR-based approach for the problem of candidate document retrieval. The IR-based approach is compared with a state-of-the-art approach, Kullback-Leibler Distance (see Section 2.7.1), which gave promising results in reducing the plagiarism detection search space [Barrón-Cedeño et al., 2009]. Evaluation is carried out using three corpora. Results showed that the IR-based approach outperformed the Kullback-Leibler Distance approach on all three corpora.

**Chapter 4** Improving the IR-based Approach with Query Expansion

This chapter begins with an overview of relevance feedback and query expansion. Following that lexical resources used for query expansion are described, which are: (1) WordNet, a general-purpose thesaurus, (2) Paraphrase Lexicon, a corpus-derived thesaurus and (3) UMLS Metathesaurus, a resource to assist in retrieving online literature related to health and biomedicine fields.

A limitation of the IR-based approach presented in the previous chapter is that it is based on exact matching. To deal with text reuse cases created with high levels of paraphrasing, query expansion is integrated into the IR-based framework. Documents that are suspected to contain reused text are expanded using synonymous terms from a lexical resource. Evaluation is carried out using the same three corpora which were used to evaluate the IR-based approach (presented in the previous chapter). Results demonstrated that integrating query expansion into the IR-based approach improves candidate retrieval performance compared to no query expansion.

**Chapter 5** Pairwise Document Comparison using Modified and Weighted N-grams

This chapter presents the modified n-gram approach, which makes an exhaustive pairwise comparison of documents to determine whether one document has reused the other. Using this approach, modified n-grams are generated by substituting and deleting words. N-grams are also weighted using probability scores obtained by training a language model. The problem of determining whether one document has reused the other is cast as a supervised document classification task.

The modified n-gram approach is evaluated using three corpora: (1) METER Corpus, (2) Short Answer Corpus and (3) MEDLINE Corpus. Results showed that using modified n-grams with substitutions and deletions methods improves performance. Further improvement is observed when appropriate weights are assigned to n-grams based on language model probability scores.

**Chapter 6** Conclusions

Presents a summary of the contributions made in this research work and discusses avenues for future work.

## 1.7 Published Work

Publications produced during this research work are as follows:

1. **R.M.A. Nawab**, M. Stevenson and P. Clough (2012) *Detecting Text Reuse with*

*Modified and Weighted N-grams.* *SEM: The First Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics, Montreal, Canada.

2. **R.M.A. Nawab**, M. Stevenson and P. Clough (2012) *Retrieving Candidate Plagiarised Documents using Query Expansion.* In Proceedings of the 34th European Conference on Information Retrieval (ECIR), Barcelona, Spain.

3. **R.M.A. Nawab**, M. Stevenson and P. Clough (2011) *Extrinsic 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, Amsterdam, Netherlands.

4. **R.M.A. Nawab**, M. Stevenson and P. Clough (2010) *University of Sheffield*, Lab Report for PAN at CLEF 2010. In Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse, Padua, Italy.

# Chapter 2

# Literature Review

## 2.1 Introduction

In previous chapter, Section 1.1 described two main types of plagiarism detection: (1) extrinsic plagiarism detection and (2) intrinsic plagiarism detection. The task of extrinsic plagiarism detection can be further categorised into mono-lingual extrinsic plagiarism detection and cross-lingual extrinsic plagiarism. An in-depth discussion of the methods proposed for these different types of plagiarism will be beyond the scope of this chapter. Therefore, this survey is restricted to the mono-lingual extrinsic plagiarism detection in natural language text which is the focus of this research work.

The rest of this chapter is divided into four parts. The first part describes state-of-the-art approaches for mono-lingual extrinsic plagiarism detection (Sections 2.3 to 2.9). The second part gives an overview of the extrinsic plagiarism detection methods used by various systems which participated in the International Competitions on Plagiarism Detection [Potthast et al., 2010b, 2011; Stein et al., 2009] (Section 2.11). The third part presents four benchmark corpora which can be used to evaluate the performance of extrinsic plagiarism detection systems (see Section 2.12). Finally, measures commonly used to evaluate the performance of plagiarism detection systems are presented

(Section 2.13).

## 2.2 Classifying Approaches to Mono-lingual Extrinsic Plagiarism

Plagiarism detection is a vast field and over the years several different methods have been proposed to automatically detect extrinsic plagiarism. The main intention of this survey is to give an overview of the methods that have proven to be effective and considered as state-of-the-art. The secondary intention is the reader should clearly understand the main ideas underlying each method.

Table 2.1 shows two recent surveys[1] that classified state-of-the-art approaches for detecting and measuring mono-lingual text reuse [Clough and Gaizauskas, 2009] and mono-lingual extrinsic plagiarism [Alzahrani et al., 2011]. These surveys give an insight into the state-of-the-art methods, which have proven to be effective in detection text reuse and are widely used. Extrinsic plagiarism detection is a kind of text reuse detection, therefore, the methods proposed for text reuse detection can also be applied to the problem of extrinsic plagiarism detection. There is an overlap between the classifications suggested for text reuse and extrinsic plagiarism detection by these researchers. In addition, the methods for text reuse and extrinsic plagiarism detection they discuss are similar. For example, “lexical similarity” [Clough and Gaizauskas, 2009] and “vector based methods” [Alzahrani et al., 2011] define similar approaches, while “overlap of n-grams” [Clough and Gaizauskas, 2009] and “character based methods” [Alzahrani et al., 2011] also describe similar approaches.

The methods presented in these two surveys can be broadly combined to create a set of the state-of-the-art methods for mono-lingual extrinsic plagiarism detection: (1)

[1] There could be other surveys as well but to the best of our knowledge, only these two are the most recent ones and therefore selected for classifying approaches for mono-lingual extrinsic plagiarism detection.

| Name | Classification of Approaches |
|---|---|
| Clough and Gaizauskas [2009] | 1. Lexical similarity<br>2. Overlap of N-grams<br>3. String or Sequence Comparison<br>4. Sentence Alignment<br>5. Summarisation and Paraphrasing<br>6. Visual Methods |
| Alzahrani et al. [2011] | 1. Character Based Methods<br>2. Vector Based Methods<br>3. Syntax Based Methods<br>4. Semantic Based Methods<br>5. Fuzzy Based Methods<br>6. Structural Based Methods |

Table 2.1: Classification of methods by different researchers for automatic detection of text reuse and extrinsic plagiarism.

lexical similarity, (2) overlap of n-grams, (3) fingerprinting, (4) string and sequence comparison, (5) probabilistic methods, (6) NLP (Natural Language Processing) methods, which can be further sub-categorised as: (i) syntactic methods and (ii) semantic methods and (7) structural methods. The following sections describe each method in detail.

## 2.3 Lexical Similarity

In Information Retrieval (IR), the most common retrieval task is *ad hoc* retrieval. In this type of retrieval, most IR systems index a static collection of documents ($D$) using an inverted index (which represents the content of a document $d \in D$ as a set of index terms). A user represents his information need as a query, which is used by the IR system to retrieve a set of relevant documents (which can be either ranked or non-ranked). The degree of similarity between a query and a document is computed using an IR model, which often involves computing the overlap of query terms and indexed terms. Therefore, an IR system tries to calculate *lexical* similarity [Baeza-Yates and Ribeiro-Neto, 2011].

The process of identifying plagiarised documents can be viewed as an Information Retrieval problem. When the problem is viewed this way the query is formed from the suspicious document. Document(s) in the reference collection from which the suspicious document is plagiarised are treated as relevant while all others are irrelevant.

Various models have been proposed for *ad hoc* retrieval including the widely used vector space model. This model has been widely employed for the development of plagiarism detection systems [Potthast et al., 2010b, 2011; Stein et al., 2009]. The subsections below describe this model and one of its popular variant (the relative frequency model) in more detail (Baeza-Yates and Ribeiro-Neto [2011] and Manning et al. [2008] present an in-depth and detailed overview of IR and various IR models).

### 2.3.1 Vector Space Model

In the vector space model [Salton et al., 1975] both the query and documents are represented as vectors in a high dimensional vector space. Each term (word or phrase) in the document collection corresponds to a dimension in the vector space. Documents that are close to the query in the vector space are retrieved. The closeness (or similarity) between a document vector $\overrightarrow{d}$ and query vector $\overrightarrow{q}$ is measured by computing the angle between them, called the cosine similarity measure, and is calculated as:

$$sim(d,q) = \frac{\overrightarrow{q} \bullet \overrightarrow{d}}{|\overrightarrow{q}| \times |\overrightarrow{d}|} = \frac{\sum_{i=1}^{n} q_i \times d_i}{\sqrt{\sum_{i=1}^{n}(q_i)^2 \times \sum_{i=1}^{n}(d_i)^2}} \tag{2.1}$$

where $|\overrightarrow{q}|$ and $|\overrightarrow{d}|$ represent the lengths of the query and document vectors respectively. This model allows for *partial matching* which enables to better estimate the similarity between a query and document.

The retrieved documents are ranked in descending order based on their degree of similarity with the query. Document(s) on top of the ranked list are more likely to be the source(s) of plagiarism.

Some terms in a document collection are more useful for identifying relevant documents than others. A number of weighting schemes, mostly based on frequency distribution of terms, have been proposed to assign weights to terms based on their relative importance in the document collection. For example, *term frequency (tf)*, defines the importance of a term based on its frequency (or occurrence) within a document; the higher the frequency of a term in a document the more relevant it is. In a document $d$, the *tf* of the $i$-th term ($t_i$) is computed as [Baeza-Yates and Ribeiro-Neto, 2011]:

$$tf_{i,d} = \frac{n_{i,d}}{\sum_k n_{k,d}} \tag{2.2}$$

where $n_{i,d}$ represents the occurrence (or frequency) of $t_i$ in document $d$ and it is normalised by the sum of frequencies of all the terms in $d$.

*Document frequency (df)* identifies a term's importance within a collection; a term appearing in a large number of documents is likely to be less important than one appearing in a few documents. *Inverse document frequency (idf)* gives more importance to a term appearing in fewer documents than the one appearing in more documents. The $idf_i$ of the term $t_i$ can be calculated as [Baeza-Yates and Ribeiro-Neto, 2011]:

$$idf_i = log\frac{|D|}{|D_i|} \tag{2.3}$$

where $|D|$ represents the total number of documents in the collection and $D_i$ represents the number of documents which contain the term $t_i$.

Term frequency (*tf*) and inverse document frequency (*idf*) have been combined to form the popular and widely used *tf.idf* weighting scheme (see Equation 2.4) [Baeza-Yates and Ribeiro-Neto, 2011]. Variants of the *tf.idf* weighting scheme have also been proposed.

$$tfidf_{i,d} = tf_{i,d} \cdot idf_i = \frac{n_{i,d}}{\sum_k n_{k,d}} \cdot log\frac{|D|}{|D_i|} \tag{2.4}$$

The vector space model (and its variants) has been used to develop plagiarism detection systems. The majority of the systems presented in the 1st [Stein et al., 2009], 2nd [Potthast et al., 2010b] and 3rd [Potthast et al., 2011] International Competitions on Plagiarism Detection applied this model (with and without *fingerprinting*; see Section 2.5 for description of fingerprinting approach), particularly for the candidate document retrieval task. Using this approach, the entire source collection is converted to fixed length character- or word-n-grams (called chunks) and indexed. Chunks are weighted using different weighting schemes including boolean weights, frequency weights and *tf.idf* weights. Each suspicious document is split into chunks of the same length as that of the source documents. Each chunk is queried in the index and the source documents for which the number of common chunks with the suspicious document are above some pre-defined threshold are marked as potential candidate documents. Different similarity measures are used for computing similarity between suspicious-source document pairs including cosine similarity measure (see Equation 2.1) and Jaccard similarity (see Equation 2.8).

Lewis et al. [2006] proposed a vector-based text similarity search algorithm (called eTBLAST) to identify highly similar citation pairs (potential cases of plagiarism) in MEDLINE[1] (see Section 2.12.2). A query is formed from the title and abstract of a MEDLINE citation (stop words are removed and remaining keywords are weighted using a term weighting scheme). eTBLAST computes the similarity score between title and abstract query and MEDLINE citations and returns a list of highly similar citations ranked by their similarity scores. The top 400 citations returned by eTBLAST are re-ranked using a sentence-alignment algorithm to generate a final ranked list of highly similar citations.

The vector space model has also been used to detect text reuse on the web [Bendersky and Croft, 2009], detect and measure text reuse in journalism [Clough, 2003b],

[1] http://www.ncbi.nlm.nih.gov/pubmed/ Last visited: 10-08-2012

detect duplicate and near-duplicate documents [Hoad and Zobel, 2003] and detect duplicate defect reports [Runeson et al., 2007].

### 2.3.2 Relative Frequency Model

A variant of the vector space model is the relative frequency model. The SCAM (Stanford Copy Analysis Mechanism) [Shivakumar and Garcia-Molina, 1995] and COPS (COpy Protection System) [Brin et al., 1995] plagiarism detection systems were developed using this model. The main difference between these two systems is that the SCAM system compares documents at word-level and the COPS system at sentence-level.

The main idea underlying this model is to compare words (or chunks) with similar frequencies. The similarity score is not effected by the sizes of documents i.e. one document can be a subset or superset of the other. Instead of the entire set of words (or chunks), a subset of words (or chunks) which have similar frequencies in the registered (or source) $R$ and suspicious $T$ documents are selected. For a word (or chunk) $w_i$ to be included in the set $C$, it should satisfy the following condition:

$$\epsilon - (\frac{f_i(R)}{f_i(T)} + \frac{f_i(T)}{f_i(R)}) > 0 \tag{2.5}$$

where $\epsilon$ is a constant, $f_i(R)$ and $f_i(T)$ are the frequencies of $w_i$ in documents R and T respectively. The choice of $\epsilon$ value is important because it determines whether a word (or chunk) should be included in the set $C$ or not. A large value will increase the size of $C$ and result in too many false positives, whereas a small value will decrease the size of $C$ and result in too few matches. The best value reported by Shivakumar and Garcia-Molina [1995] was $\epsilon = 2.5$. To compute the similarity (or overlap) score between $R$ and $T$ an asymmetric subset measure $S(T, R)$ is defined as:

$$S(T, R) = \frac{\sum_{w_i \in C} f_i(R) \times f_i(T)}{\sum_{i=1}^{N} f_i(T) \times f_i(T)} \tag{2.6}$$

The subset measure is normalised by the suspicious document alone, using the frequencies of the words appearing in the set $C$. It returns a high similarity score when $T$ is a subset of $R$. The final similarity measure is computed as:

$$similarity(T, R) = max[S(T, R), S(R, T)] \tag{2.7}$$

or maximum of the two asymmetric subset measures. The value of the similarity score is between 0 to 1.

## 2.4 Overlap of N-grams

### 2.4.1 N-grams

An n-gram is an adjacent string of tokens (characters or words). A set of *overlapping* n-grams can be generated by representing a text as a string of tokens and moving a sliding window of one token at a time from the start to the end of the string. A string of $n$ tokens with a window of $m$ tokens will generate $(n-m)+1$ n-grams. For example, the set of word tri-grams for the string "*i ride a new car*" will be: {*i ride a, ride a new, a new car*}.

For a given input text, various parameters can be changed in the n-gram generation process including whether n-grams are overlapping or non-overlapping and fixed or variable length. In most applications, overlapping n-grams with fixed length are used. The choice of the n-gram size is important because a small n-gram size will result in too many matches, whereas a large size will result in too few matches. In plagiarism detection, different researchers have reported different optimal values for the n-gram length. For example, Lane et al. [2006] reported that word trigrams are the best comparison unit. Barrón-Cedeño et al. [2009] showed that both word bigrams and trigrams are suitable and Ceska [2009] found word 4-grams to be the most useful. Errami et al. [2010] successfully applied word 6-grams to identify highly similar citation pairs in MEDLINE[1] (see Section 2.12.2). The word 6-grams were weighted using the language model probability scores to give more importance to rare 6-grams and less to frequent ones. Bigram probabilities were used to compute the probabilities of 6-grams. The similarity score between a citation pair was computed by summing the probability scores of the common 6-grams in the two citations divided by the sum of the probability scores of the 6-grams of the smaller citation (overlap similarity coefficient; see Equation 2.10).

N-grams have been effectively used in applications related to text reuse including identifying plagiarised documents [Barrón-Cedeño et al., 2009; Clough and Stevenson, 2011; Lane et al., 2006], copy detection [Shivakumar and Garcia-Molina, 1995], measuring text reuse in journalism [Clough et al., 2002] and on the web [Chiu et al., 2010]. In addition, they have proved to be useful in other NLP applications including building statistical *language models* [Jurafsky and Martin, 2008], spelling error detection and correction [Bergsma et al., 2009], automatic evaluation of Machine Translation [Papineni et al., 2002] and summarisation systems [Lin, 2004].

### 2.4.2 Similarity Measures for Computing Overlap of N-grams

Since plagiarised documents are likely to share more n-grams then non-plagiarised ones, the n-gram overlap similarity score can be used to distinguish plagiarised documents from non-plagiarised ones. The underlying assumption is that if the similarity score between a suspicious-source document pair is higher then certain threshold, the source document was reused to create the plagiarised one.

[1] http://www.ncbi.nlm.nih.gov/pubmed/ Last visited: 10-08-2012

A number of measures have been proposed to quantify the degree of overlap between two sets of n-grams [Broder, 1997; Manning and Schütze, 1999]. Generally, the similarity score is computed by counting the number of common n-grams normalised by the length of one or both set(s) of n-grams. A similarity measure either falls into the category of asymmetric similarity measure or symmetric similarity measure. In the former case, the length of only one of the sets is employed in the normalisation process whereas in the latter case, normalisation is carried out using the lengths of both sets. Four widely employed similarity measures are discussed below (a detailed discussion of all the similarity measures will be beyond the scope of this section).

### Jaccard

The Jaccard similarity co-efficient is a symmetric measure based on set theoretic principles. This measure treats the document pair to be compared as sets of n-grams. If $S\left(A,n\right)$ and $S\left(B,n\right)$ are the sets of n-grams of length $n$ in documents A and B respectively then the Jaccard similarity co-efficient is given by:

$$S_{jaccard}(A,B) = \frac{|S(A,n)\bigcap S(B,n)|}{|S(A,n)\bigcup S(B,n)|} \tag{2.8}$$

The value of $S_{jaccard}(A,B)$ ranges between 0 and 1, where 0 means no match and 1 means a perfect match.

### Dice

Dice is a variant of the Jaccard similarity co-efficient and is also a symmetric measure. If S(A,n) is the set of n-grams of length $n$ in document A and S(B,n) is the set of n-grams of length $n$ in document B then the Dice similarity co-efficient is given by:

$$S_{dice}(A,B) = 2 \times \frac{|S(A,n)\bigcap S(B,n)|}{|S(A,n)| + |S(B,n)|} \tag{2.9}$$

The value of similarity score is between 0 and 1, where 0 means that two documents are entirely different and 1 means that they are exactly the same.

### Overlap

Overlap similarity co-efficient is an asymmetric variant of Jaccard. If the sets of n-grams of length $n$ in documents A and B are represented by S(A,n) and S(B,n) respectively, then the similarity between them with this measure is calculated as:

$$S_{overlap}(A,B) = \frac{|S(A,n) \bigcap S(B,n)|}{min(|S(A,n)|, |S(B,n)|)} \tag{2.10}$$

The domain of similarity score is $[0, 1]$, where 0 means no overlap and 1 means perfect overlap.

### Containment

Broder [1997] proposed the asymmetric containment measure to quantify the degree of text within a document $(A)$ that has been reused in another document $(B)$. If $S\,(A, n)$ and $S\,(B, n)$ are the sets of n-grams of length $n$ in documents $A$ and $B$ respectively then the containment similarity measure is given by:

$$S_{containment}(A,B) = \frac{|S(A,n) \bigcap S(B,n)|}{|S(A,n)|} \tag{2.11}$$

A similarity score of 1 means that document $A$ is "roughly contained" in document $B$ and a score of 0 means that none of the n-grams in $A$ occur in $B$.

## 2.4.3 N-gram Comparison for Evaluation

Automatic evaluation systems have been developed based on n-gram comparison approach, for example, BLEU [Papineni et al., 2002] and ROUGE [Lin, 2004]. BLEU is commonly used to automatically evaluate the performance of Machine Translation (MT) systems and ROUGE is widely used to automatically evaluate the quality of a summary. To evaluate the quality of a candidate translation/summary, both BLEU and ROUGE compute the number of overlapping n-grams between the candidate translation/summary and reference translations/summaries.

To better evaluate the quality of a candidate translation/summary, both BLEU and ROUGE use similar n-gram clipping methods to clip the count of n-grams. N-gram clipping can be useful in computing the n-gram overlap similarity score for plagiarism detection (see Section 5.3.3). The n-gram overlap approach used by BLEU is described below.

BLEU modifies the standard precision measure (see Section 2.13) because the automatic translation ('candidate') produced by an MT system is likely to generate more words than in the reference translation(s). Consider the following example taken from Papineni et al. [2002]:

| | |
|---|---|
| Candidate | the the the the the the the |
| Reference 1 | the cat is on the mat |
| Reference 2 | there is a cat on the mat |

All the seven word unigrams in the candidate translation appear in the reference translations, so the unigram precision of the candidate translation will be:

$$precision = \frac{m}{m_t} = \frac{7}{7} = 1$$

where $m$ is the number of word unigrams common in the candidate and reference translations and $m_t$ is the total number of word unigrams in the candidate translation. Although candidate translation does not contain the same amount of content as in the reference translations still it has a perfect score of 1. To avoid this, BLEU modifies the calculation of the precision score by clipping the count of each word $m_w$ in the candidate translation. For each $m_w$, the total count is clipped to its maximum total count $m_{max}$ in any of the reference translations. In the above example, the word unigram "the" appears twice and once in the "Reference 1" and "Reference 2" translations respectively. Thus, the count of the word "the" will be clipped to the maximum total count in the "Reference 1" translation, which means $m_w = 7$ and $m_{max} = 2$. The $m_w$ for all the word unigrams in the candidate translation will be summed after clipping and divided by the total number of word unigrams in the candidate translation. In the above example, the modified unigram precision score will be:

$precision = \frac{2}{7} = 0.28$

This n-gram clipping approach can be used for any length of n-grams.

### 2.4.4 Sentence-level Comparison

Instead of using fixed length n-grams, some studies have focused on sentence-level comparison (which can be treated as variable length n-grams).

White and Joy [2004] proposed a sentence-based method for plagiarism detection in student assignments both for natural language text and source code. Their method represents each documents as a set of sentence(s). Each suspicious sentence is compared with all the sentences of all the documents in the collection. If the number of common words between two sentences is more than certain threshold they are marked for further analysis. Similarity at sentence-level is combined to compute the document-level similarity for plagiarism detection.

Gustafson et al. [2008] proposed a plagiarism detection approach based on word similarity at sentence-level. The proposed system uses word correlation factors extracted from 880,000 Wikipedia[1] articles by (1) counting the frequency of co-occurrence and (2) finding the relative distance of each Wikipedia article, to identify similar/related words between two sentences (thus allowing for partial word matching as compared to exact). Similarity between a pair of sentences is computed by counting the number of common words in them. Sentence-level similarity is used to calculate the similarity at document-level.

## 2.5 Fingerprinting

Another popular approach for plagiarism detection is fingerprinting. Using this approach, the content of a document is represented as a set of *fingerprints*. A *fingerprint*

[1] http://www.wikipedia.org/

is a unique integer, which is generated by first selecting a substring/subsequence from a document and then applying a *hash function* to it.

In the fingerprinting approach, a set of substrings is selected to represent the overall content of the document. These substrings are passed to a *hash function*, for example, *MD5* [Rivest, 1992], which transforms substrings to fingerprints. The set of fingerprints generated by applying *hash function* on all the substrings represents the document fingerprint. Fingerprints of the entire source collection are generated and indexed. Fingerprints representing the suspicious document are generated in the same way. Each fingerprint of the suspicious document is queried against the index to find similar fingerprints. Generally, the similarity between a suspicious-source document pair is determined by the number of common fingerprints normalised by the length of one or both documents (see Section 2.4.2). A number of variants of the basic fingerprinting approach have been proposed. Potthast [2011] recently provided a comprehensive overview of the fingerprinting approaches for plagiarism detection in large data collections.

While designing a fingerprinting approach, four main issues should be considered [Hoad and Zobel, 2003]. The first is fingerprint generation, i.e. how a substring should be transformed into a fingerprint. The second is fingerprint granularity, the size of substring used to generate a fingerprint. The third is fingerprint resolution, i.e. the number of fingerprints (all or a subset) used to represent a document. Finally the substring selection strategy, i.e. which substrings in a document should be selected and passed to the *hash function* to generate their fingerprints.

The basic fingerprinting approach and its variants have been used for plagiarism detection [Grman and Ravas, 2011; Kasprzak and Brandejs, 2010; Lyon et al., 2001; Scherbinin and Butakov, 2009], illegal copy detection [Brin et al., 1995; Shivakumar and Garcia-Molina, 1995], text-based Information Retrieval for near similarity search in large document collections [Stein, 2005], duplicate or near-duplicate documents detection in large collections [Hoad and Zobel, 2003] and text reuse detection at sentence/passage level [Seo and Croft, 2008].

The fingerprinting approach is similar to the n-gram overlap approach (see Section 2.4). The main difference is that input chunks are run through *hash function* first. In addition, the vector space model (see Section 2.3.1) can be used as a fingerprint retrieval model instead of using chunks without hashing.

The main advantage of the fingerprinting approach is that it is fast and can be effectively applied to large document collections. The disadvantages of this approach are: (1) the process of fingerprint generation is affected by the change of even a single character and (2) its time and space requirements are high since fingerprints have to be generated from substrings and then stored in an index.

## 2.6 String and Sequence Comparison

In plagiarism detection, an original text is often altered (or obfuscated) to hide plagiarism. The process of plagiarism detection can be considered as a sequence alignment problem, which aims to identify similar fragments of text between suspicious-source document pair (text in a document is represented as sequence of tokens). Consequently, the algorithms proposed for aligning biological sequences can also be used for plagiarism detection. Note that this section gives an overview of the sequence comparison approaches in context of the biological sequence alignment.

Similar to the obfuscation in plagiarism, during the evolution process a gene normally goes through different mutations (or changes). The most common mutations include insertions or deletions (also called *indels*) of a single element or subsequence and modifications in a single element. Biologists are mainly interested in identifying the evolution changes in genomes over time. For this purpose, DNA, RNA or protein sequences are aligned to acquire information about genes that have similar function, structure and evolutionary process [Mount, 2004].

### 2.6.1 Sequence Alignment

Mount [2004] defines biological sequence alignment as "*the procedure of comparing two (pair-wise alignment) or more (multiple sequence alignment) sequences by searching for a series of individual characters or character patterns that are in the same order in the sequences*". The two main approaches for sequence alignment are: (1) global alignment and (2) local alignment. These approaches align two sequences using a popular method called *dynamic programming*, which guarantees an optimal (or best) alignment given a particular scoring function.

In global alignment, the alignment process is extended to the entire lengths of the two sequences. This type of alignment is suitable for sequences that are similar and of almost equal size. Needleman-Wunsch is a general global alignment algorithm to align biological sequences [Needleman and Wunsch, 1970].

It is likely that two sequences which look dissimilar will have small common regions of similarity. Local alignment is used to identify the similar portions (or subsequences) of text between two sequences. Local alignment is most suitable for divergent sequences that contain only small similar subsequences. Smith-Waterman is a general local alignment algorithm [Smith and Waterman, 1981]. For sequences "ABABCDEDDCFCF" and "ABCDDDCCF", possible global and local alignments are:

Global Alignment

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | A | B | C | D | E | D | D | C | F | C | F |
| A | - | - | B | C | D | - | D | D | C | - | C | F |

Local Alignment

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | A | B | C | D | E | D | D | - | C | F | C | F |
| - | - | A | B | C | D | - | D | D | C | C | F | - | - |

Algorithms based on *dynamic programming* guarantee optimal alignment but they are slow for aligning long sequences and searching large databases. To speed up the alignment process, a number of *heuristic* methods based on approximate optimal alignment have been proposed including FASTA [Pearson, 1990], BLAST [Altschul et al., 1990] and Gapped BLAST [Altschul et al., 1997]. These methods are very fast but do not guarantee optimal alignment.

### 2.6.2 Edit Operations

In the alignment process, text is represented as a sequence of tokens. The granularity of a token can vary from a single character to entire sentences. A popular approach to compare (or align) two sequences is to compute the similarity between them by counting the number of editing operations required to convert one sequence into the other. Sequence comparison approaches are *order preserving*, i.e. the order of the tokens in the sequences being compared is important [Clough and Gaizauskas, 2009].

Sankoff and Kruskal [1983] suggested four *edit operations* for transforming one sequence into another: (1) insertions and deletions (or indels), (2) substitutions, (3) compressions and expansions and (4) transpositions (or swaps). These edit operations can be used to calculate the difference score between two sequences. In sequence alignment, a *gap* occurs when a token is inserted or deleted (denoted by '-'). Consider the following edit operations for a single character token.

(x,x): a match

(x,-): the deletion of x

(-,y): the insertion of y

(x,y): the replace of x by y (for x≠y)

For two input sequences, a number of different alignments can be obtained. Consider the following possible alignments between the sequences $SeqX$ and $SeqY$:

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SeqX | A | C | C | G | T | - | A | - | C | C | G | T |
| SeqY | A | C | C | G | T | T | A | C | C | G | T | T |

One obvious question is, which is the best (or optimal) alignment among all possible alignments. To identify the optimal alignment, the sequence similarity score is quantified by a *scoring function* which assigns different costs (or weights) to different edit operations. The *scoring function* is defined by the number of matches, number of mismatches (substitutions) and number of gaps (insertions and deletions). The overall score of an alignment is computed by adding the scores of all the individual operations: matches, mismatches and gaps. For example, consider the following scoring function:

w(x,x) = 0

w(x,-) = w(-,y) = 1

w(x,y) = 1 (for x≠y)

The distance score with this scoring function for the two alignments of the sequences *SeqX* and *SeqY* will be 1 and 3 respectively.

### 2.6.3 Edit Distance and Longest Common Subsequence

When the minimum cost is computed to convert one sequence into the other, it produces the well-known *edit distance* method. One constraint in this method is that sequence transformation should be carried out using only two edit operations: (1) insertions and deletions (or indels) and (2) substitutions. Moreover, when the cost of *indels* = 1 and *substitutions* = 2, then it gives rise to the widely used *Levenshtein distance* [Levenshteiti, 1966]. In addition, if insertions or deletions are the only edit operations used, this produces the *longest common subsequence* (lcs) approach. Given a string, a subsequence is a string of consecutive tokens obtained by deleting zero or more tokens from that string. Given two strings to be compared, $X$ and $Y$, the *lcs* is the longest subsequences common between them. For example, if $X =$"abcdef" and $Y =$"abgdef", *def* is a subsequence and *abdef* is the longest common subsequence.

A limitation of the *lcs* approach is that it fails to identify optimal similarity when groups of contiguous tokens are rearranged to create a new string (known as the *block move* problem [Wise, 1993]). Consider the following example:

| | |
|---|---|
| **Seq1** | a big dog bit the postman |
| **Seq2** | the postman was bitten by a big dog |

In this case, the *lcs* is "*a big dog*". However, there is other common information between these two strings ("*the postman*"), which is missed by *lcs* due to the *block move* problem.

### 2.6.4 Greedy String Tiling

In plagiarism detection, the problem of *block move* was first addressed by Wise [1993, 1995]. He proposed the Greedy String Tiling (GST) algorithm, which can efficiently detect *block moves*. The algorithm has a run time of $O(n^3)$, but has been optimised to run in linear time using a string matching algorithm Running Karp-Rabin Greedy String Tiling (RKR-GST) [Wise, 1993].

RKR-GST makes use of *tiles*, consecutive subsequences of maximal length that occur as one-to-one pairing between two input strings. RKR-GST aligns two input strings (*text* and *pattern*) such that as much as possible of *pattern* is covered by tiles shared with *text*. Once a token has been used in a tile it is marked and cannot be used again. A *minimum match length* parameter is set to avoid accidental matches by imposing a minimum length on tiles. RKR-GST works through *text* and *pattern* by applying multiple passes of a two stage process: scanpattern and markarrays [Wise, 1993].

- **Stage 1 (Scanpattern)**: The longest matching substrings are found in this stage. Tokens of *pattern* string are compared with tokens in the *text* string that are not marked. If a match is found then it is extended until the end of match or string is reached. This stage generates all the matches of maximal length.

- **Stage 2 (Markarrays)**: This stage stores the tiles collected in previous stage and marks the tokens that are used in the tiles so that they may not be used again in the next pass.

When all maximal matches are found or the minimum match length is reached, the algorithm terminates. The algorithm is greedy in 'selection' and 'matching'. In selection, if more than one token is available, it will select the first occurrence. Longer matches are preferred to shorter ones.

Approaches based on sequence alignment have been used for plagiarism detection. Su et al. [2008] used the Levenshtein distance and simplified Smith-Waterman local alignment algorithm to detect plagiarism. Irving [2004] also proposed a variant of the Smith-Waterman local alignment algorithm to identify plagiarism in students assignments. Bagdis [2008] adapted the heuristic-based local alignment algorithm (BLAST [Altschul et al., 1990]) for identifying plagiarism in free text. Burrows et al. [2004] applied variants of local alignment algorithm for detecting source code plagiarism in large code repositories. Clough and Stevenson [2011] used longest common subsequence for plagiarism detection in free text. Elhadi and Al-Tobi [2009] combined the longest common subsequence method with syntactical features extracted using part-of-speech tagging to identify duplicate documents.

Greedy String Tiling has been used for plagiarism detection in free text [Nawab et al., 2010, 2011] and program code [Wise, 1996], biological sequence alignment [Wise, 1993, 1995] and measuring text re-use in journalism [Clough et al., 2002].

## 2.7 Probabilistic Methods

Probabilistic methods have been successfully used for mono-lingual [Barrón-Cedeño et al., 2009] and cross-lingual [Barron-Cedeno et al., 2008][1] plagiarism detection. The subsection below describes the Kullback-Leibler Divergence method, which was adapted as a symmetric distance measure to efficiently reduce the search space for mono-lingual plagiarism detection.

### 2.7.1 Kullback-Leibler Distance

Kullback-Leibler Divergence ($KL_d$) or *relative entropy* [Kullback and Leibler, 1951] is defined as "*the average number of bits that are wasted by encoding events from a distribution p with a code based on a 'not-quite-right' distribution q*" [Manning and Schütze, 1999]. Given two probability distributions $P$ and $Q$ with mass functions $P(x)$ and $Q(x)$ over an event space $X$, the $KL_d$ calculates how different $P$ and $Q$ are as:

$$KL_d(P||Q) = \sum_{x \in X} P(x) log \frac{P(x)}{Q(x)} \tag{2.12}$$

Kullback-Leibler Divergence is a difference measure i.e. $KL_d(P||Q) = 0$ iff $P = Q$. Also it is an asymmetric measure i.e. $KL_d(P||Q) \neq KL_d(Q||P)$. Variants of $KL_d$ have been proposed which transform the original asymmetric measure into a symmetric measure. This approach has been used with promising results in a variety of applications including plagiarism detection [Barrón-Cedeño et al., 2009], document clustering [Pinto et al., 2007] and image retrieval [Do and Vetterli, 2000].

To retrieve candidate documents for plagiarism detection, Barrón-Cedeño et al. [2009] used a variant of the Kullback-Leibler Divergence called *Kullback-Leibler Symmetric Distance*, $KL_\delta$ (see Equation 2.13). The proposed method models a suspicious document and documents in the reference collection ($D$) as probability distributions

[1] Probabilistic model is based on statistical machine translation.

and compares them by computing $KL_\delta$. Documents are converted into probability distributions by first removing stop words and stemming [Porter, 1980], and then computing *tf.idf* weights for the remaining word unigrams. Assume $P_d$ and $Q_s$ are the probability distributions for a document in the reference collection, $d \in D$, and a suspicious document, $s$, respectively. The *Kullback-Leibler Symmetric Distance* between them (over a feature vector $X$) is computed as follows:

$$KL_\delta(P_d|Q_s) = \sum_{x \in X} (P_d(x) - Q_s(x)) log \frac{P_d(x)}{Q_s(x)} \tag{2.13}$$

Results from Barrón-Cedeño et al. [2009] showed that the overall accuracy and speed of the plagiarism detection system improved by applying the *Kullback-Leibler Symmetric Distance* for reducing the plagiarism detection search space. The system's performance without search space reduction was precision = 0.73, recall = 0.63 and $F_1$ = 0.68. Integrating the search space reduction step, performance improved to precision = 0.75, recall = 0.74 and $F_1$ = 0.75. The execution time also reduces from 2.32 seconds to 0.19 seconds.

## 2.8 NLP Techniques

The techniques for plagiarism detection discussed so far employ features based on the distribution of words, phrases, sentences etc. However, these features mostly compare suspicious-source document pairs at string level. To further improve the performance of plagiarism detection systems NLP techniques have been incorporated into existing approaches.

Clough [2003a] and Ceska and Fox [2009] highlighted the need of applying NLP techniques to plagiarism detection. The NLP techniques applied to plagiarism detection fall into two broad approaches: (1) syntactic and (2) semantic.

### 2.8.1 Syntactic Approaches

In linguistics, syntax is the study of the principles or rules which govern the process of combining words to make phrases and combining phrases to make grammatical sentences. A set of grammatical rules to describe a sentence is called a *grammar*. In syntactic analysis, a parser is used to identify the structure of a sentence by applying a grammar. The structure of a sentence is often displayed graphically using a *syntax tree* or represented by *labeled bracketing* [Jurafsky and Martin, 2008].

Syntactic features for plagiarism detection are often based on Part Of Speech (POS) tags and/or phrase structure information [Alzahrani et al., 2011]. A POS is a lexical category (or word class) which is assigned to a word. Words with different POS tags can be combined to make phrases. The most common lexical categories are: noun, verb, pronoun, preposition, adverb, conjunction, participle and article [Jurafsky and Martin, 2008].

Uzuner et al. [2005] proposed a plagiarism detection system that used *context free grammar* to extract syntactic features including sentence initial and final phrase structure, semantic verb classes, argument structures of verb phrases and syntactic classes of verb phrases. They called these syntactical features "syntactic elements of expression". Results showed that these syntactic features improve performance in detecting plagiarism created by paraphrasing.

Chong et al. [2010] used various pre-processing and NLP techniques to normalise documents and reported that they help to improve performance of extrinsic plagiarism detection systems. Documents were pre-processed by sentence segmentation, tokenisation, lowercase, stop-word removal, punctuation removal, part of speech tagging, stemming, lemmatization, number replacement, chunking and dependency parsing. The most promising results were obtained using dependency parse relations matching.

Mozgovoy et al. [2007] applied dependency parsing at sentence-level to normalise the effect of word reordering. The plagiarism detection system proposed by Mozgovoy et al. [2005] was used for experiments (this system applies suffix array data structure to identify matching substrings between document pairs). Results showed that integrating parsing into the plagiarism detection system improves performance.

### 2.8.2 Semantic Approaches

Semantic features are often based on identifying synonymous or related words. Lexical resources like WordNet (see Section 4.3.1) can help to identify semantic similarity for plagiarism detection. Alzahrani et al. [2011] suggest that plagiarism created by paraphrasing is likely to be detected by applying semantic approaches. They also argue that less attention has been paid to employing semantic approaches because of the problems related to algorithm complexity and semantic representation.

WordNet has been the most commonly used resource to aid in the detection of semantic similarity for plagiarism detection. Chen et al. [2010] used three methods provided by ROUGE [Lin, 2004] (see Section 2.4.3) for plagiarism detection including longest common subsequence, skip bigrams and n-gram co-occurrence statistics. WordNet was integrated into these approaches to identify synonym replacement. Two measures were used to identify relationships between a suspicious-source word unigram pair: (1) synonym-based measure and (2) relationship-based measure. In the first approach, the similarity score between two unigrams is computed by counting the number of common synonym words in the synsets of suspicious and source unigrams, normalised by the total number of unique synonym words in two synsets. Each synset of the suspicious unigram is computed with all the synsets of the source unigram. The highest similarity score between suspicious-source synsets is used to compute similarity at sentence level. The second approach is similar to the first one, except that the hierarchal information provided by the hypernyms/hyponyms relationships is used to compute the depth of a synset. The lowest depth score is used to compute similarity at sentence level.

In EuroWordNet, each synset/sense is mapped to a unique *Inter Lingual Index* (ILI). For synonym recognition (documents written in Czech), Ceska and Fox [2009] searched for a word in EuroWordNet and if a match was found its ILI was retrieved. An ILI was retrieved using (1) first sense (ILI of the first sense was returned), (2) sense selection after word sense disambiguation (context of the word was used to select the best sense) and (3) all senses (ILIs of all the synsets were returned). In last case, two words were considered as matched if one of their sense (or ILI) matches. In addition, hierarchal information (hypernym relationships) contained in the EuroWordNet was used to generalise 'specific words' with more 'general words'. For example, the specific words "car" and "truck" can be replaced with more general word "vehicle", which is a hypernym of both words. However, results with EuroWordNet did not show any significant improvement compared to a baseline approach.

Chong and Specia [2011] showed that using WordNet synsets to generalise content words, i.e. word unigrams, in the suspicious and source documents improves performance. Their method expands each content word in both the suspicious and source documents with all synonymous words in WordNet. The similarity score is computed by counting the number of common synsets in two documents normalised by the total number of synsets in both documents.

## 2.9 Structural Methods

The methods discussed in the previous sections rely on the text of the document. However, methods have been proposed which utilise the document structure. Generally, structural methods break a document into its constituent parts (sections, subsections, paragraphs etc.) and represent it as a tree structure in which each word is represented by a child node. Each level ($L$) of the tree represents a different part of the document. The structural similarity between a source and suspicious document, at a specified level

$L$, can be computed using a similarity function. If the similarity score exceeds some predefined threshold then the suspicious document is likely to contain plagiarism. Tree like document representations are more suitable for structured/semi-structured documents which are already divided into logical segments, such as theses and research papers.

Recently, Alzahrani et al. [2011] suggested that structural features can be categorised into (1) block-specific tree structure features and (2) content-specific tree structure features. The former method represents a document as a hierarchical structure of blocks. For example, *document-pages-paragraph* and *document-paragraph-sentence*. The latter method represents a document based on its semantic structure. For example, *document-section-paragraph* and *class-concept-chunk*.

For plagiarism detection, Chow and Rahman [2009] represented Web documents in HTML format as block-specific tree structures. The structure of an HTML document was represented as a three layered *document-pages-paragraph* tree structure. Each HTML document was parsed and paragraph blocks extracted using HTML tags like “<p>” and “<br>”. Paragraph blocks were merged to create a page and if the number of words in a page exceeded a certain threshold a new page was created. Each page is further divided into smaller blocks (called paragraphs) in a similar way to page creation. The upper layer of the three layered tree structure was used for search space reduction (or candidate document retrieval) by performing document clustering and the bottom layer was used to identify portions (or paragraphs) of plagiarised text using the cosine similarity measure. Recently, Zhang and Chow [2011] used the *document-paragraph-sentence* document representation for plagiarism detection.

Zini et al. [2006] represented a document as a three layered tree structure to compute document similarity at different granularity levels. The first level (or level 0) represented words, level 1 represented sentences and level 2 represented paragraphs. Levenshtein edit distance [Levenshteiti, 1966] (see Section 2.6.3) was used to compute similarity between different levels of the trees.

Wang et al. [2008] used *content-specific* tree structure based on *document-section-chunk* for plagiarism detection in Chinese academic research papers. The authors argue that different sections have different importance in a paper. Sections which describe "research method" and "experiments and analysis" are the most important. Their approach assigns higher weights to these sections and lower weights to other ones. The plagiarism score is computed by counting the number of common overlapping chunks between two paragraphs.

## 2.10 Efficiency of Plagiarism Detection Methods

This section examines the effect of different types of plagiarism on the extrinsic plagiarism detection methods discussed in the previous sections. Recently, Alzahrani et al. [2011] categorised plagiarism into: (1) literal plagiarism and (2) intelligent plagiarism. In the literal plagiarism, the original text is reused as verbatim (word to word copy) or with minor modifications to create the plagiarised document. They suggest that this type of plagiarised text can occur as: (1) exact copy, (2) near copy and (3) restructuring. In the intelligent plagiarism, the original text is modified to hide plagiarism. This type of plagiarism can be produced by: (1) paraphrasing, (2) summarising, (3) translating and (4) adopting other's ideas.

Table 2.2 shows the effect of different types of plagiarism on the efficiency of mono-lingual extrinsic plagiarism detection methods.[1] The symbol "✓" means that a plagiarism type can be detected by a plagiarism detection method. Lexical similarity (see Section 2.3) and probabilistic approaches (see Section 2.7) fall into the category of "Vector-Based" method and n-gram overlap (see Section 2.4), fingerprinting (see Section 2.5) and string and sequence comparison (see Section 2.6) fall into the category of "Character-Based" method.

[1] Note that Alzahrani et al. [2011] also presented intrinsic and cross-lingual plagiarism but only methods for mono-lingual extrinsic plagiarism detection are presented here.

| | Types of Plagiarism | | | | | | |
|---|---|---|---|---|---|---|---|
| | Literal Plagiarism | | | Intelligent Plagiarism | | | |
| Technique | Copy | Near Copy | Restructuring | Paraphrasing | Summarising | Translation | Idea |
| Vector-Based | ✓ | ✓ | ✓ | | | | |
| Character-Based | ✓ | ✓ | | | | | |
| Structural-Based | ✓ | ✓ | ✓ | | | | |
| Syntactic-Based | ✓ | ✓ | ✓ | | | | |
| Semantic-Based | ✓ | ✓ | ✓ | ✓ | | | |

Table 2.2: Mono-lingual extrinsic plagiarism detection methods and their efficiency in detecting different plagiarism types [Alzahrani et al., 2011]

All methods can easily identify plagiarism created by exact and near copy. The majority of methods are also not affected by restructuring of the original text. However, only "semantic-based" methods can identify plagiarism created by paraphrasing, which highlights the difficulty in detecting this type of plagiarism. To detect "translated" plagiarism, cross-lingual methods will be suitable. None of the methods can detect plagiarism created by summarisation because it is hard to find relationships between an original text and its summary. In addition, plagiarism of ideas cannot be detected because it is beyond the capability of current automatic plagiarism detection systems.

## 2.11 The PAN International Competitions on Plagiarism Detection

This section presents an overview of the three International Competitions on Plagiarism Detection (2009-2011) that have been organised under the umbrella of the PAN workshop: *Uncovering Plagiarism, Authorship and Social Software Misuse*. The first competition was held with SEPLN 2009 conference, while the 2nd and 3rd were held with the CLEF 2010 and 2011 conferences. The main goal was to develop standard resources and evaluation measures to enable a direct comparison of different methods for plagiarism detection.

The following sections give an overview of the plagiarism detection tasks, measures used to evaluate the performance of the participating plagiarism detection systems and algorithms/approaches used by majority of the participants for plagiarism detection.

### 2.11.1 Task

The two main tasks of each competition were: (1) extrinsic plagiarism detection and (2) intrinsic plagiarism detection. Since the focus of this work is on mono-lingual extrinsic plagiarism detection, only this task will be discussed in detail. The extrinsic plagiarism detection task included both mono-lingual and cross-lingual plagiarism. In the case of mono-lingual plagiarism, both the plagiarised and source texts were in English, whereas in the case of cross-lingual plagiarism, the source text was in either German or Spanish and the plagiarised text in English.

Each corpus was set up as $(D_{susp}, D_{src}, S)$, where $D_{susp}$ represents the suspicious collection, $D_{src}$ represents the source collection and $S$ represents the annotations for plagiarism cases between $D_{susp}$ and $D_{src}$. The plagiarism detection tasks were defined as [Stein et al., 2009].

- **Extrinsic Plagiarism Detection**

  *Given $D_{susp}$ and $D_{src}$ the task is to identify the sections in $D_{susp}$ which are plagiarised, and their source sections in $D_{src}$.*

- **Intrinsic Plagiarism Detection**

  *Given only $D_{susp}$ the task is to identify the plagiarised sections.*

In the 2010 competition, the intrinsic and extrinsic tasks were merged into a single task. Participants were allowed to participate in one or both tasks. In each competition, each task was divided into two phases.

- **Training Phase**

  In this phase, the training corpus $(D_{susp}, D_{src}, S)$ was released (with plagiarism annotations) for the development of plagiarism detection systems.

- **Testing Phase**

  In this phase, the test corpus $(D_{susp}, D_{src})$ was released (without plagiarism annotations). Participants were asked to submit their plagiarism detections using this corpus.

The winner of the competition was the participant whose system most accurately detected the plagiarism in the test corpus.

In the extrinsic plagiarism detection task, each training/testing corpus was divided into suspicious and source collections with a ratio of $50\% - 50\%$. In the suspicious collection, half of the documents were plagiarised and the remaining half non-plagiarised. Cases of plagiarism were created in two ways: (1) artificial - automatically created and (2) simulated - manually created (for more detail on corpus creation see Section 2.12.1).

### 2.11.2 Evaluation Measures for PAN Competitions

In these competitions, performance of the extrinsic plagiarism detection systems was evaluated at the section-level (instead of document-level) using the so-called *plagdet* (plagiarism detection) score, which is a combination of precision, recall and granularity measures.

A plagiarised document $d_q$ can be defined as a sequence of characters which are labeled as either plagiarised or non-plagiarised (note that there is a minimum and maximum length of $d_q$). A contiguous sequence of plagiarised characters makes a plagiarised fragment (or section) $s$. $S$ defines the set of all plagiarised fragments in

$d_q$, such that there is no overlap between plagiarised fragments, i.e. $\forall s_i, s_j \in S : i \neq j \rightarrow (s_i \cap s_j = \emptyset)$. Similarly $r \subset d_q$ represents the plagiarised fragment detected by the plagiarism detection system. $R$ is the set of all the detected fragments in $d_q$ [Stein et al., 2009].

The annotation of a plagiarism case is a four tuple: $(d_{src}, s_{src}, d_{plg}, s_{plg})$, where $s_{src}$ represents the source passage which was reused from the source document $d_{src}$ to create the plagiarised passage $s_{plg}$ and then it was randomly inserted into the plagiarised document $d_{plg}$ (see Section 2.12.1.2 for more detail on plagiarism case generation). The output of the plagiarism detection system for a plagiarism case is also formed from the same four tuples [Stein et al., 2009].

By considering characters as retrieval units, the precision $(prec(S, R))$ and recall $(rec(S, R))$ (see Section 2.13 for definitions of precision and recall) of the plagiarism detection system can be computed as [Stein et al., 2009]:

$$prec(S, R) = \frac{1}{|R|} \sum_{r \in R} \frac{|\bigcup_{s \in S}(r \sqcap s)|}{|r|} \tag{2.14}$$

$$rec(S, R) = \frac{1}{|S|} \sum_{s \in S} \frac{|\bigcup_{r \in R}(s \sqcap r)|}{|s|} \tag{2.15}$$

$$where \quad s \sqcap r = \begin{cases} s \cap r & \text{if r detects s,} \\ \emptyset & \text{otherwise.} \end{cases}$$

$S$ is the set of all plagiarised sections, $R$ is the set of all detected sections reported by the plagiarism detection system, $s \in S$ and $r \in R$.

A problem with precision and recall measures is that they do not penalise those plagiarised fragments which are reported more than once (or overlapping plagiarised fragments) for a single plagiarised fragment. To overcome this problem, a third measure is introduced to quantify the granularity of the plagiarism detection system.

$$gran(S,R) = \frac{1}{|S_R|} \sum_{s \in S_R} |C_s| \tag{2.16}$$

where $S_R = \{s | s \in S \wedge \exists r \in R : s \cap r \neq \emptyset\}$ represents that subset of $S$ which is detected and $C_s = \{r | r \in R \wedge s \cap r \neq \emptyset\}$ represents the subset of $R$ which detects a section $s$ [Stein et al., 2009].

The final score of the plagiarism detection system $plagdet(S,R)$ is computed by combining $prec(S,R)$, $rec(S,R)$ and $gran(S,R)$ as:

$$plagdet(S,R) = \frac{F_1}{log_2\,(1 + gran(S,R))} \tag{2.17}$$

where $F_1$ is the harmonic mean of precision ($prec(S,R)$) and recall ($rec(S,R)$) (see Section 2.13).

### 2.11.3 Approaches

In three PAN competitions (2009-2011), a total of 32 groups participated and 9 of them participated more than once. This section gives a summary of the approaches used by most of the participants for the detection of mono-lingual extrinsic plagiarism detection.

The extrinsic plagiarism detection framework shown in Figure 2.1 was followed by the majority of the systems that attempted the task of extrinsic plagiarism detection in the PAN competitions. The three main steps of this framework are: (1) candidate retrieval, (2) detailed analysis and (3) post-processing.

Given collections of suspicious $D_{plg}$ and source $D_{src}$ documents, in the candidate retrieval stage, a small subset of candidate source documents $D'_{src}$ is retrieved from $D_{src}$ for each suspicious document $d_{plg} \in D_{plg}$. In the detailed analysis stage, each suspicious document $d_{plg}$ is exhaustively compared with each candidate document $d_{src} \in D'_{src}$ to

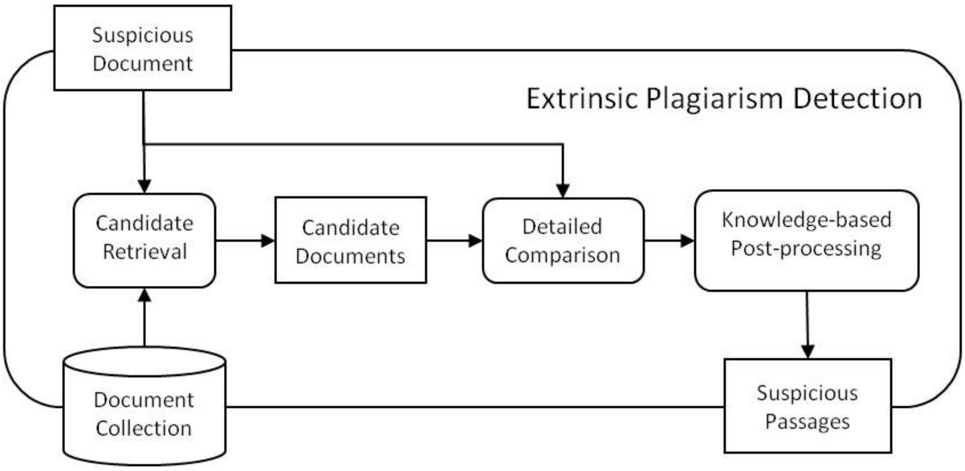

Figure 2.1: Generic steps for detecting plagiarised text [Stein et al., 2007]

identify plagiarised ($S_{plg}$) source ($S_{src}$) section pairs, where $S_{plg} \in d_{plg}$ and $S_{src} \in d_{src}$. Finally, in the post-processing stage, the section pairs identified in the previous step are filtered and remaining section pairs are reported as detections by the plagiarism detection system.

Below is a summary of the approaches used by the participants in these competitions (based on the framework shown in Figure 2.1). The focus is on retrieval and algorithmic aspects of the participating systems.

- **Candidate Document Retrieval:** For the candidate document retrieval task, the majority of the participants applied an IR-based approach with fingerprinting. Using this approach, the entire source collection $D_{src}$ is converted to fixed length word n-grams and hashed to generate fingerprints. After hashing, the $D_{src}$ is stored in an inverted index. To normalise the effects of obfuscation, each word n-gram is sorted before hashing. Some approaches apply stop word removal and stemming to reduce the effects of obfuscation.

  For each $d_{plg} \in D_{plg}$ fingerprints are generated in the same way. Each fingerprint in the $d_{plg}$ is queried in the index and source document $d_{src} \in D_{src}$, which shares at least $k$ fingerprints with $d_{plg}$ is marked as a potential candidate document.

- **Detailed Analysis:** For this stage, sequence alignment algorithms were used with match merging heuristics to get aligned plagiarised-source section pairs. In the first step, exact matching word n-grams (or seeds) were extracted between $d_{plg}$ and $d_{src}$. In the second step, match merging rules were applied on seeds to get longer aligned sections. A match rule decided whether two matches should be joined to get longer passages or not. Normally, rules were applied in the order of precedence. In the final step, aligned plagiarised-source section pairs were reported by the plagiarism detection system.

- **Post-processing:** In the post-processing, before reporting the final detections, the plagiarised-source section pairs (detected in the previous stage) were filtered. For example, sections shorter than a pre-defined length or whose similarity score was less than given threshold under a retrieval model were discarded. In addition, sections that were ambiguous (could have been derived from multiple source documents) were discarded.

## 2.12 Evaluation Resources for Plagiarism Detection

It is difficult to build a test collection with real examples of plagiarism due to the issue of confidentiality [Clough, 2003a]. In the last few years, the research community has made efforts to construct standard evaluation resources for plagiarism detection. These enable direct comparison of existing approaches for plagiarism detection and help to identify the levels of obfuscation that are easy/difficult to detect.

The next section describes the benchmark corpora that can be used for the evaluation of extrinsic plagiarism detection systems.

### 2.12.1 PAN-PC Corpora

An outcome of the three International Competitions on Plagiarism Detection (2009-2011) is a set of three benchmark corpora to evaluate plagiarism detection systems: (1) PAN-PC-09 Corpus [Stein et al., 2009], (2) PAN-PC-10 Corpus [Potthast et al., 2010a] and (3) PAN-PC-11 Corpus [Potthast et al., 2011] (see Section 2.11 for a detailed overview of these competitions). The following subsections describe the main characteristics of the PAN-PC corpora, how the cases of plagiarism were generated for these corpora and a brief description of each corpus.

#### 2.12.1.1 Characteristics of PAN-PC Corpora

The following factors were considered while developing the PAN-PC series of corpora [Barrón-Cedeño, 2012]:

- **Availability:** Evaluation resource(s) for plagiarism detection are not freely available, so one of the main goal was to develop corpora which can be made freely available. For this reason, freely available e-books on English literature from the Project Gutenberg[1] were used as base documents to construct the three PAN-PC corpora.

- **Embedded Plagiarism:** Instead of identifying plagiarism at the document level, the focus was on identifying fragments of text that are plagiarised and their corresponding source(s) fragments.

- **Scale:** Another aim was to develop corpora which contain thousands of documents making the plagiarism detection task more realistic and challenging.

- **Supporting Extrinsic and Intrinsic Plagiarism:** The two main subtasks for the plagiarism detection were: (1) extrinsic plagiarism detection task (source of

[1] http://www.gutenberg.org/ Last Visited: 31-05-2012

plagiarism is hidden in a large reference collection) and (2) intrinsic plagiarism detection task (no reference collection is available). Therefore, both types of plagiarism were included while constructing these corpora.

- **Variety of Plagiarism Types:** Cases of plagiarism in these corpora were created in three different ways: (1) exact copy - original text (or fragment) is reused verbatim (word to word copy) in the plagiarised document, (2) modified copy - original fragment is obfuscated (altered) before reusing for plagiarism and (3) translated copy - original fragment is translated from one language to another before being used in the plagiarised document. The *modified* cases of plagiarism can be created: (1) artificially and (2) manually. The former approach is cheap and quick but not ideal because plagiarism cases generated using this approach are unlikely to occur in real world (see Table 2.6), whereas the latter is expensive and time consuming but cases of plagiarism are more realistic. The majority of the plagiarism cases in these corpora were generated artificially and a small proportion were manually created.

- **Language Variety:** The corpora contain cases of cross-lingual plagiarism created by automatically translating text fragments from German or Spanish into English (only these three languages were used).

- **Positive and Negative Examples:** It is unlikely that every suspicious document examined for plagiarism will in fact be plagiarised. Therefore, in the collection of documents to be examined, 50% of the documents are plagiarised and remaining 50% not plagiarised.

#### 2.12.1.2 Case Generation

In the PAN-PC corpora, the process of generating a plagiarism case can be divided into two steps: (1) extraction-insertion and (2) obfuscation [Barrón-Cedeño, 2012].

- **Extraction-Insertion:** A fragment of source text $s_{src}$ is selected from the source document, which is used to create the plagiarised fragment $s_{plg}$ (either artificially or manually). Then the $s_{plg}$ is randomly inserted into the suspicious document to create a plagiarised document. Note that a single plagiarised document can contain plagiarised text fragment(s) from one or more source documents.

- **Obfuscation:** In order to create a modified copy of the source fragment $s_{src}$, it is first obfuscated (artificially or manually) to create $s_{plg}$ and then inserted into the suspicious document.

The cases of simulated plagiarism were generated by asking workers on the Amazon's Mechanical Turk[1] to rewrite $s_{src}$ passages to create $s_{plg}$ passages. Workers were instructed to heavily paraphrase the original passage and all cases were reviewed to check their quality.

The cases of artificial plagiarism were created by applying following operations on $s_{src}$ to generate $s_{plg}$ [Potthast et al., 2010a].

1. **Random operations:** Words in $s_{src}$ are randomly inserted, shuffled, removed or replaced to create $s_{plg}$.

2. **Semantic word variation:** Words in $s_{src}$ are substituted with their synonyms, hypernyms, hyponyms or antonyms, which are chosen at random from WordNet, to generate $s_{plg}$. Note that it reflects semantic-based paraphrasing.

3. **POS-preserving word shuffling:** The sequence of words in $s_{src}$ are re-arranged to create $s_{plg}$ in a way that it preserves the original part of speech sequence. Note that it reflects syntax-based paraphrasing.

4. **Machine translation:** The source fragment $s_{src}$ is translated using an automatic machine translation system from German or Spanish to English to create $s_{plg}$.

[1] https://www.mturk.com/mturk/welcome Last visited: 01-06-2012

This technique creates cases of cross-lingual plagiarism.

The cases of artificial plagiarism were created with three rewrite levels: (1) no obfuscation, (2) low obfuscation and (3) high obfuscation. In case of no obfuscation, $s_{src}$ is used verbatim to create $s_{plg}$, while in case of low and high obfuscations, $s_{src}$ is paraphrased using techniques discussed above to create $s_{plg}$ (see Table 2.6 for examples of artificial and simulated cases of plagiarism). The plagiarism cases created by high obfuscation are more strongly paraphrased compared to low obfuscation.

The next sections give a brief overview of the three PAN-PC corpora: (1) PAN-PC-09 Corpus, (2) PAN-PC-10 Corpus and (3) PAN-PC-11 Corpus for the extrinsic plagiarism detection task.

#### 2.12.1.3 PAN-PC-09 Corpus

The PAN-PC-09 Corpus[1] was developed for the 1st International Competition on Plagiarism Detection and only contains artificial (automatically generated) cases of plagiarism (both mono- and cross-lingual). This corpus contains 41,223 documents with 94,202 artificial cases of plagiarism for both intrinsic and extrinsic plagiarism detection tasks. For the extrinsic plagiarism detection task, the PAN-PC-09 test corpus contained 7214 suspicious documents (half plagiarised and half non-plagiarised) and the same number of source documents.

Table 2.3 gives a brief summary of the main statistics of the PAN-PC-09 corpus. In this corpus, 50% of the documents are used to make the source collection and the remaining 50% makes the suspicious collection (half plagiarised and half non-plagiarised). The length of a document varies from a single page to 1000 pages. 90% of the plagiarism cases are mono-lingual (in English) and remaining 10% are cross-lingual (automatically translated from German or Spanish to English). Only 20% of the cases of mono-lingual

[1] http://www.uni-weimar.de/cms/medien/webis/research/corpora/pan-pc-09.html Last visited: 29-05-2012

| Document Statistics | | | | Obfuscation Statistics | |
|---|---|---|---|---|---|
| Document Purpose | | Document Length | | | |
| source documents | 50% | short (1-10 pp.) | 50% | none | 35% |
| suspicious documents | | medium (10-100 pp.) | 35% | paraphrasing | |
| — with plagiarism | 25% | long (100-1000 pp.) | 15% | — automatic (low) | 35% |
| — without plagiarism | 25% | | | — automatic (high) | 20% |
| | | | | translation ({de, es} to en) | 10% |

Table 2.3: Statistics for the PAN-PC-09 Corpus [Stein et al., 2009]

plagiarism are created with high obfuscation (artificial).

#### 2.12.1.4 PAN-PC-10 Corpus

The PAN-PC-10 Corpus[1] [Potthast et al., 2010a] was created to evaluate the performance of the systems presented in the 2nd International Competition on Plagiarism Detection. It contains both automatic (artificial) and manual (simulated) cases of plagiarism (see Section 2.12.1.2). There are 27,073 documents with 68,558 cases of plagiarism in this corpus (70% of the documents are used for the extrinsic plagiarism detection task and remaining 30% for the intrinsic plagiarism detection task). The PAN-PC-10 Corpus is an enhanced version of the PAN-PC-09 Corpus.

Table 2.4 shows some statistics for the PAN-PC-10 Corpus. For the extrinsic plagiarism detection task, the PAN-PC-10 test corpus contains 12,134 suspicious documents (half plagiarised and half non-plagiarised) and the same number of source documents for the extrinsic plagiarism detection task. In total, 6,067 suspicious documents are plagiarised: none (artificial) = 1,916 (31.58%); low (artificial) = 1,354 (22.32%); high (artificial) = 1,337 (22.04%); simulated = 903 (14.88%) and translated = 557 (9.18%). Note that for 903 documents plagiarised with simulated obfuscation, 411 plagiarised documents contain only cases of simulated obfuscation and the remaining 492 documents contain both simulated and none (artificial) obfuscations.

In this corpus, 6% of the plagiarism cases were created manually (simulated) to make

[1] http://www.uni-weimar.de/cms/medien/webis/research/corpora/pan-pc-10.html Last visited: 29-05-2012

| | | Document Statistics | | | |
|---|---|---|---|---|---|
| Document Purpose | | Plagiarism per Document | | Document Length | |
| source documents | 50% | hardly (5%-20%) | 45% | short (1-10 pp.) | 50% |
| suspicious documents | | medium (20%-50%) | 15% | medium (10-100 pp.) | 35% |
| — with plagiarism | 25% | much (50%-80%) | 25% | long (100-1000 pp.) | 15% |
| — without plagiarism | 25% | entirely (> 80%) | 15% | | |
| | | Plagiarism Case Statistics | | | |
| Obfuscation | | Case Length | | Topic Match | |
| none | 40% | short (50-150 words) | 34% | intra-topic cases | 50% |
| artificial | | medium (300-500 words) | 33% | inter-topic cases | 50% |
| — low obfuscation | 20% | long (3000-5000 words) | 33% | | |
| — high obfuscation | 20% | | | | |
| simulated | 6% | | | | |
| translated | 14% | | | | |

Table 2.4: Statistics for the PAN-PC-10 Corpus [Potthast et al., 2010a]

the plagiarism detection task more challenging and realistic. Topical relationships were also identified between suspicious-source document pairs (see Table 2.4). The aim was to ensure some relationship between the source (from which $s_{src}$ is selected and then $s_{plg}$ is created) and suspicious (in which $s_{plg}$ is inserted) documents. A clustering technique was applied to group source and suspicious documents into 20 different clusters (e.g. religion, science or history). A pair of suspicious-source document can belong to: (1) intra-topic or (2) inter-topic. In the former case, the suspicious-source document pair is in the same cluster, whereas in the latter case the two documents are in different clusters.

#### 2.12.1.5 PAN-PC-11 Corpus

The PAN-PC-11 Corpus[1] was developed for the 3rd International Competition on Plagiarism Detection. In this corpus, documents were plagiarised using both artificial and simulated cases of plagiarism. This corpus contains 34,939 documents with 61,064 cases of plagiarism. For the extrinsic plagiarism detection task, the PAN-PC-11 test corpus contains 11,094 suspicious documents (half plagiarised and half non-plagiarised) and the same number of source documents (see Section 2.12.1.5). In the suspicious collec-

[1] http://www.uni-weimar.de/cms/medien/webis/research/corpora/pan-pc-11.html Last visited: 29-05-2012

| | | Document Statistics | | | |
|---|---|---|---|---|---|
| Document Purpose | | Plagiarism per Document | | Document Length | |
| source documents | 50% | hardly (5%-20%) | 57% | short (1-10 pp.) | 50% |
| suspicious documents | | medium (20%-50%) | 15% | medium (10-100 pp.) | 35% |
| — with plagiarism | 25% | much (50%-80%) | 18% | long (100-1000 pp.) | 15% |
| — without plagiarism | 25% | entirely (> 80%) | 10% | | |
| | | Plagiarism Case Statistics | | | |
| Obfuscation | | Case Length | | | |
| none | 18% | short (<150 words) | 35% | | |
| paraphrasing | | medium (150-1150 words) | 38% | | |
| — automatic (low) | 32% | long (>1150 words) | 27% | | |
| — automatic (high) | 31% | | | | |
| — manual | 8% | | | | |
| translation | | | | | |
| — automatic | 10% | | | | |
| — automatic + manual correction | 1% | | | | |

Table 2.5: Statistics for the PAN-PC-11 Corpus [Potthast et al., 2011]

tion, 5,547 documents are plagiarised: none (artificial) = 114 (2.05%); low (artificial) = 2,369 (42.70%); high (artificial) = 2,404 (43.33%); simulated = 105 (1.89%) and translated = 555 (10%).

A summary of statistics for this corpus is given in Table 2.5. This corpus is constructed on similar guidelines as that of the PAN-PC-10 Corpus. However, more emphasis was given to plagiarism created with paraphrasing (either manual or artificial) compared to the previous two corpora. 71% of the cases were created with paraphrasing and only 18% using no obfuscation (cut and paste).

#### 2.12.1.6 Examples of Plagiarism Cases

Table 2.6 shows examples of artificial and simulated cases of plagiarism in the PAN-PC corpora. As can be seen from these examples that cases of "none" obfuscation are created by copying the source text. In case of "low" and "high" obfuscations, the source text is paraphrased (automatically) and the rewritten text looks different from the original. However, the paraphrased text in not meaningful and this type of plagiarism is unlikely to occur in the real world. In the case of "simulated" obfuscation,

| |
|---|
| **None Obfuscation (Artificial)**<br>**Source:** The first agrarian movement after the enactment of lex Licinia took place in the year 338, after the battle of Veseris in which the Latini and their allies were completely conquered.<br>**Rewrite:** The first agrarian movement after the enactment of lex Licinia took place in the year 338, after the battle of Veseris in which the Latini and their allies were completely conquered. |
| **Low Obfuscation (Artificial)**<br>**Source:** Mr. Loring P. Rixford, Room 24, Menisini Building, 231 Post Street, San Francisco, is secretary. Brochure Series Competitions. From time to time, as opportunity offers, competitions in design will be conducted by THE BROCHURE SERIES. An upright or cabinet piano case, the subject of the first one, badly needs the attention of good designers.<br>**Rewrite:** Loring P. Rixford, area 24, Menisini edifice, 231 position Street, San Francisco, is head. book serial game. From day to case, as opportunity offers, contestant in arrangement will be conducted by THE BROCHURE SERIES. An erect or cabinet softness humiliation, the subject of the first one, well necessitate the work of good designers. |
| **High Obfuscation (Artificial)**<br>**Source:** "To John Bone, Bricklayer, died Dec. 14, 1794, aged 48 years." There is, however, one stone which may be included in the category of trade memorials, though its subject was not a mechanic. Mr. John Cade was a schoolmaster at Beckenham, and appears to have been well liked by his pupils, who, when he prematurely died, placed a complimentary epitaph over his grave.<br>**Rewrite:** aged 48 years. was at Beckenham, died Dec. 14 To were Bone, longitude sixty Bricklayer however, was one stone Mr. room trade. Cade of coast 1794,, appears to been good our by his, who, when not a. and prematurely in the category of memorials, of lost is in,, and complimentary epitaph over may s inhabitants. |
| **Simulated Obfuscation (Manual)**<br>**Source:** The emigrants who sailed with Gilbert were better fitted for a crusade than a colony, and, disappointed at not at once finding mines of gold and silver, many deserted; and soon there were not enough sailors to man all the four ships.<br>**Rewrite:** The people who left their countries and sailed with Gilbert were more suited for fighting the crusades than for leading a settled life in the colonies. They were bitterly disappointed as it was not the America that they had expected. Since they did not immediately find gold and silver mines, many deserted. At one stage, there were not even enough man to help sail the four ships. |

Table 2.6: Example plagiarism cases from PAN-PC corpora

the original text is efficiently paraphrased to create the rewritten text. These cases are more realistic examples of plagiarism.

### 2.12.2 MEDLINE Corpus

MEDLINE (Medical Literature Analysis and Retrieval System Online) is an online database of research articles in the area of medicine and related fields.[1] The MEDLINE database is regularly updated with new research publications. At July 2012, the database contains more than 21.8 million citations.[2] The majority of citations consist of a title and abstract, in addition to other useful information, for example, author(s) name, publication date, journal name. The MEDLINE/PubMed Baseline Repository[3] presents a static view of the MEDLINE database at the time each baseline repository is generated.

Errami et al. [2008, 2010] used an automatic text similarity tool called eTBLAST [Errami et al., 2007; Lewis et al., 2006] to identify highly similar citation pairs in MEDLINE. The aim of this study was to identify real potential cases of plagiarism in the biomedical domain. A total 79,383 highly similar Medline citation pairs were identified and compiled in the *Deja vu* database.[4] Each duplicate citation pair was classified into four categories:[5] (1) duplicate citation pairs having Shared Author (SA), (2) duplicate citation pairs written by Different Authors (DA) i.e. no-shared authors, (3) duplicate citation pairs published in the Same Journal (SJ) and (4) duplicate citation pairs published in Different Journals (DJ) [Errami et al., 2008].

Out of 79,383 highly similar citation pairs identified using eTBLAST [Errami et al., 2007; Lewis et al., 2006], only a subset of 2,106 citation pairs have been manually examined and verified as true duplicate citation pairs. Among the manually examined

[1] http://www.ncbi.nlm.nih.gov/pubmed/ Last visited: 27-06-2012

[2] http://www.nlm.nih.gov/bsd/revup/revup_pub.html#med_update Last visited: 27-06-2012

[3] http://mbr.nlm.nih.gov/ Last visited: 27-06-2012

[4] http://dejavu.vbi.vt.edu/dejavu/duplicate/ Last visited: 12-10-2011

[5] There are also other categories but these four are more relevant to plagiarism.

**MEDLINE Corpus**

**Source:** Gammaglutamyl transpeptidase **is an enzyme primarily located in the brush border of the proximal convoluted** tubules of the kidney. Its unique localisation in the renal **cells most easily damaged by ischaemia and its ease of assay** provides **the rationale for its use in the measurement** of renal ischaemic injury. Using a standard experimental animal model, canine urinary gamma-GT activity was shown to be increased up to 70-fold following 90 min of unilateral renal ischaemia and was significantly raised following only 5 min ischaemia. The urinary gamma-GT was used as a measure of ischaemic injury associated with renal transplantation in man and 20 consecutive patients undergoing kidney transplant were studied by daily 24-hour urinary gamma-GT estimations and **excellent correlation was obtained between raised** enzyme activity **and the clinical diagnosis of transplant rejection**.

**Rewrite:** The sites of ischaemic injury within the kidney are reviewed and the diagnostic value of measurements of plasma and urinary enzymes in renal ischaemic injury and in renal homotransplant rejection in experimental animals and man is examined. Gamma-glutamyl transpeptidase (gamma-GT) **is an enzyme primarily located in the brush border of the proximal convoluted** tubule of the kidney. Its unique localization in the **cells most easily damaged by ischaemia and its ease of assay** provide **the rationale for its use in the measurement** and diagnosis of renal ischaemic injury. gamma-GT activity was measured in dogs undergoing varying periods of renal ischaemia and under conditions of local renal hypothermia and was shown to be a sensitive indicator of ischaemic injury. Twenty consecutive patients undergoing renal homotransplantation were studied by daily estimation of their 24-h urinary gamma-GT activity; **excellent correlation was obtained between raised** levels of this enzyme **and the clinical diagnosis of transplant rejection**.

Table 2.7: Example duplicate citation pair from 265 manually examined and verified duplicate citation pairs in the *Deja vu* database.

duplicate citation pairs, 265 pairs are written by Different Authors (DA) and 1,841 pairs have Shared Authors (SA). Although highly similar citation pairs are identified at title and abstract level, Errami et al. [2008] suggested that highly similar duplicate citation pairs with no shared author are potential cases of plagiarism.

Table 2.7 shows an example of a plagiarism case in the MEDLINE corpus. There are five long exact matches (shown in bold font) whose length is greater than five tokens in the source and rewritten texts. These long exact matches are unlikely to occur by chance in a small passage and are strong indicator of plagiarism. There are also other exact matches whose length is less than five tokens. It can also be noted that some of the matches could be even longer but there are some insertions in the rewritten text.

### 2.12.3 METER Corpus

The METER Corpus [Gaizauskas et al., 2001] was created to measure text reuse in journalism as part of the METER project.[1] The corpus contains 1,716 documents, 771 Press Association (PA) articles and 945 news stories published by nine British newspapers. Some of the news stories were based on the PA articles. The news stories were related to two domains: (1) British court and law reporting and (2) showbusiness. 769 news articles were about court and law reporting and the remaining 176 on showbusiness.

Each news story was manually examined and based on the amount of text reused from the PA source text classified at document level into one of the following categories:

**Wholly Derived (WD)** the newspaper article is likely to be derived entirely from the PA source text.

**Partially Derived (PD)** some of the newspaper article is derived from the PA source text.

**Non Derived (ND)** the news story is likely to be written independently of the PA source text.

Out of 945 news stories, 301 are WD, 438 are PD and 206 are ND. Documents belonging to the WD and PD categories can be combined to make a set of "Derived" documents, whereas ND documents can be treated as a set of "Non-Derived" documents.

Although, text reuse in journalism is acceptable, Clough [2003b] suggested that this corpus can also be used to evaluate the performance of plagiarism detection sys-

[1] http://nlp.shef.ac.uk/meter/ Last visited: 01-06-2012

**METER Corpus**

**Source:** The waterlogged conditions that ruled out play yesterday still prevailed at Bourda this morning, and it was not until mid-afternoon that the match restarted. Less than three hours' play remained, and with the West Indies still making their first innings reply to England's total of 448, there was no chance of a result. At tea the West Indies were two for 139.

**Rewrite:** Waterlogged conditions ruled out play this morning, but the match resumed with less than three hours' play remaining for the final day. The West Indies are making a first innings reply to England's total of 448. At tea the West Indies were 139 for two, but there's no chance of a result.

Table 2.8: Example of text reuse from the METER corpus

tems. This corpus has been used to evaluate extrinsic plagiarism detection system, for example, Barrón-Cedeño et al. [2009].

Table 2.8 shows an example of text reuse in the METER Corpus. The source text has been paraphrased to create the news story (rewrite).

### 2.12.4 Short Answer Corpus (SAC)

The Short Answer Corpus [Clough and Stevenson, 2011] contains examples of simulated plagiarism designed to simulate plagiarism in academia. It was created as answers to five questions on a range of topics in Computer Science. The length of each answer was between 200-300 words. All the documents in this corpus were manually (simulated) created. A total of 19 subjects participated in the corpus generation process. Each participant answered each of the following five questions only once to create plagiarised and non-plagiarised documents.

(A) What is inheritance in object oriented programming?
(B) Explain the PageRank algorithm that is used by the google search engine.
(C) Explain the Vector Space Model that is used for Information Retrieval.
(D) Explain Bayes Theorem from probability theory.
(E) What is dynamic programming?

Subjects were allowed to use any part of the original Wikipedia[1] article to produce the plagiarised answer. Instructions were provided for creating plagiarised answers with different rewrite levels:

**Near copy** Use the source Wikipedia article to answer the question by using cut-and-paste operations. However, the length of the answer should be between 200-300 words.

**Light revision** The answer to the question should be based on the source Wikipedia page. The original text should be altered by paraphrasing techniques like synonym replacement and changing the grammatical structure. Moreover, in sentences the information order should be preserved.

**Heavy revision** Again the answer should be based on the original Wikipedia page but it should be generated by rephrasing the original text such that same content is expressed using different linguistic expressions. This may include sentence merging and splitting.

Instructions for creating the non-plagiarised answers are as follows:

**Non-plagiarism** Subjects were instructed to answer the question using their own knowledge and what they have learned from the learning material (lecture notes, relevant sections from textbooks etc.) provided to them. While answering a question they can look at other relevant material but not Wikipedia.

A total of 95 documents were created, 57 were plagiarised (near copy = 19, light revision = 19 and heavy revision = 19) and remaining 38 were non-plagiarised. The set of non-plagiarised documents is useful in evaluating the ability of a plagiarism detection system to discriminate plagiarised documents from non-plagiarised ones. In total,

[1] http://www.wikipedia.org/

| **Short Answer Corpus** |
|---|
| **Source:** In object-oriented programming, inheritance is a way to form new classes (instances of which are called objects) using classes that have already been defined. The inheritance concept was invented in 1967 for Simula. |
| **Rewrite:** When we talk about inheritance in object-oriented programming languages, which is a concept that was invented in 1967 for Simula, we are usually talking about a way to form new classes and classes are instances of which are called objects and involve using classes that have already been defined. |

Table 2.9: Example of simulated case of plagiarism (heavy revision) from the Short Answer Corpus

this corpus contains 100 documents, 95 suspicious documents and 5 source Wikipedia articles.

Table 2.9 shows an example of plagiarism in the Short Answer Corpus. This is an example of heavy revision and it can be seen that source text has been paraphrased to create the rewritten text. Both the source and rewritten texts convey the same content but have used different linguistic expressions.

### 2.12.5 Analysis of Evaluation Resources

For experiments presented in this thesis, evaluation was carried out using four benchmark corpora (see Chapters 3, 4 and 5): (1) PAN-PC-10 Corpus, (2) MEDLINE Corpus, (3) Short Answer Corpus and (4) METER Corpus. Table 2.10 summarises the main characteristics of these corpora. In the PAN-PC corpora, the PAN-PC-09 corpus only contains artificial cases of plagiarism which are not realistic (see Table 2.6). The other two PAN corpora contain artificial and simulated cases of plagiarism. The PAN-PC-11 corpus was developed using similar approach as that of the PAN-PC-10 corpus. Therefore, among three PAN-PC corpora, the PAN-PC-10 Corpus was selected for experiments.

These corpora are chosen for this study because they contain: (1) a variety of

| | Corpora | | | |
|---|---|---|---|---|
| | PAN-PC-10 | MEDLINE | SAC | METER |
| Domain | English literature | Biomedical | Computer Science | Journalism |
| Reuse Type | artificial, simulated | real | simulated | real |
| Obfuscation Levels | none, low, high | not defined | none, low, high | WD, PD, ND |
| Source Collection | 12,134 | 19,569,568 | 5 | 771 |
| Suspicious Collection | 12,134 | 79,383 | 95 | 945 |

Table 2.10: Summary of main characteristics of the four benchmark corpora used to evaluate the performance of proposed approaches.

different types of text reuse: artificial, simulated and real, (2) text reuse examples with different levels of obfuscation and (3) reused documents of different lengths and from different domains (see Table 2.10). This makes the evaluation task more realistic and challenging.

## 2.13 Evaluation Measures

In Information Retrieval (IR), the set of documents that completely satisfy a user's information need (represented as a query) are called *relevant* and the set of documents that are returned by an IR system against a user's query are called *retrieved*. The performance of an IR system can be evaluated using the precision and recall measures. Precision is defined as the fraction of retrieved documents that are relevant against query (see Equation 2.18) and recall is defined as the fraction of relevant documents that are retrieved against query (see Equation 2.19) [Baeza-Yates and Ribeiro-Neto, 2011].

$$precision = \frac{|retrieved \bigcap relevant|}{|retrieved|} \tag{2.18}$$

$$recall = \frac{|retrieved \bigcap relevant|}{|relevant|} \tag{2.19}$$

The value of precision measure is between 0 to 1, where 0 means that none of the relevant document is retrieved and 1 means that all the retrieved documents are relevant. The value of recall measure also ranges between 0 to 1, where 0 means no relevant documents have been retrieved and 1 means all relevant documents have been retrieved.

Generally there is a trade off between precision and recall. Maximum recall (100%) can be achieved if all the documents in the source collection are assumed to be relevant, but precision will be low. To overcome this problem, different measures have been proposed to combine precision and recall. A popular measure that combines precision and recall is the $F$ measure. It is computed as [Baeza-Yates and Ribeiro-Neto, 2011]:

$$F_\alpha = \frac{(1+\alpha^2) \cdot p \cdot r}{\alpha^2 \cdot p + r} \tag{2.20}$$

where $p$ is precision, $r$ is recall and $\alpha$ is the weight assigned to precision or recall. If equal weights are assigned to precision and recall i.e. $\alpha = 1$, result is the $F_1$ measure (the harmonic mean of precision and recall), which is computed as:

$$F_1 = \frac{2 \cdot p \cdot r}{p + r} \tag{2.21}$$

The standard IR evaluation measures can be used for evaluating the performance of plagiarism detection systems. For plagiarism detection, the source documents that were used to create the plagiarised document can be treated as the set of *relevant* documents, while the set of documents returned by the plagiarism detection system as potential sources can compose the set of *retrieved* documents for the plagiarised document. Using the sets of *relevant* and *retrieved* documents, precision, recall and F measures can be computed using Equations 2.18, 2.19 and 2.20 respectively.

## 2.14 Chapter Summary

The task of plagiarism detection is divided into two main subtasks: (1) extrinsic plagiarism detection and (2) intrinsic plagiarism detection. In extrinsic plagiarism, the plagiarised-source text pair can be in the same language (mono-lingual) or different languages (cross-lingual). This chapter described state-of-the-art methods for the mono-lingual extrinsic plagiarism detection including lexical similarity, overlap of n-grams, fingerprinting, string or sequence comparison, probabilistic methods, syntactic methods, semantic methods and structural methods. After that an overview of the International Competitions on Plagiarism Detection was presented, which discussed the plagiarism detection tasks, measures used to evaluate the performance of participating systems and methods proposed to detect plagiarism.

Benchmark corpora that can be used to evaluate the performance of extrinsic plagiarism detection systems were presented: (1) PAN-PC Corpora (PAN-PC-09 Corpus, PAN-PC-10 Corpus and PAN-PC-11 Corpus), (2) MEDLINE Corpus, (3) METER Corpus and (4) Short Answer Corpus. These corpora contain artificial, simulated and real examples of text reuse. Finally, measures commonly used to evaluate the performance of plagiarism detection systems precision, recall and $F$ measure are described.

# Chapter 3

# IR-Based Framework for Candidate Document Retrieval

## 3.1 Introduction

This chapter describes an Information Retrieval (IR)-based approach for the problem of candidate document retrieval (see Section 3.3). When the problem is viewed this way, the query is formed from the suspicious document. The proposed IR-based approach is compared with a state-of-the-art approach (Kullback-Leibler Distance; see Section 2.7.1) for the candidate document retrieval task. Evaluation is carried out using three benchmark corpora: (1) PAN-PC-10 Corpus, (2) MEDLINE Corpus and (3) Extended Short Answer Corpus.

The rest of this chapter is structured as follows: Section 3.2 describes the motivation for attempting the candidate document retrieval problem with an IR-based approach. Section 3.3 presents the proposed IR-based framework. Section 3.4 presents the datasets, evaluation measure and implementation details. Finally, results and analysis are presented in Section 3.5.

## 3.2 Motivation for using IR-based Approach

Two broad approaches for candidate document retrieval are [Potthast et al., 2010b, 2011; Stein et al., 2009]: fingerprinting and Information Retrieval (IR). In fingerprinting (see Section 2.5), the content of a document is represented by hashing subsequences or substrings of the words in a document. Similarity between a pair of documents is computed by counting the number of common fingerprints. In the IR-based approach (see Section 2.3), the document suspected to contain plagiarised text is used as a query against an index of potential source documents to retrieve a set of ranked source documents. The document(s) at the top of the list are more likely to be the source(s) of plagiarism than those ranked lower down.

To identify the most suitable approach for the problem of candidate document selection, analysis of the systems presented in the 2nd [Potthast et al., 2010b] and 3rd [Potthast et al., 2011] International Competitions on Plagiarism Detection is carried out for the extrinsic plagiarism detection task (see Section 2.11.1 for description of the task). The 1st International Competition on Plagiarism Detection (PAN 2009 Competition) [Stein et al., 2009] is not included in this analysis because it only contains artificial examples of plagiarism which are unlikely to appear in real cases of plagiarism (see Table 2.6 for artificial examples of plagiarism in the PAN-PC corpora).

In each PAN competition, the formal evaluation of the participating extrinsic plagiarism detection systems was carried out at passage-level. The final run submission of a plagiarism detection system should report the position of the plagiarised passage in the suspicious document and position of its corresponding source passage in the source document (which was used to create the plagiarised passage). If a plagiarism detection system correctly identifies the source document used to plagiarise the suspicious document but does not correctly report the position of the source passage, it will not contribute to the overall plagiarism detection score (see Section 2.11.2).

The main goal of the candidate document retrieval task is to identify the source document(s) that are used to plagiarise. Therefore, for this analysis, the final run submissions of all the participating systems are used to calculate precision, recall and $F_1$ scores at the document-level,[1] i.e. checking whether the source document(s) used to create a plagiarised document are identified or not in the final run submission. The performance of a plagiarism detection system at passage-level is ignored.

Note that in the PAN 2010 and 2011 competitions there were no separate retrieval results for the candidate document selection stage. Therefore, for this analysis, plagiarism passage detection results are used as indirect evidence of extent of retrieval success/failure. However, it is not necessary that an original source document (used to plagiarise a document) which is not reported in the final passage level results was not retrieved at the candidate document selection stage. The following two subsections present the document-level analysis of the PAN 2010 and 2011 competitions.

### 3.2.1 PAN 2010 Competition

Based on the passage-level formal evaluation of the participating systems in the competition, systems that came first [Kasprzak and Brandejs, 2010] (kasprzak) and second [GuangZhou et al., 2010] (zou) used a hash (or fingerprinting) approach; the system that came third [Muhr et al., 2010] (muhr) attempted the problem with an IR-based approach; the system that came fourth [Grozea and Popescu, 2010] (grozea) used character 16-grams to make a pairwise comparison of documents; and the system that came fifth [Oberreuter et al., 2010] (oberreuter) used word bi-grams and tri-grams features for detecting plagiarism.[2]

Table 3.1 shows the document-level performance of the top 5 extrinsic plagiarism detection systems presented in the PAN 2010 competition for detecting extrinsic plagia-

[1] The final run submissions were obtained from the organisers of the competitions.

[2] In total, 18 groups participated in the PAN 2010 competition for the extrinsic plagiarism detection task. For a comprehensive analysis, only the top 5 systems are discussed.

| PAN 2010 Competition | | | | | | | |
|---|---|---|---|---|---|---|---|
| All Obfuscations | | | | Simulated Obfuscation | | | |
| Participant | $P$ | $R$ | $F_1$ | Participant | $P$ | $R$ | $F_1$ |
| muhr | 0.8983 | 0.8234 | 0.8457 | muhr | 0.5645 | 0.4856 | 0.5221 |
| kasprzak | 0.8936 | 0.8124 | 0.8369 | grozea | 0.4643 | 0.3879 | 0.4227 |
| zou | 0.8859 | 0.7784 | 0.8139 | zou | 0.4388 | 0.3631 | 0.3974 |
| grozea | 0.7178 | 0.5787 | 0.6222 | kasprzak | 0.3686 | 0.2942 | 0.3272 |
| oberreuter | 0.6888 | 0.5509 | 0.5957 | oberreuter | 0.3582 | 0.2966 | 0.3245 |
| Low Obfuscation | | | | High Obfuscation | | | |
| muhr | 0.9675 | 0.9108 | 0.9314 | muhr | 0.9518 | 0.7803 | 0.8429 |
| kasprzak | 0.9734 | 0.8999 | 0.9277 | zou | 0.9438 | 0.7252 | 0.8040 |
| zou | 0.9758 | 0.8893 | 0.9231 | grozea | 0.9390 | 0.6705 | 0.7632 |
| grozea | 0.9434 | 0.7501 | 0.8201 | kasprzak | 0.9201 | 0.6705 | 0.7572 |
| oberreuter | 0.9333 | 0.7332 | 0.8057 | oberreuter | 0.9030 | 0.6142 | 0.7106 |
| None Obfuscation | | | | Translated Obfuscation | | | |
| kasprzak | 0.9824 | 0.9606 | 0.9653 | kasprzak | 0.9730 | 0.9477 | 0.9547 |
| zou | 0.9843 | 0.9390 | 0.9546 | muhr | 0.8990 | 0.8222 | 0.8451 |
| muhr | 0.9712 | 0.9274 | 0.9410 | zou | 0.8777 | 0.7259 | 0.7779 |
| oberreuter | 0.9721 | 0.8374 | 0.8867 | grozea | 0.0829 | 0.0255 | 0.0354 |
| grozea | 0.9607 | 0.8149 | 0.8659 | oberreuter | 0.0287 | 0.0074 | 0.0111 |

Table 3.1: Document-level performance for detecting extrinsic plagiarism with different types of obfuscation for the top 5 systems presented in the PAN 2010 competition

rism.[1] Participating systems are ranked using the $F_1$ score. In this table, "Simulated Obfuscation" means that only the documents plagiarised with simulated cases of plagiarism are used to evaluate a system's performance (411 documents; see Section 2.12.1.4); "Low Obfuscation" means that only the documents plagiarised with low (artificial) obfuscation are used (1,354 documents; see Section 2.12.1.4); "All Obfuscations" means that all the plagiarised documents containing all types of obfuscation are used (6,607 documents; see Section 2.12.1.4) and so on.

According to these results, *muhr's* IR-based approach ($F_1$ = 0.8457) outperforms all other approaches in detecting various types of obfuscation (see performance for *All Obfuscations* in Table 3.1). This high score indicates that overall systems performed

[1] For this analysis, suspicious plagiarised documents in the PAN-PC-10 test corpus are used (see Section 2.12.1.4).

well in detecting extrinsic plagiarism.

Regarding systems performance in detecting a particular obfuscation, *muhr's* approach gives the best results with simulated ($F_1 = 0.5221$), low ($F_1 = 0.9314$) and high ($F_1 = 0.8429$) obfuscations. This shows that this approach is more robust for detecting paraphrased text (both automatic and manual).

For cases of no obfuscation, the hash-based approach (*kasprzak*; $F_1 = 0.9653$) performs slightly better than the IR-based approach (*muhr*; $F_1 = 0.9410$). However, for cross-lingual plagiarism detection (translated obfuscation), *kasprzak's* performance is much higher than *muhr*. Since the focus of this work is on mono-lingual extrinsic plagiarism detection, performance on translated plagiarism will not influence the selection of the appropriate approach.

Among different types of obfuscation, it can be noted that it was relatively straightforward to identify none, low, high and translated obfuscations. However, systems failed to give promising results for simulated obfuscation. This highlights the fact that simulated (manually paraphrased) cases of plagiarism in the PAN-PC-10 Corpus [Potthast et al., 2010a] are the most difficult to detect.

### 3.2.2 PAN 2011 Competition

To further investigate the most appropriate approach for the candidate document retrieval task, document-level analysis of the extrinsic plagiarism detection systems presented in the PAN 2011 competition [Potthast et al., 2011] is also carried out. For this analysis, suspicious plagiarised documents in the PAN-PC-11 test corpus are used (see Section 2.12.1.5).

Based on the formal evaluation, the system that came first [Grman and Ravas, 2011] (grman) in the competition attempted the problem by computing the number of common words between a pair of suspicious-source passage; the system coming second [Grozea and Popescu, 2011] (grozea) made pairwise comparison of documents using

| PAN 2011 Competition | | | | | | | |
|---|---|---|---|---|---|---|---|
| All Obfuscations | | | | Simulated Obfuscation | | | |
| Participant | $P$ | $R$ | $F_1$ | Participant | $P$ | $R$ | $F_1$ |
| grman | 0.7547 | 0.6342 | 0.6660 | grozea | 0.8822 | 0.4251 | 0.5552 |
| grozea | 0.7305 | 0.6379 | 0.6553 | grman | 0.8257 | 0.3251 | 0.4528 |
| oberreuter | 0.4674 | 0.4081 | 0.4167 | oberreuter | 0.7261 | 0.2815 | 0.3918 |
| palkovskii | 0.4552 | 0.4137 | 0.4135 | palkovskii | 0.7419 | 0.2721 | 0.3812 |
| rao | 0.4536 | 0.4197 | 0.4080 | torrejon | 0.7140 | 0.2389 | 0.3407 |
| torrejon | 0.4849 | 0.3585 | 0.3951 | cooke | 0.6553 | 0.1126 | 0.1846 |
| cooke | 0.4383 | 0.2894 | 0.3280 | rao | 0.5453 | 0.1058 | 0.1537 |
| nawab | 0.3316 | 0.2827 | 0.2848 | nawab | 0.5361 | 0.0859 | 0.1367 |
| ghosh | 0.0318 | 0.0520 | 0.0322 | ghosh | 0.0690 | 0.0201 | 0.0263 |
| Low Obfuscation | | | | High Obfuscation | | | |
| grozea | 0.9414 | 0.9368 | 0.9262 | grman | 0.4956 | 0.3162 | 0.3606 |
| grman | 0.9561 | 0.8822 | 0.9056 | grozea | 0.4843 | 0.3149 | 0.3555 |
| palkovskii | 0.8243 | 0.7921 | 0.7802 | rao | 0.1876 | 0.1412 | 0.1461 |
| oberreuter | 0.8104 | 0.7754 | 0.7725 | oberreuter | 0.1999 | 0.1191 | 0.1368 |
| torrejon | 0.7920 | 0.6291 | 0.6815 | cooke | 0.1533 | 0.0681 | 0.0874 |
| rao | 0.6600 | 0.6389 | 0.6171 | torrejon | 0.1337 | 0.0634 | 0.0791 |
| nawab | 0.6547 | 0.5803 | 0.5823 | nawab | 0.0643 | 0.0422 | 0.0456 |
| cooke | 0.5971 | 0.3832 | 0.4429 | palkovskii | 0.0556 | 0.0353 | 0.0394 |
| ghosh | 0.0568 | 0.0879 | 0.0577 | ghosh | 0.0080 | 0.0162 | 0.0087 |
| None Obfuscations | | | | Translated Obfuscation | | | |
| grman | 0.9857 | 0.9961 | 0.9888 | grman | 0.9562 | 0.9367 | 0.9404 |
| cooke | 0.9328 | 0.9427 | 0.9327 | cooke | 0.8520 | 0.7465 | 0.7821 |
| grozea | 0.8962 | 0.9955 | 0.9293 | grozea | 0.8340 | 0.7278 | 0.7605 |
| oberreuter | 0.9206 | 0.9368 | 0.9241 | rao | 0.6640 | 0.6758 | 0.6435 |
| torrejon | 0.9018 | 0.8836 | 0.8857 | torrejon | 0.5666 | 0.3964 | 0.4512 |
| palkovskii | 0.8263 | 0.9978 | 0.8799 | palkovskii | 0.4801 | 0.3440 | 0.3792 |
| rao | 0.6643 | 0.7790 | 0.6746 | oberreuter | 0.0198 | 0.0073 | 0.0099 |
| nawab | 0.6808 | 0.7280 | 0.6729 | ghosh | 0.0000 | 0.0003 | 0.0001 |
| ghosh | 0.1337 | 0.3454 | 0.1619 | nawab | 0.0000 | 0.0000 | 0.0000 |

Table 3.2: Document-level performance for detecting extrinsic plagiarism with different types of obfuscation for all the systems presented in the PAN 2011 competition

character 16-grams. Word 4-grams (candidate document selection stage) and 3-grams (detailed analysis stage) were used by the system that came third [Oberreuter et al., 2011] (oberreuter). The systems that came fourth [Cooke et al., 2011] (cooke), fifth [Torrejón and Ramos, 2011] (torrejon) and sixth [Rao et al., 2011] (rao) employed an IR-based approach. The system that came seventh [Palkovskii et al., 2011] (palkovskii) applied Wordnet-based semantic similarity measures to detect extrinsic plagiarism. An IR-based approach was also used by the systems that came eighth [Nawab et al., 2011] (nawab) and ninth [Ghosh et al., 2011] (ghosh).[1]

Table 3.2 shows the document-level performance of all the systems presented in the PAN 2011 competition. In case of "All Obfuscations", the best result is obtained using *grman's* approach ($F_1 = 0.6660$). However, this score is quite low compared to the best performance in the PAN 2010 competition (*muhr*; $F_1 = 0.8457$). Similarly, results for translated, low and high obfuscation are lower than those reported in the PAN 2010 competition. This indicates that it is difficult to detect paraphrased text because majority of the plagiarism cases in this corpus are created with paraphrasing (see Section 2.12.1.5). Regarding simulated obfuscation, although the best result (*grozea*; $F_1 = 0.5552$) is high compared to the PAN 2010 competition (*muhr*; $F_1 = 0.5221$), there is still room for improvement.

Systems perform well in detecting cases of low obfuscation (*grozea*; $F_1 = 0.9262$) but highest $F_1$ score (*grman*) of 0.3606 is achieved for high obfuscation. Low performance in detecting highly obfuscated cases seems to be the main reason for overall low performance because 43.33% of the plagiarised documents contain this type of obfuscation (see Section 2.12.1.5). Finally, for none and translated obfuscations, the best results are achieved by *grman* with $F_1$ scores of 0.9888 and 0.9404 respectively.

To conclude, document-level analysis of the PAN 2010 and 2011 competitions showed that the best document-level performance for detecting different types of obfus-

[1] In total, 9 groups participated in the extrinsic plagiarism detection task of the PAN 2011 competition and all of them are presented in the analysis.

cation ("All Obfuscations") is obtained using IR-based approaches (*muhr*; $F_1 = 0.8457$ (see Table 3.1) and *grman*; $F_1 = 0.6660$ (see Table 3.2)). These methods also perform well in detecting plagiarism when the source text has been paraphrased; cases of low, high and simulated obfuscation, which is the main focus of this work. Therefore, the problem of candidate document selection is attempted using an Information Retrieval approach. The following section describes the proposed approach in detail.

## 3.3 IR-Based Approach

Figure 3.1 shows the proposed process for retrieving candidate source documents using an IR-based approach. The source collection is indexed with an IR system (an offline step). The candidate retrieval process can be divided into four main steps: (1) preprocessing, (2) query formulation, (3) retrieval and (4) result merging. These steps are described as follows:

1. **Pre-processing:** Each suspicious document is split into sentences.[1] Each sentence is lower-cased, stop words[2] and punctuation marks are removed. The remaining words in a sentence are stemmed using the Porter Stemmer [Porter, 1980].

2. **Query Formulation:** Sentences from the suspicious document are used to make a query. The length of a query can vary from a single sentence to all the sentences appearing in a document, i.e. the entire document, because text reused for plagiarism can be obtained from one or more documents and the amount of text reused for plagiarism can vary from a single sentence to an entire document. A long query is likely to perform well in situations when large portions of text are reused for plagiarism. On the other hand, small portions of plagiarised text are

[1] NLTK sentence detector [Loper and Bird, 2002] was used for these experiments.

[2] NLTK [Loper and Bird, 2002] stop word list of 127 words in English was used.

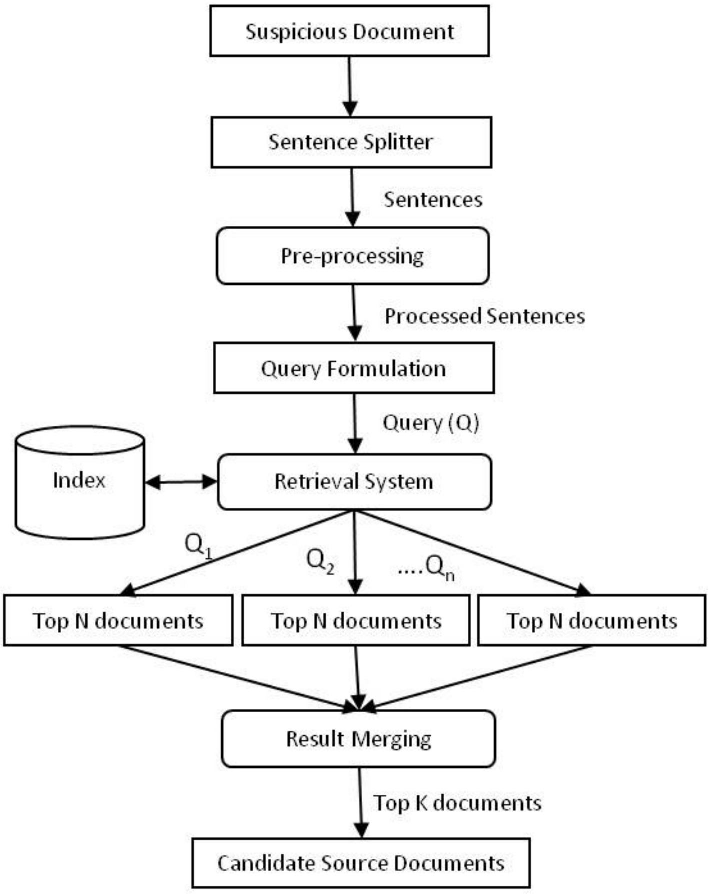

Figure 3.1: Process of candidate document retrieval

likely to be effectively detected by a short query. Therefore, the choice of query length is important to get good results.

3. **Retrieval:** Terms are weighted using the *tf.idf* weighting scheme (see Equation 2.4). Each query is used to retrieve relevant source documents from the source collection.

4. **Result Merging:** The top $N$ source documents from the result sets returned against multiple queries are merged to generate a final ranked list of source documents. In a list of source documents retrieved from a query, document(s) at the top of the list are likely to be the source(s) of plagiarism for that query. In

addition, portions of text from a single source document can be reused at different places in the same plagiarised document. Therefore, selecting only the top $N$ documents for each query in the result merging process is likely to lead to the original source document(s) appearing at the top of the final ranked list of the documents.

A standard data fusion approach called CombSUM method [Fox and Shaw, 1994] is used to generate the final ranked list of documents by combining the similarity scores of source documents retrieved against multiple queries. In the CombSUM method, the final similarity score, $S_{finalscore}$, is obtained by adding the similarity scores of source documents obtained from each query $q$:

$$S_{finalscore} = \sum_{q=1}^{N_q} S_q\left(d\right) \tag{3.1}$$

where $N_q$ is the total number of queries to be combined and $S_q\left(d\right)$ is the similarity score of a source document $d$ for a query $q$.

The top $K$ documents in the ranked list generated by the CombSUM method are marked as potential candidate source documents.

### 3.3.1 Implementation

Two popular and freely available Information Retrieval systems are used to implement the proposed IR-based framework: (1) Terrier [Ounis et al., 2005] and (2) Lucene [Hatcher et al., 2004]. Terrier is used to create indices of the PAN-PC-10 (Section 3.4.1.1) and the Extended Short Answer (Section 3.4.1.2) source collections and Lucene is used to index the MEDLINE source collection (Section 3.4.1.3).[1] In a source collection, documents are pre-processed by converting the text into lower case and removing all

[1] Two different IR systems were used for these experiments because an index of the MEDLINE source collection using Lucene was available. To save time and effort experiments for the MEDLINE corpus were carried out using the Lucene IR system.

non-alphanumeric characters. Stop words[1] are removed and stemming is carried out using the Porter Stemmer [Porter, 1980].

In both Terrier and Lucene, terms are weighted using the *tf.idf* weighting scheme. In Terrier, documents against a query term are matched using the TAAT (Term-At-A-Time) approach. Using this approach, each query term is matched against all posting lists to compute the similarity score. In Lucene, the similarity score between query and document vectors is computed using the cosine similarity measure (see Equation 2.1).

### 3.3.2 Comparison to Other IR-Based Approaches

In extrinsic plagiarism detection, a conventional IR-based approach has been used for retrieving candidate source documents by, for example, Vania and Adriani [2010] and Rao et al. [2011] (participants of the 2nd [Potthast et al., 2010b] and 3rd [Potthast et al., 2011] International Competition on Plagiarism Detection respectively). Using the proposed approach, the entire source collection is indexed using an IR system. A suspicious document is used as a query to retrieve a ranked list of potential source documents. The source document which appears in the top $k$ documents of the ranked list or whose similarity score is greater than a certain threshold is marked as a potential candidate document. The disadvantage of this approach is that if small portion of text is reused for plagiarism, it will be difficult to identify similarity between suspicious-source document pair by using the entire suspicious document as query. In addition, it is likely to identify topical similarity between documents instead of overlap for plagiarism detection.

Unconventional IR-based approaches have also been applied for extrinsic plagiarism detection. For example, the systems coming first [Kasprzak and Brandejs, 2010] and second [GuangZhou et al., 2010] in the 2nd International Competition on Plagiarism Detection [Potthast et al., 2010b] used a fingerprinting retrieval model for candidate

[1] NLTK [Loper and Bird, 2002] stop word list of 127 words in English was used.

document retrieval. Using their proposed approach, the entire source collection is broken into chunks of overlapping word 5-grams. Each chunk is hashed and indexed. Fingerprints of same length are also generated for the suspicious document. Each suspicious fingerprint is used as a query to retrieve potential source documents. If a suspicious-source document pair shares $k$ fingerprints then the source document is marked as potential candidate document. The system that came third in the competition [Muhr et al., 2010] also applied an unconventional IR-based approach. Using their proposed approach, each source document is converted to overlapping blocks of 40 tokens and indexed using an IR system. Queries are formed from the suspicious document by splitting it into blocks of the same length as that of the source document. Source blocks similar to the query are retrieved using various heuristics. The matching query-source block pairs are merged to generate long aligned passages. The disadvantage of such approaches is that it is difficult to apply them for practical plagiarism detection, for example identifying sources of plagiarism from the web.

IR-based approaches have also been applied for plagiarism detection by treating each passage in a document as a separate sub-document, for example Costa et al. [2010]. Using their proposed approach, each source document is split into sub-documents of 100 words with an overlap of 50 words. The entire collection of sub-documents is indexed. A suspicious document is also split into sub-documents of the same length. Each suspicious sub-document is used as a query to retrieve potential candidate sub-documents. Similarity between each suspicious-source sub-document pair is computed using cosine distance. Matching adjacent suspicious-source sub-document pairs are joined to generate longer aligned passages. Again, the problem with this type of approaches is that it is difficult to apply them in real situations for plagiarism detection.

The proposed IR-based approach (Section 3.3) has been used by the author in [Nawab et al., 2011, 2012b]. The key difference between the conventional IR-based approaches used by Vania and Adriani [2010] and Rao et al. [2011], and our proposed approach is that they use entire suspicious document as a query whereas our approach splits a suspicious document into queries and combines the results of multiple queries using the CombSUM method to generate a final ranked list of potential source documents. The proposed approach can be easily and efficiently applied to real world scenarios. In addition, query expansion can be incorporated into this IR-based approach to deal with cases of plagiarism created with paraphrasing (see Chapter 4).

## 3.4 Experimental Setup

The proposed IR-based approach is compared with a state-of-the-art approach, Kullback-Leibler Distance (see Section 2.7.1), for the candidate document retrieval task. This section presents the three benchmark datasets and evaluation measures used to compare the two approaches.

### 3.4.1 Datasets

Evaluation is carried out using three datasets: (1) PAN-PC-10 Corpus, (2) Extended Short Answer Corpus and (3) MEDLINE Corpus. These corpora are chosen because they are benchmarks, and contain a range of types of plagiarism and documents from different domains (see Section 2.12.5). This makes the evaluation task more challenging and realistic. This section describes the suspicious and source collections in each dataset used for these experiments.

#### 3.4.1.1 PAN-PC-10 Corpus

From the PAN-PC-10 Corpus [Potthast et al., 2010a] (see Section 2.12.1.4), 10,479 documents written in English are selected to form the source collection. Documents in the suspicious collection are plagiarised with artificial (automatically generated) and simulated (manually paraphrased) cases of plagiarism. In total, there are 1,644 plagiarised documents in the suspicious collection: none (artificial) = 411, low (artificial) = 411, high (artificial) = 411 and simulated = 411 (see Table 2.6 for examples of plagiarism with these types of obfuscation). This corpus contains only 411 suspicious plagiarised documents with simulated cases of plagiarism (and all of them are selected for experiments). Therefore, the same number of suspicious plagiarised documents (411) are also randomly selected for each type of artificial obfuscation including none, low and high.[1]

#### 3.4.1.2 Extended Short Answer Corpus

There are total 100 documents (57 plagiarised, 38 non-plagiarised and 5 source Wikipedia[2] articles) in the Short Answer Corpus [Clough and Stevenson, 2011] (see Section 2.12.4). These documents contain examples of simulated (manually created) plagiarism which are useful for evaluation but the corpus is too small for experiments on candidate document selection. The corpus is extended with examples from the Web to make the problem of candidate document selection more challenging.

The five questions (or learning tasks) used for the Short Answer Corpus (see Section 2.12.4) were used as queries for the Google search engine.[3] Against each query, the top 99 articles retrieved were stored. These were combined with the five original Wikipedia articles to create a source collection of 500 documents. The suspicious collection contains 57 plagiarised documents with different levels of obfuscation including: near copy (or none) = 19, low revision (or low) = 19 and heavy revision (or high) = 19. This corpus is referred as the Extended Short Answer Corpus.[4]

For these experiments, the entire Extended Short Answer Corpus is used.

[1] The entire suspicious collection of the PAN-PC-10 corpus contains 12,134 suspicious documents for the extrinsic plagiarism detection task (half plagiarised and half non-plagiarised)(see Section 2.12.1.4).

[2] http://www.wikipedia.org/

[3] http://www.google.co.uk/ Retrieved on: 20-07-2011

[4] This corpus can be freely downloaded from http://staffwww.dcs.shef.ac.uk/people/R.Nawab/ExtendedSAC.rar

#### 3.4.1.3 MEDLINE Corpus

For these experiments, 19,569,568 citations in the 2011 MEDLINE/PubMed Baseline Repository[1] form the source collection (see Section 2.12.2). The suspicious collection contains 260 citations that have been manually examined and verified as duplicates.[2] These citation pairs are selected because they have no shared author, which makes them potential cases of plagiarism [Errami et al., 2008].

### 3.4.2 Evaluation Measure

The goal of the candidate document retrieval task is to identify all the source document(s) for each suspicious document while returning as few non-source documents as possible. It is important for all source documents to be included in the top ranked documents returned by the system since otherwise they will not be identified during later stages of processing. Consequently, recall is more important than precision for this problem (see Section 2.13 for description of precision, recall and $F_1$ measures).

Averaged recall for the top $K$ documents is used as the evaluation measure for these experiments. The averaged recall measure first computes the recall score for each suspicious document and then takes the average. Given a set of $N$ suspicious documents, the averaged recall score is calculated as:

$$R_{avg} = \frac{1}{N}\sum_{i=1}^{N} R_i \tag{3.2}$$

where $R_i$ is the recall score of the $i_{th}$ suspicious document.

Figure 3.2 shows an example of calculating averaged recall score for the candidate document selection ($K = 5$). Sets of relevant and retrieved documents are represented by *Annotations* and *Detections* respectively (source documents which are identified are

[1] The 2011 MEDLINE/PubMed Baseline Repository was downloaded from http://mbr.nlm.nih.gov/ on 25-04-2011

[2] The citation pairs can be downloaded from the Deja vu database http://dejavu.vbi.vt.edu/dejavu/duplicate/ Last visited: 06-03-2012

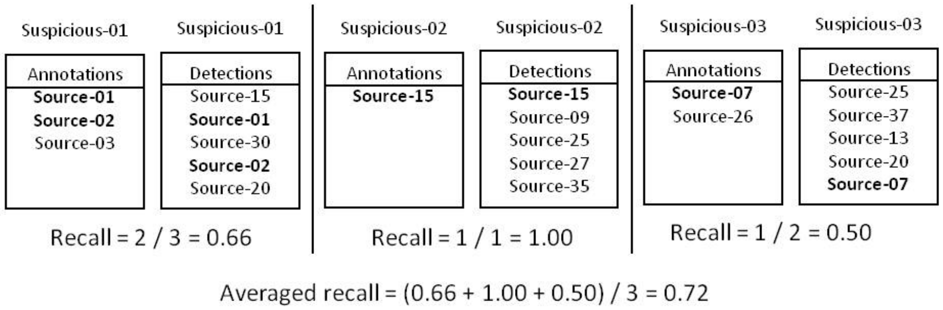

Figure 3.2: Example showing calculation of averaged recall score

in bold font). It can be noted from this example that the rank of a source document in the top $K$ documents is unimportant. As long as all the source documents appear in the top $K$ documents the averaged recall score will be 1, regardless of whether a source document appears in the first or $K_{th}$ rank.

### 3.4.3 Parameter Setting

The IR-based approach requires two parameters to be set: (1) the number of sentences used in formulating a query ($Q$) and the number of top $N$ retrieved documents used in the result merging process (see Section 3.3). The parameters ($Q$ and $N$) are set automatically using 3-fold cross validation (see results in Sections 3.5.1, 3.5.2 and 3.5.3). A suspicious collection is split into three folds with two being used to identify the optimal values for the parameters and the remaining third for evaluation. The results of the three runs are then averaged.

## 3.5 Results and Analysis

### 3.5.1 Results for PAN-PC-10 Corpus

Table 3.3 presents the results for the PAN-PC-10 corpus for various degrees of obfuscation: none, low, high and simulated. Averaged recall scores are reported for the top 5, 10, 15 and 20 source documents. Overall, it can be observed that the proposed approach

|  |  | Avg. Recall for top K documents |  |  |  |
|---|---|---|---|---|---|
| Obfuscation | Approach | 5 | 10 | 15 | 20 |
| None (artificial) | Kullback-Leibler | 0.2151 | 0.2496 | 0.2652 | 0.2781 |
|  | IR-based Approach | **0.6625** | **0.7621** | **0.7820** | **0.7980** |
| Low (artificial) | Kullback-Leibler | 0.2065 | 0.2348 | 0.2523 | 0.2623 |
|  | IR-based Approach | **0.6562** | **0.7602** | **0.7952** | **0.8145** |
| High (artificial) | Kullback-Leibler | 0.1897 | 0.2134 | 0.2331 | 0.2404 |
|  | IR-based Approach | **0.5879** | **0.6784** | **0.7170** | **0.7369** |
| Simulated (manual) | Kullback-Leibler | 0.1707 | 0.1975 | 0.2092 | 0.2178 |
|  | IR-based Approach | **0.5109** | **0.5758** | **0.6095** | **0.6274** |
| All | Kullback-Leibler | 0.1955 | 0.2238 | 0.2400 | 0.2497 |
|  | IR-based Approach | **0.6044** | **0.6941** | **0.7259** | **0.7442** |

Table 3.3: Performance for different types of obfuscation in the PAN-PC-10 corpus

outperforms the baseline approach (Kullback-Leibler Distance) by a large margin for all types of obfuscation. Improvement in performance is statistically significant (Wilcoxon signed-rank test, $p < 0.05$) [Wilcoxon et al., 1973].

The Kullback-Leibler Distance method (see Section 2.7.1) fails to give promising results for any level of obfuscation. A possible reason for low performance is that in the PAN-PC-10 corpus, small portions of plagiarised text are randomly inserted into documents to create suspicious plagiarised documents, so it becomes difficult to get high similarity scores between suspicious-source document pairs. Consequently, the original source document(s) do not appear in the top 20 candidate documents and low results are obtained.

Using the proposed approach, the lowest results are obtained for simulated obfuscation with top 20 candidate documents (0.6274) indicating the difficulty in detecting this type of obfuscation. The reason for very low performance in the PAN-PC-10 corpus is that small passages with simulated obfuscation were inserted randomly into suspicious documents, which made the retrieval task difficult. The average length of an inserted passage with simulated obfuscation was just 55 words and on average two passages were inserted into each document.

For $K = 20$, good averaged recall figures are obtained for low (artificial) (0.8145) and high (artificial) (0.7369) obfuscations highlighting the fact that IR-based approach is more robust to changes in the source and can detect paraphrased text. The difference in performance for detecting manual (simulated) and artificial (low and high) plagiarism demonstrates that human rewrites are more complex then those created by automated system.

For none (artificial) obfuscation, there is still room for improvement (0.7980). The most likely reason for low performance is that very small passages are randomly inserted verbatim into long suspicious plagiarised documents. During the result merging process of multiple queries the original source documents get a lower rank in the final ranked list of source documents. Consequently, original source documents do not appear in the top 20 documents and remain undetected.

### 3.5.2 Results for Extended Short Answer Corpus

Averaged recall scores for the top 1 to 5 candidate documents for the Extended Short Answer Corpus are reported in Table 3.4. The Extended Short Answer Corpus is smaller than the PAN-PC-10 corpus so results are reported for fewer of the top $K$ candidate documents identified. Results are shown for the various levels of obfuscation included in the corpus: none (near copy), low (light revision) and high (heavy revision).

As expected, retrieval performance decreases as the level of obfuscation increases. Both the IR-based approach and the Kullback-Leibler Distance method achieve 100% recall for the lowest level of obfuscation (none), indicating that detecting plagiarism in this corpus is straightforward when the text has not been modified. The two approaches also give promising results in identifying low obfuscation. However, the maximum recall achieved for high obfuscation is lower for all approaches, demonstrating that the problem is more difficult. Improvement in performance with the proposed approach is statistically significant for high obfuscation (Wilcoxon signed-rank test, $p < 0.05$).

| | | Avg. Recall for top K documents | | | | |
|---|---|---|---|---|---|---|
| Obfuscation | Approach | 1 | 2 | 3 | 4 | 5 |
| None | Kullback-Leibler | **0.9444** | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| | IR-based Approach | **0.9444** | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| Low | Kullback-Leibler | 0.5789 | **0.8421** | **0.9474** | 0.9474 | 0.9474 |
| | IR-based Approach | **0.6316** | **0.8421** | **0.9474** | **1.0000** | **1.0000** |
| High | Kullback-Leibler | 0.3684 | 0.4211 | 0.4737 | 0.6316 | 0.6316 |
| | IR-based Approach | **0.5789** | **0.6842** | **0.7368** | **0.7895** | **0.8947** |
| All | Kullback-Leibler | 0.6306 | 0.7544 | 0.8070 | 0.8597 | 0.8597 |
| | IR-based Approach | **0.7183** | **0.8421** | **0.8947** | **0.9298** | **0.9649** |

Table 3.4: Performance for different types of obfuscation in the Extended Short Answer Corpus

The Kullback-Leibler Distance performs well for none and low obfuscations but performance significantly decreases for high obfuscation indicating that the approach is not suited to the detection of plagiarism when the text has been heavily paraphrased. However, performance of the proposed IR-based approach is more robust to high level of paraphrasing.

It can also be noted from these results that both the proposed and baseline approaches perform well on this dataset as compared to their performance on the PAN-PC-10 corpus (see Table 3.3). A possible reason for good performance is that in this corpus, a portion of text from the original Wikipedia article is used to create the plagiarised documents with different levels of obfuscation. Therefore, similarity between suspicious-source document pairs is likely to be high and plagiarism is detected.

### 3.5.3 Results for MEDLINE Corpus

Two different source collections are used for experiments with the IR-based and Kullback-Leibler Distance approaches. For experiments with the IR-based approach, the entire MEDLINE collection of 19,569,568 citations (or documents) is used as the source collection. Since the Kullback-Leibler Distance method (see Section 2.7.1) is based on pairwise comparison of documents, it would be computationally very expensive to use

| | Avg. Recall for top K documents | | | | |
|---|---|---|---|---|---|
| Approach | 1 | 5 | 10 | 15 | 20 |
| Kullback-Leibler | 0.7596 | 0.8154 | 0.8442 | 0.8558 | 0.8596 |
| IR-based Approach | **0.8769** | **0.9173** | **0.9250** | **0.9288** | **0.9288** |

Table 3.5: Performance for the MEDLINE Corpus

the entire MEDLINE collection for experiments. Therefore, to make a fair comparison of the two approaches, a randomly selected subset of 3 million citations from MEDLINE is used as the source collection for experiments with the Kullback-Leibler Distance approach. This subset also contains the sources of plagiarised citations.

Table 3.5 shows the results for candidate document retrieval with the MEDLINE corpus. Results are reported for top 1, 5, 10, 15 and 20 candidate source documents. In contrast to the PAN-PC-10 results (see Table 3.3), averaged recall figures are also reported for the top document because using this corpus high recall is obtained for $K = 1$. Similar to the PAN-PC-10 and the Extended Short Answer results (see Tables 3.3 and 3.4 respectively), retrieval performance increases as the number of retrieved documents increases.

The IR-based approach performs better than the Kullback-Leibler Distance approach. Improvement in performance is statistically significant (Wilcoxon signed-rank test, $p < 0.05$). The highest recall achieved by the Kullback-Leibler Distance method is 0.8596 for top 20 candidate documents. The proposed approach achieves a recall of 0.8769 for $K = 1$, which is still higher than the maximum recall obtained using the Kullback-Leibler Distance method. This high recall score indicates the strength of the proposed method in detecting sources of plagiarism from large reference collections.

### 3.5.4 Exploring Thresholds

Sections 3.5.1, 3.5.2 and 3.5.3 presented the results for the three datasets (note that for these results parameters were set automatically using 3-fold cross validation (see

Section 3.4.3)). This section aims to explore the most appropriate features for plagiarism detection using the proposed IR-based approach. Experiments presented in this section are carried out using 10,479 source documents and 411 suspicious plagiarised documents obfuscated with cases of simulated plagiarism from the PAN-PC-10 corpus [Potthast et al., 2010a] (see Section 3.4.1.1).

Results are shown in Table 3.6. In this table, $Q$ is the number of sentences used to form a query and $N$ is the number of top retrieved documents for each query that are used in the result merging process. As can be seen from the results, the highest recall for the top 15 documents is obtained with $Q = 1$ and $N = 1$. However, for top 20 documents, the best results are obtained with $Q = 1$ and $N = 2$.

Overall, it is observed that as the size of query $Q$ or/and the number of retrieved top $N$ source documents per query increases the retrieval performance decreases (see Table 3.6). This indicates that a small passage (query) from a suspicious text can more efficiently identify the source of plagiarism compared to a long passage. Moreover, this also highlights that small passages of text are likely to be combined from multiple sources to make the plagiarism detection task more difficult instead of copying long chunks of text from a single source. The best result with low value of $N$ demonstrates that documents returned by non-plagiarised text influences the result merging process (based on score-based fusion approach), which ultimately affects the retrieval performance. Given a plagiarized text with high obfuscation, its similarity score with source text will be reduced. So when the value of $N$ is high, in the result merging process the non-relevant source documents come up on top of the ranked list and the original source document goes down in the list. As a consequence, the retrieval performance is decreased.

Note that same set of experiments is carried out for the Extended Short Answer Corpus, the MEDLINE Corpus and three levels of artificial obfuscation (none, low and high) in the PAN-PC-10 Corpus.

| Parameters | | Avg. Recall for top K documents | | | |
|---|---|---|---|---|---|
| Q | N | 5 | 10 | 15 | 20 |
| 1 | 1 | **0.5109** | **0.5758** | **0.6095** | 0.6273 |
| 1 | 2 | 0.4886 | 0.5576 | 0.5937 | **0.6281** |
| 1 | 3 | 0.4481 | 0.5357 | 0.5823 | 0.6042 |
| 1 | 4 | 0.4193 | 0.5101 | 0.5491 | 0.5941 |
| 1 | 5 | 0.4023 | 0.4878 | 0.5320 | 0.5653 |
| 2 | 5 | 0.3451 | 0.4282 | 0.4688 | 0.4919 |
| 3 | 5 | 0.3187 | 0.3852 | 0.4315 | 0.4866 |
| 4 | 5 | 0.2506 | 0.3378 | 0.3942 | 0.4412 |
| 5 | 5 | 0.2295 | 0.3175 | 0.3755 | 0.4051 |
| 1 | 10 | 0.3321 | 0.4071 | 0.4554 | 0.4862 |
| 2 | 10 | 0.2676 | 0.3447 | 0.3929 | 0.4234 |
| 3 | 10 | 0.2267 | 0.3220 | 0.3694 | 0.4015 |
| 4 | 10 | 0.2015 | 0.2498 | 0.3171 | 0.3451 |
| 5 | 10 | 0.1809 | 0.2466 | 0.2851 | 0.3191 |
| 1 | 15 | 0.2908 | 0.3686 | 0.4120 | 0.4363 |
| 2 | 15 | 0.2393 | 0.3033 | 0.3609 | 0.3917 |
| 3 | 15 | 0.1938 | 0.2595 | 0.3183 | 0.3585 |
| 4 | 15 | 0.1675 | 0.2125 | 0.2571 | 0.2981 |
| 5 | 15 | 0.1577 | 0.2113 | 0.2413 | 0.2741 |
| 1 | 20 | 0.2680 | 0.3325 | 0.3832 | 0.4100 |
| 2 | 20 | 0.2259 | 0.2794 | 0.3297 | 0.3585 |
| 3 | 20 | 0.1805 | 0.2267 | 0.2676 | 0.3074 |
| 4 | 20 | 0.1602 | 0.1967 | 0.2449 | 0.2766 |
| 5 | 20 | 0.1407 | 0.1869 | 0.2247 | 0.2534 |

Table 3.6: Results for 411 suspicious plagiarised documents containing only simulated obfuscation in the PAN-PC-10 corpus using the IR-based approach

Regarding optimal parameter values (see Section 3.4.3), for all three datasets, the best results are obtained using a single sentence as a query ($Q$). However, an optimal value for the number of source documents retrieved against each query ($N$) is different for different corpora. The optimal value for the PAN-PC-10 Corpus is $N = 1$; the Extended Short Answer Corpus is $N = 12$ and the MEDLINE Corpus is $N = 10$. This difference in optimal parameter value is likely to happen due to the different strategies used in creating these corpora. In the PAN-PC-10 corpus, plagiarised passages are randomly inserted into documents to create plagiarised documents, whereas in the other two corpora, an original piece of text is rewritten (or reused verbatim) to create the plagiarised text and the length of the plagiarised-source document pair is almost same.

### 3.5.5 Summary of Results

The proposed IR-based approach is compared with a state-of-the-art approach, Kullback-Leibler Distance, for the candidate document retrieval task. Results showed that it is relatively straightforward to detect verbatim (word to word copy) and slightly modified copies of reused text but more challenging to detect paraphrased text. In addition, detecting automatically paraphrased cases of text reuse is relatively easy but it is more difficult to detect manually paraphrased ones. This indicates that paraphrasing techniques used by humans in rewriting a piece of text are more sophisticated than automatic systems. It was also observed that detecting small passages of reused text inserted randomly into long suspicious documents is more difficult compared to identifying longer passages.

The proposed approach outperforms the baseline approach on all three datasets. The Kullback-Leibler Distance approach performs well at detecting reuse created with a low level of obfuscation but breaks down as the level of obfuscation increases. However, the IR-based framework proved to be more robust for detecting various levels of obfuscation particularly when the source text has been paraphrased.

## 3.6 Chapter Summary

This chapter presented an IR-based framework for the problem of candidate document retrieval. The proposed method was compared with a state-of-the-art method (Kullback-Leibler Distance) on three benchmark datasets including (1) PAN-PC-10 Corpus, (2) Extended Short Answer Corpus and (3) MEDLINE Corpus. Averaged recall score for the top $K$ candidate source documents was used as an evaluation measure. Results showed that the proposed method outperformed the baseline approach on all three datasets. Kullback-Leibler Distance method performs well when text has not been modified but breaks down for paraphrased text. However, the IR-based method proved to be more robust in detecting plagiarism cases created with paraphrasing.

# Chapter 4

# Improving the IR-based Approach with Query Expansion

## 4.1 Introduction

Chapter 3 presented an IR-based framework (see Section 3.3) for retrieving candidate documents. Promising results were obtained with the proposed framework in detecting various types of obfuscation (see Tables 3.3, 3.4 and 3.5). However, it was observed that there is still room for improvement particularly when the source text has been heavily paraphrased. To further improve the candidate retrieval performance two broad approaches for query expansion are explored [Baeza-Yates and Ribeiro-Neto, 2011]: (1) a pseudo relevance feedback method based on term co-occurrence statistics (see Section 4.4.1) and (2) query expansion using knowledge bases (see Section 4.4.2). In the former case, three different methods are investigated including (1) WordNet, a general purpose hand-crafted thesaurus (see Section 4.4.2.1), (2) a corpus-derived thesaurus generated using an automatic paraphrase generation system [Callison-Burch, 2008] (see Section 4.4.2.2) and (3) UMLS Metathesaurus, a thesaurus for biomedicine and related fields (see Section 4.4.2.3).

Barrón-Cedeño [2012] investigated different strategies for mono-lingual paraphrasing to identify the paraphrases which are most difficult to detect. They used simulated (manually paraphrased) cases of plagiarism in the PAN-PC-10 Corpus [Potthast et al., 2010a] for this study. Paraphrases were categorised into six main categories: (1) morphology-based changes, (2) lexicon-based changes, (3) syntax-based changes, (4) discourse-based changes, (5) semantics-based changes and (6) miscellaneous changes. Their analysis showed that lexical substitution is the most common editing operation used in paraphrasing for plagiarism and plagiarised text is a summarised version of the original text. Therefore, to capture the most common paraphrasing mechanisms for plagiarism, the content words of the document which is suspected to contain plagiarised text are expanded with synonymous words using different query expansion approaches.

The rest of this chapter is organised as follows: Section 4.2 gives an overview of relevance feedback and query expansion. Section 4.3 describes different thesauri used for query expansion. Section 4.4 describes how query expansion is applied for text reuse and plagiarism detection. Section 4.5 presents the datasets and evaluation measure. Finally, Section 4.6 discusses results of experiments.

## 4.2 Relevance Feedback and Query Expansion

In Information Retrieval (IR), a user presents his information need in the form of a query. A user might not use the same query term to express a concept that was used by the author of a document. This will create a vocabulary gap between the terms in the query and relevant documents. Since many IR systems are based on exact matching, the query terms will not match the document terms and relevant documents will not be retrieved. In addition, users often find it difficult to formulate a query which completely satisfies their information needs.

To avoid these problems, methods have been proposed to transform an initial query $q$ to a modified query $q_m$ (called query reformulation), which is more likely to be a better representation of user's information need. Previous studies have shown that query expansion is useful for improving the retrieval performance [Fang, 2008; Lu and Mu, 2009; Riezler et al., 2007].

Similar to the problem of vocabulary mismatch in IR, a vocabulary gap is also created between source and reused texts when an author tries to hide text reuse by replacing words/phrases in the source text with their appropriate synonymous words or phrases. In this situation, exact matching algorithms will fail to detect similarities between the source and reused text pair and text reuse may not be detected. A possible solution is to incorporate query expansion into existing approaches for text reuse (and plagiarism) detection.

A comprehensive discussion of all the methods for query reformulation is beyond the scope of this section (see Baeza-Yates and Ribeiro-Neto [2011] and Manning et al. [2008] for a detailed description on IR and query reformulation). Some of the most popular and commonly used methods, which are investigated in this research work are presented in the sections below.

### 4.2.1 Relevance Feedback

Relevance feedback is a well-established approach for query reformulation. The process of transforming an initial query $q$ to a modified query $q_m$ is carried out as follows [Manning et al., 2008]:

- A user submits an initial query $q$ to the IR system.
- The IR system returns an initial set of relevant documents $D_r$.
- The user annotates each document in $D_r$ as either relevant or irrelevant.

- The feedback information provided by the user on $D_r$ is used by the system to reformulate $q$ to $q_m$, which is likely to better represent the user's information need.

- The modified query $q_m$ is used by the system to return a final set of relevant documents.

The process of formulating a good modified query can take one or more iterations because the document collection is not known in advance.

Pseudo relevance feedback (also called *blind relevance feedback*) automates the manual part of the relevance feedback, which involves user tagging documents as either relevant or non-relevant to collect feedback information. Instead of asking the user to annotate the initial set of ranked documents retrieved against an initial query, the top $K$ ranked documents in the result set are assumed to be relevant and are used in the query reformulation process.

Regarding the retrieval performance, both precision and recall have been shown to improve using relevance feedback [Baeza-Yates and Ribeiro-Neto, 2011].

#### 4.2.1.1 Rocchio's Algorithm for Relevance Feedback

A popular and classical method for relevance feedback is Rocchio's framework [Rocchio, 1971]. Using this method, the feedback information is integrated into the well-known vector space model. For a given query $q$, two main assumptions underlying this method are: (1) term-weight vectors of the documents relevant to $q$ are similar to each other and (2) term-weight vectors of the non-relevant documents for $q$ are dissimilar to those of the relevant documents. The main goal is to generate a revised query $q_m$ such that it gets closer to the set of relevant documents and further away from the set of non-relevant documents in the vector space.

Ideally, information about the set of documents relevant to a query $q$ is known in advance. In such a situation, if $C_r$ and $C_{nr}$ represent the set of relevant and non-relevant documents respectively for $q$, then an optimal query vector $q_{opt}$ for distinguishing between relevant and non-relevant documents will be given by [Baeza-Yates and Ribeiro-Neto, 2011]:

$$\overrightarrow{q_{opt}} = \frac{1}{|C_r|} \sum_{\forall d_j \in C_r} \overrightarrow{d_j} - \frac{1}{|C_{nr}|} \sum_{\forall d_j \in C_{nr}} \overrightarrow{d_j} \tag{4.1}$$

where $d_j$ is a document and $\overrightarrow{d_j}$ is the weighted term vector associated to it. The optimal weighted term vector $q_{opt}$ is obtained by subtracting the centroids of non-relevant documents vectors from those of relevant ones.

In the majority of practical situations the documents relevant to $q$ will not be known in advance. To overcome this problem, partial information (documents tagged as relevant or non-relevant by a user) is utilised to compute the optimal query vector $q_{opt}$. If $D_r$ and $D_{nr}$ represent the set of relevant and non-relevant documents (obtained by user's relevance judgement) the modified query $q_m$ is calculated as [Baeza-Yates and Ribeiro-Neto, 2011]:

$$\overrightarrow{q_m} = \alpha \overrightarrow{q} + \frac{\beta}{|D_r|} \sum_{\forall d_j \in D_r} \overrightarrow{d_j} - \frac{\gamma}{|D_{nr}|} \sum_{\forall d_j \in D_{nr}} \overrightarrow{d_j} \tag{4.2}$$

where $q$ is the original query and $\alpha$, $\beta$, $\gamma$ are adjustable weights. This process can have multiple iterations to reformulate a good modified query $q_m$.

Rocchio's method adds similar/related expansion terms to the initial query. In addition, it gives more weight to some query terms and less to others. Both positive and negative feedback information can be obtained. However, most IR systems prefer positive information over the negative by setting $\gamma < \beta$ (negative information is mostly set to 0). A system ignores negative information by setting $\gamma = 0$ [Manning et al., 2008].

The two main strengths of this method are simplicity and improved results. It is simple because weights for modified query terms are calculated directly from the result set. Empirical investigations have shown that this method also improves both precision and recall [Baeza-Yates and Ribeiro-Neto, 2011].

### 4.2.2 Query Expansion

The second main approach for query reformulation is query expansion, the process of adding search terms to a query with the aim of improving retrieval performance. For instance, the query "car" could be expanded to "car cars automobile vehicle". The process of query expansion can be applied to an initial query, reformulated query or both. Moreover, the addition of expansion terms to original query terms can be combined with term reweighting. For example, expansion terms can be assigned less weight than original ones.

Various approaches have been proposed for query expansion, for example, automated techniques using pseudo relevance feedback or utilising knowledge bases (e.g. thesauri) and interactive techniques involving users in selecting the expansion terms [Efthimiadis, 1996]. Two main factors should be considered in generating a modified query with query expansion: (1) the source used for suggesting additional search terms and (2) the method used for selecting additional search terms. The source for creating an expanded query can be either the initial set of relevant documents returned by the system or knowledge bases (e.g. WordNet, UMLS Metathesaurus) [Efthimiadis, 1996].

The main challenge in this type of query reformulation is deciding how to generate expanded queries. The most common method for suggesting similar/related terms to create expanded queries is to use a thesaurus. The two main methods for generating a thesaurus are: (1) manual thesaurus generation and (2) automatic thesaurus generation [Manning et al., 2008]. The subsections below discuss each in detail.

#### 4.2.2.1 Manual Thesaurus Generation

The process of manual thesaurus generation is time consuming and requires human experts (or editors). To build reliable and high quality resources it is necessary that editors should be domain experts. A manual thesaurus for query expansion can be created using one of the following methods [Manning et al., 2008].

**Controlled Vocabulary** A controlled vocabulary can be defined as a set of authorised terms that are used to annotate pieces of information (documents, articles etc.) to improve the retrieval performance. Each concept in a controlled vocabulary is assigned a canonical term. A controlled vocabulary is designed and updated by human editors. Various domains have utilised controlled vocabularies to improve search results. For example, *Library of Congress Subject Headings* is a popular controlled vocabulary for organising library documents. Unified Medical Language System (UMLS) is another well-known example of controlled vocabularies. Its main goal is to improve performance in retrieving biomedical research articles stored in the MEDLINE database [Manning et al., 2008].

**Thesaurus** In constructing a thesaurus, human editors group a set of synonymous words (bearing the same meaning) into a single concept. Roget's thesaurus and WordNet are well-known examples of manual thesauri. Another multi-lingual and multi-purpose thesaurus is UMLS Metathesaurus [Manning et al., 2008].

The main advantages of the manual thesaurus-based query reformulation are: (1) it requires no input from users, (2) it is more accurate than a thesaurus generated automatically since resources used for generating a revised query are created by domain experts, (3) it generally increases recall and is more suitable for those applications that aim to improve recall and (4) it is widely used in many research fields, for example, biomedicine, computer science, engineering. However, manual thesaurus generation is expensive and time consuming. In addition, a thesaurus needs to be constantly updated to cope up with the developments in research and terminologies within a field [Manning et al., 2008].

#### 4.2.2.2 Automatic Thesaurus Generation

An alternative method for building a thesaurus is to generate it automatically. The most common and widely used method for building a thesaurus automatically is based on term co-occurrence statistics. In this approach, it is assumed that terms that co-occur in a paragraph or document are likely to be similar/related. Simple count statistics based on term co-occurrence can be used to identify similar terms [Grefenstette, 1994].

The main problem encountered in building such a thesaurus is determining how to establish the association between terms. Term ambiguity can very easily lead to noisy expansion terms. For example, an original query “Apple computers” may be expanded to “Apple green red fruit computers”.

## 4.3 Thesauri for Query Expansion

The previous section described various approaches for revising (or expanding) a query to improve retrieval performance. This section presents three thesauri (or knowledge bases) that are used for query expansion in this research work: (1) WordNet, a general-purpose database which is suitable for expanding keywords in a document written in English (see Section 4.3.1), (2) the Paraphrase Lexicon, a corpus-derived thesaurus, which enables expansion of keywords with single and multi-word paraphrases (see Section 4.3.2) and (3) the UMLS Metathesaurus, a thesaurus which is suitable for expanding terms in medical and health related documents (see Section 4.3.3). The following subsections describe these thesauri in more detail.

### 4.3.1 WordNet

WordNet [Miller et al., 1990] is a large hand-crafted lexical database for the English language.[1] WordNet has proved to be an effective resource in improving retrieval performance [Fang, 2008; Gong et al., 2006; Gonzalo et al., 1998; Liu et al., 2004]. The basic unit of WordNet is a synset (or sense) - a group of synonymous words or collocations[2] that can be replaced in a given context. Each synset represents a unique concept and belongs to one grammatical class: noun, verb, adjective or adverb. A short gloss (or definition) is associated with each synset.

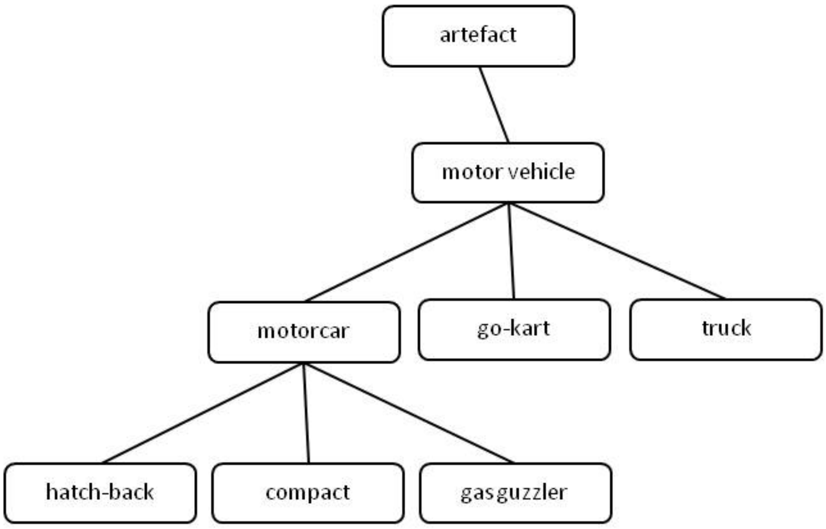

Figure 4.1: Fragment of WordNet Concept Hierarchy: nodes correspond to synsets; edges indicate the hypernym/hyponym relation, i.e. the relation between superordinate and subordinate concepts [Bird et al., 2009]

The majority of the synsets are connected to each other through different semantic relations including hypernym, hyponym, holonym and meronym. The hypernym/hyponym relations define the relationship between superordinate and subordinate (see Figure 4.1).

[1] There are now WordNets for several other languages.

[2] A collocation is a group of words which combine together to give a specific meaning, for example, “car park”.

In WordNet, a word can belong to one or more synsets. A word which belongs to multiple synsets (i.e. has several senses) is considered ambiguous. Normally, a word sense disambiguation algorithm is used to identify the most appropriate sense of an ambiguous word using its context. The senses of a word are ranked by their frequency in a sense tagged corpus.

### 4.3.2 Paraphrase Lexicon

An alternative resource for generating expanded queries is the Paraphrase Lexicon proposed by Callison-Burch [2008]. Paraphrases generated using this system have been used successfully for linguistic steganography [Chang and Clark, 2010] and machine translation [Callison-Burch et al., 2006]. Previously, paraphrases have also been used for query expansion and found to be useful for improving the performance of question answering systems [Riezler et al., 2007] and for automatic evaluation of summarization or translation systems [Kauchak and Barzilay, 2006].

| Word | Paraphrase | Probability Score |
|---|---|---|
| first | first and foremost | 0.0927 |
| first | very first | 0.0250 |
| first | particular | 0.0213 |
| first | first of all | 0.0192 |
| first | most important | 0.0125 |

Table 4.1: An example output of automatic paraphrase generation system [Callison-Burch, 2008] for the word "first"

The paraphrase generation system [Callison-Burch, 2008] automatically extracts paraphrases from Europarl Corpus [Koehn, 2005] (which contains parallel documents from the European Parliament). To improve the quality of paraphrases, complex syntactic labels are used so that a phrase and its paraphrases have the same syntactic type. For paraphrase generation, the system parses the English side of a parallel corpus and extracts phrase labels alongside bilingual phrase pairs. If English phrases share a common foreign language word/phrase among their possible translations and have the same syntactic type then they are assumed to be potential paraphrases of each other. Multiple paraphrases can be extracted for a phrase and are ranked using score based on phrase translation probability, which is computed using maximum likelihood estimation (see Table 4.1 for an example output generated using the automatic paraphrase generation system).[1]

### 4.3.3 Unified Medical Language System (UMLS)

WordNet and Paraphrase Lexicon methods are suitable for creating expanded queries for the general English text. However, one of the evaluation corpus (MEDLINE Corpus; see Section 2.12.2) contains biomedical research literature. To generate appropriate expanded queries for the MEDLINE Corpus, synonymous terms from the UMLS Metathesaurus are used.

A huge amount of online literature is available on biomedical sciences and is increasing day by day. The large repositories of biomedical data makes it difficult to efficiently retrieve relevant documents against a query. The Unified Medical Language System (UMLS)[2] aims to assist in developing computer systems that can effectively process and search text related to health and biomedicine. The National Library of Medicine (NLM)[3] periodically updates and freely distributes the UMLS knowledge sources and a set of associated software tools.

The main component of UMLS is the Metathesaurus. The commonly used supporting software tool is MetaMap. The following subsections briefly discuss them. For more detailed and in-depth information on UMLS and its supporting software tools, see the online *UMLS Reference Manual*.[4]

[1] The software for automatic paraphrase generation can be freely downloaded from http://www.cs.jhu.edu/~ccb/howto-extract-paraphrases.html Last visited: 31-05-2012

[2] http://www.nlm.nih.gov/research/umls/ Last visited: 31-05-2012

[3] http://www.nlm.nih.gov/ Last visited: 31-05-2012

[4] http://www.ncbi.nlm.nih.gov/books/NBK9676/ Last visited: 31-05-2012

#### 4.3.3.1 UMLS Metathesaurus

The Metathesaurus is the main component of UMLS. It is a large database of more than 100 multi-lingual controlled source vocabularies and classifications, which contain information about concepts (related to biomedical and health), concept names and relationships between concepts. Although it is a multi-purpose resource, it can be customised for specific applications/purposes using software tools. Query expansion using UMLS Metathesaurus has proved to be useful in improving retrieval performance for Healthcare Information Retrieval systems [Aronson and Rindflesch, 1997; Lu and Mu, 2009].

The basic unit of the Metathesaurus is the concept. The same concept can be referred to using different terms. One of the main goal of Metathesaurus is to group all the equivalent terms from different source vocabularies into a single concept. Thus, a concept is a collection of synonym terms (similar to synset/sense in WordNet). Each concept in Metathesaurus is assigned a unique identifier called a CUI (Concept Unique Identifier).

UMLS contains a set of tables (or files). The information about concept names, key features associated to each concept name (e.g. language, name type, source vocabulary) and concept identifiers is stored in the *MRCONSO* table. It contains information in multiple languages.

Table 4.2 shows some entries in the *MRCONSO* table in English for the term "`Gamma-glutamyl transpeptidase`", whose CUI is `C0202035`. The synonymous terms, "`Gamma glutamyl transpeptidase measurement`", "`GTP measurement`" and "`Gamma glutamyl transferase measurement`" can be used as expansion terms to expand the original term "`Gamma-glutamyl transpeptidase`".

| **Input Term** |
|---|
| `Gamma-glutamyl transpeptidase` |
| **MRCONSO Table Entries in English for the CUI** `C0202035` |
| `C0202035` ENG `Gamma glutamyl transferase measurement` |
| `C0202035` ENG `Gamma glutamyl transpeptidase measurement` |
| `C0202035` ENG `GTP measurement` |

Table 4.2: Example showing some of the MRCONSO table entries in English for the term "`Gamma-glutamyl transpeptidase`", whose CUI is `C0202035`. "ENG" means that entry is in English language. Note that each MRCONSO table entry contains other information as well but for simplicity only relevant information is presented.

#### 4.3.3.2 MetaMap

MetaMap is a key supporting tool for UMLS. The objective of this tool is to efficiently link terms mentioned in input text to concepts in UMLS Metathesaurus.

MetaMap performs syntactic/lexical analysis of the input text to map Metathesaurus concepts to input terms. The mapping process is described as follows [Aronson and Lang, 2010]:

- tokenisation, sentence boundary detection and identification of abbreviation/acronym;
- Part Of Speech (POS) tagging;
- input terms are looked up in the SPECIALIST lexicon;[1]
- finally, shallow parsing is carried out with the SPECIALIST minimal commitment parser to identify phrases and their lexical heads.

After the identification of phrases, further analysis of each phrase is carried out in the following steps.

- Variant generation: for all phrases, variants are generated using table lookup.

[1] The SPECIALIST lexicon is a lexicon of general English and biomedical terms. It stores morphological, syntactic and orthographic information about each term.

| **Input Term** |
|---|
| Gamma-glutamyl transpeptidase |
| **MetaMap Output without WSD** |
| Meta Mapping:<br>C0202035:Gamma-glutamyl transpeptidase<br>Meta Mapping:<br>C0017040:gamma glutamyl transpeptidase |
| **MetaMap Output with WSD** |
| Meta Mapping:<br>C0202035:Gamma-glutamyl transpeptidase |

Table 4.3: Simplified example output from MetaMap with and without Word Sense Disambiguation (WSD) being applied. In each entry "C0202035:Gamma-glutamyl transpeptidase", "C0202035" represents the UMLS CUI to which the input term "Gamma-glutamyl transpeptidase" is mapped by MetaMap.

- Candidate identification: Metathesaurus strings (known as *candidates*) are generated by matching phrase text to input text. These *candidates* are treated as intermediate results and their matching with input text is also evaluated.

- Mapping construction: final mappings are obtained by combining all the *candidates* identified in the previous step and evaluating them on the basis of their matching with the phrase text (see *Meta Mappings* generated with "MetaMap Output without WSD" in Table 4.3).

- Word Sense Disambiguation (WSD) (optional): MetaMap also includes the option of carrying out WSD to attempt to select between candidates when there are multiple possible CUIs for a term [Humphrey et al., 2006] (see *Meta Mappings* generated with "MetaMap Output with WSD" in Table 4.3).

Table 4.3 shows simplified example output generated by MetaMap with and without WSD.[1] It can be noted that there are two *Meta Mappings* when WSD option is not used (similar to *all senses* in WordNet), while there is only one *Meta Mapping* with WSD

[1] MetaMap output also contains other information. To make it easier for the reader to understand, the simplified output is presented.

(similar to *sense selection after word sense disambiguation* in WordNet). Also, during parsing, MetaMap treats the phrase “Gamma-glutamyl transpeptidase” as a single term instead of treating it as two separate terms: “Gamma-glutamyl” and “transpeptidase”. MetaMap treats many multi-word phrases as single terms.

## 4.4 Applying Query Expansion for Text Reuse and Plagiarism Detection

Section 4.2 described approaches for query reformulation and Section 4.3 presented three knowledge bases that are used for query expansion in this work. This section describes how query expansion approaches are applied to deal with cases of text reuse in which an original text has been paraphrased to create the reused one. The two widely used query expansion approaches investigated to add additional search terms to an original query term are: (1) pseudo relevance feedback and (2) query expansion with knowledge bases.

For the experiments presented in this chapter, only content words of the document which is suspected to contain reused text (suspicious document) are expanded in the query expansion process (stop words and numbers are ignored).

### 4.4.1 Pseudo Relevance Feedback

This is a popular and widely explored query expansion method based on term co-occurrence statistics (see Section 4.2.1) [Baeza-Yates and Ribeiro-Neto, 2011; Bai et al., 2005; Peat and Willett, 1991; Qiu and Frei, 1993; Smeaton and Van-Rijsbergen, 1983]. There are three key issues in this approach: (1) how many top ranked documents should be used for identifying similar/related terms, (2) how many terms should be used as expanded terms and (3) what weight should be assigned to expanded terms.

#### 4.4.1.1 Implementation

Experiments using pseudo relevance feedback are carried out using the Terrier IR system [Ounis et al., 2005], which provides state-of-the-art automatic pseudo relevance feedback method for query expansion based on a new Divergence From Randomness (DFR) framework [Amati and Van Rijsbergen, 2002] (see Table 4.4 for an example of query expansion using this approach). Two parameters can be set for query expansion: (1) number of documents to be used for extracting expansion terms and (2) number of expansion terms to be extracted from top ranked documents. For assigning weights to expanded terms a DFR term weighting model is applied.

During the query expansion process, the most informative terms are selected as expansion terms from top-returned documents. There are different DFR term weighting models which can be used to assign appropriate weights to the expansion terms and one such model is the $Bo1$ weighting model (the default in Terrier). It computes the weight of a term $t$ in a set of top ranked documents using the following equation:

$$w\left(t\right) = tf_x \cdot log_2 \frac{1 + P_n}{P_n} + log_2\left(1 + P_n\right) \tag{4.3}$$

where $tf_x$ defines a query term's frequency in a set of top ranked documents. $P_n$ is defined by $\frac{F}{N}$, where F represents a term's frequency in the corpus and N is the total number of documents in the corpus.

### 4.4.2 Query Expansion with Knowledge Bases

This section describes how the three knowledge bases, WordNet (see Section 4.3.1), Paraphrase Lexicon (see Section 4.3.2) and UMLS Metathesaurus (see Section 4.3.3.1) are applied for query expansion.

#### 4.4.2.1 Query Expansion using WordNet

For each query term in a suspicious document, WordNet is consulted to identify the synsets in which it occurs and additional search terms selected from them. Some query terms occur in more than one synset and are considered to be ambiguous. In such cases, the process of word sense disambiguation (WSD) is applied to identify the most appropriate sense. Normally, a WSD algorithm utilizes the information from the surrounding context of the target query term to identify the most suitable sense. Three different approaches are explored for identifying additional search terms: (1) *first sense*, (2) *all senses* and (3) *sense selection after word sense disambiguation*.

In the *first sense* approach, additional search terms are selected only from the first synset containing the query term. In the *all senses* approach, additional search terms can be selected from any of the synsets containing the query term. In the sense selection after word sense disambiguation approach, additional search terms are chosen from the synset that is selected after disambiguation (see Table 4.4 for examples of query expansion using these three methods).

After the synset(s) have been identified from WordNet the additional search terms have to be selected from them. All terms in the synset(s) are ranked based on their frequency in the British National Corpus[1] (BNC) and the highest ranked term(s) used as additional search term(s).

**Word Sense Disambiguation (WSD)**

The process of identifying the correct sense of a word in a given context when that word has multiple senses is called Word Sense Disambiguation. Agirre and Soroa [2009] proposed a graph-based unsupervised word sense disambiguation algorithm. The proposed system first creates a Lexical Knowledge Base (LKB) using WordNet. The content words (noun, verbs, adjectives and adverbs) from an input sentence are represented

[1] *BNC frequency list for all the words* http://www.kilgarriff.co.uk/bnc-readme.html was used for experiments.

as nodes of a graph using the knowledge in the LKB. A target word is disambiguated by applying the personalized page rank algorithm to rank the vertices of the graph. Results showed that this system achieved a recall of 58.6 and 57.4 on Senseval-2 and Senseval-3 all words datasets respectively and outperformed previous approaches for WSD [Agirre and Soroa, 2009]. Therefore, this system is used to disambiguate the content words in suspicious documents.[1]

For these experiments, the process of WSD begins by splitting a suspicious document into sentences. A POS tag is associated to each word in a sentence using the NLTK POS tagger [Loper and Bird, 2002]. Each word in a sentence is lemmatized using a WordNet lemmatizer and stop words are removed. The WSD system is used to determine the possible correct sense of each content word in the sentence (see Table 4.4 for an example of query expansion using this method).

#### 4.4.2.2 Query Expansion using a Paraphrase Lexicon

The suspicious documents in a corpus are used to create a Paraphrase Lexicon of *paraphrases* or *lexical equivalents* in two steps. In the first step, a list of unique keywords (stop words and numbers are ignored) is generated using the entire suspicious collection. In the second step, the list of phrases (or keywords) is given as input to the automatic paraphrase generation system [Callison-Burch, 2008] which outputs a Paraphrase Lexicon. Each entry in the Paraphrase Lexicon is of the form: *word, paraphrase or lexical equivalent, probability score* (see Table  4.1 for an example output generated by the automatic paraphrase generation system).

The Paraphrase Lexicon is used for query expansion by ranking the possible paraphrases of a query term based on their probability score and the highest ranked one used as an additional search term (see Table 4.4 for an example of query expansion using this method).

[1] The software for Word Sense Disambiguation can be freely downloaded from http://ixa2.si.ehu.es/ukb/ Last visited: 05-07-2012

#### 4.4.2.3 Query Expansion using UMLS Metathesaurus

For query expansion, MetaMap is used to map terms to their corresponding CUIs in UMLS Metathesaurus (see Section 4.3.3.2). The UMLS Metathesaurus's *MRCONSO* table is consulted to extract expansion terms for each CUI (see Section 4.3.3.1).

An input term is mapped to UMLS CUIs in two ways: (1) CUI mapping with WSD and (2) CUI mapping without WSD. In the former case, synonym terms for query expansion are selected using the CUI which is selected after applying WSD, whereas in the latter case, additional search terms can be selected using any of the mapped CUI(s).

A suitable resource to rank the synonymous terms extracted from UMLS was not found. Therefore, each input term is expanded with 1 randomly selected additional search term (see Table 4.5 for examples of query expansion using these approaches).

### 4.4.3 Examples of Query Expansion

Table 4.4 shows examples of expanded queries created using different query expansion methods based on pseudo relevance feedback, WordNet and Paraphrase Lexicon (where $w$ is the weight assigned to an additional search term). If the sense of a word in WordNet only contains the word itself then that word is not expanded. For example, in the *first sense* approach, the query word “century” is not expanded for this reason. It can be noted from these examples that additional search terms extracted from WordNet and Paraphrase Lexicon are more appropriate then the ones generated using the pseudo relevance feedback method.

Table 4.5 shows examples of expanded queries created using the UMLS Metathesaurus. An additional search term is added to a query term in two ways: (1) treating multi-word input and expansion terms as phrases (see examples of *WSD Phrase* and *Without-WSD Phrase*) and (2) treating multi-word input and expansion terms as a sequence of single words (see examples of *WSD* and *Without-WSD*).

| Query Sentence | first published century magazine |
|---|---|
| Pseudo Relevance Feedback | first published century magazine evans^$w$ philadelphia^$w$ biography^$w$ letters^$w$ |
| First Sense (WordNet) | first number^$w$ one^$w$ published print^$w$ century magazine mag^$w$ |
| All Senses (WordNet) | first low^$w$ published issue^$w$ century hundred^$w$ magazine cartridge^$w$ |
| WSD (WordNet) | first published write^$w$ century magazine mag^$w$ |
| Paraphrase Lexicon | first first^$w$ and^$w$ foremost^$w$ published advertised^$w$ century cooperation^$w$ magazine journal^$w$ |

Table 4.4: Examples of expanded queries using pseudo relevance feedback, WordNet and paraphrase lexicon ($w$ is the weight assigned to an additional search term)

| Query Sentence | hbf correlated total hemoglobin concentration |
|---|---|
| WSD | hbf fetal^$w$ hemoglobin^$w$ correlated correlation^$w$ total hemoglobin concentration finding^$w$ of^$w$ hemoglobin^$w$ concentration^$w$ |
| Without-WSD | hbf foetal^$w$ hemoglobin^$w$ correlated correlation^$w$ total of^$w$ total^$w$ hemoglobin concentration finding^$w$ of^$w$ hemoglobin^$w$ concentration^$w$ |
| WSD Phrase | hbf ‘‘fetal hemoglobin’’^$w$ correlated ‘‘correlation’’^$w$ total ‘‘hemoglobin concentration’’ ‘‘finding of hemoglobin concentration’’^$w$ |
| Without-WSD Phrase | hbf ‘‘foetal hemoglobin’’^$w$ correlated ‘‘correlation’’^$w$ total ‘‘of total’’ ‘‘hemoglobin concentration’’ ‘‘finding of hemoglobin concentration’’^$w$ |

Table 4.5: Examples of expanded queries using UMLS Metathesaurus ($w$ is the weight assigned to an additional search term)

## 4.5 Experimental Setup

### 4.5.1 Datasets

Three datasets used for evaluating IR-based approach in the previous chapter are also used for the evaluation of query expansion approaches including (1) PAN-PC-10 Corpus (see Section 3.4.1.1), (2) Extended Short Answer Corpus (see Section 3.4.1.2) and (3) MEDLINE Corpus (see Section 3.4.1.3).

Only the 411 documents obfuscated with simulated plagiarism cases are used for the experiments using the PAN-PC-10 Corpus. The examples of none, low and high obfuscations are created automatically by randomly altering text and inserting synonyms from WordNet. These automatic examples are not used in the query expansion experiments since the types of obfuscation they contain are different from those that would be observed in real cases of plagiarism and the fact that they are created using WordNet would be an unfair advantage to some of our query expansion approaches.

For the PAN-PC-10 and Short Answer corpora, each keyword from a suspicious document is expanded with 1, 2 and 3 additional search terms using WordNet (see Sections 4.4.2.1) and Paraphrase Lexicon (see Section 4.4.2.2). However, for the MEDLINE Corpus, each query term in a suspicious MEDLINE citation is expanded with 1 additional search term (see Section 4.4.2.3). For all three corpora, additional search terms are assigned weight from the range: {1.0, 0.5, 0.1, 0.05, 0.01}. This range of weights is selected to evaluate the affect of assigning equal, small and very small weights to the expansion terms.

### 4.5.2 Evaluation Measure

The evaluation measure, averaged recall for top $K$ documents (see Section 3.4.2), which was used to evaluate the performance of the IR-based approach (see Chapter 3) is also used to evaluate the query expansion approaches.

### 4.5.3 Parameter Setting

The proposed query expansion approaches require various parameters to be set, including the number of additional search terms ($S$), the weights assigned to the expansion terms ($W$), the number of top-returned documents to be used for extracting expansion terms ($D$) and the number of expansion terms to be extracted from top ranked documents ($T$). Similar to the experiments with the IR-based approach, these parameters are set automatically using 3-fold cross validation (see Section 3.4.3).

## 4.6 Results and Analysis

This section presents the experiments that are carried out to explore the effect of query expansion on candidate document retrieval performance. Note that, for all the experiments with query expansion, the values of parameters $Q$ (the number of sentences used to make a query) and $N$ (the number of top ranked documents used in result merging process) are fixed to their optimal values. On all three corpora, the optimal value of $Q$ was 1, whereas for the PAN-PC-10 Corpus, MEDLINE Corpus and Extended Short Answer Corpus the optimal values for $N$ were 1, 10 and 12 respectively (see Section 3.5.4).

### 4.6.1 Results for PAN-PC-10 Corpus

Table 4.6 shows the results for simulated examples of plagiarism in the PAN-PC-10 corpus using different query expansion approaches. Averaged recall figures are reported for the top 5, 10, 15 and 20 documents. The IR-based approach without query expansion is the approach used in the previous chapter ("IR-based Approach" in Table 4.6). Overall, results show that performance with the proposed IR-based framework can be improved using query expansion.

As expected, the averaged recall figure increases as the number of retrieved documents increases. Query expansion based on WordNet and Paraphrase Lexicon improves retrieval performance. For query expansion based on WordNet similar performance is observed for the *first sense* and *all senses* approaches, indicating that expansion terms selected using these methods are effective in improving performance. The best results

|  |  | Avg. Recall for top K documents |  |  |  |
|---|---|---|---|---|---|
| Obfuscation | Approach | 5 | 10 | 15 | 20 |
| Simulated | IR-based Approach | 0.5109 | 0.5758 | 0.6095 | 0.6274 |
|  | Pseudo Relevance | 0.4497 | 0.5182 | 0.5673 | 0.5969 |
|  | First Sense (WordNet) | 0.5158 | 0.5852 | 0.6277 | 0.6472 |
|  | All Senses (WordNet) | **0.5215** | 0.5851 | 0.6249 | 0.6456 |
|  | WSD (WordNet) | 0.5125 | 0.5710 | 0.5925 | 0.6188 |
|  | Paraphrase Lexicon | 0.5211 | **0.6062** | **0.6359** | **0.6602** |

Table 4.6: Performance with 411 simulated plagiarised documents from the PAN-PC-10 Corpus

are obtained using the Paraphrase Lexicon, except for average recall for the top 5 documents. Although improvement in performance is small, the differences are statistically significant (Wilcoxon signed-rank test, $p < 0.05$) [Wilcoxon et al., 1973] compared to the next highest result. In the case of average recall at 5 documents, the difference between using all senses and the Paraphrase Lexicon is not significant. Improvement with query expansion based on knowledge bases highlights the fact that words have been substituted with lexical equivalents to hide plagiarism.

Query expansion based on pseudo relevance feedback does not improve performance. The reason is that expansion terms are selected automatically which could result in the selection of noisy expansion terms. Query expansion based on WSD(WordNet) does not help to improve performance. A possible reason for low performance is that the inappropriate assignment of sense by the automatic WSD system [Agirre and Soroa, 2009] adds noise in the form of expansion words to an original word because the PAN-PC-10 Corpus was created using documents on English literature from the Project Gutenberg[1] (see Section 2.12.1.1).

[1] http://www.gutenberg.org/ Last Visited: 31-05-2012

| | | Avg. Recall for top K documents | | | | |
|---|---|---|---|---|---|---|
| Obfuscation | Approach | 1 | 2 | 3 | 4 | 5 |
| None | IR-based Approach | **0.9444** | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| | Pseudo Relevance | 0.5263 | 0.6842 | 0.7368 | 0.7368 | 0.8421 |
| | First Sense (WordNet) | 0.8889 | 0.9444 | 0.9444 | **1.0000** | **1.0000** |
| | All Senses (WordNet) | 0.7778 | 0.9444 | 0.9444 | **1.0000** | **1.0000** |
| | WSD (WordNet) | 0.8889 | 0.9444 | 0.9444 | **1.0000** | **1.0000** |
| | Paraphrase Lexicon | 0.8889 | 0.9444 | 0.9444 | 0.9444 | **1.0000** |
| Low | IR-based Approach | **0.6316** | 0.8421 | 0.9474 | **1.0000** | **1.0000** |
| | Pseudo Relevance | 0.5263 | 0.6842 | 0.7368 | 0.7368 | 0.8421 |
| | First Sense (WordNet) | 0.5789 | 0.6316 | 0.8947 | 0.8947 | 0.9474 |
| | All Senses (WordNet) | 0.3684 | 0.8947 | 0.8947 | 0.9474 | **1.0000** |
| | WSD (WordNet) | 0.5789 | 0.8421 | 0.8947 | 0.8947 | 0.9474 |
| | Paraphrase Lexicon | 0.4211 | **0.9474** | **1.0000** | **1.0000** | **1.0000** |
| High | IR-based Approach | **0.5789** | 0.6842 | 0.7368 | 0.7895 | 0.8947 |
| | Pseudo Relevance | 0.4737 | 0.5263 | 0.6316 | 0.6842 | 0.6842 |
| | First Sense (WordNet) | 0.3684 | 0.6316 | 0.7368 | 0.7895 | **0.9474** |
| | All Senses (WordNet) | 0.2105 | 0.6316 | 0.6842 | 0.8421 | **0.9474** |
| | WSD (WordNet) | 0.4211 | 0.6842 | 0.7895 | 0.8421 | **0.9474** |
| | Paraphrase Lexicon | 0.4211 | **0.7368** | **0.8421** | **0.8947** | **0.9474** |

Table 4.7: Performance for different types of obfuscation in the Extended Short Answer Corpus

### 4.6.2 Results for Extended Short Answer Corpus

Averaged recall scores for the top 1 to 5 candidate documents for the Extended Short Answer Corpus are reported in Table 4.7. The Extended Short Answer Corpus is smaller than the PAN-PC-10 Corpus so results are reported for fewer of the candidate documents identified. Results are shown for the various levels of obfuscation included in the corpus: none (or near copy), low (or light revision) and high (or heavy revision).

Similar to the results with the PAN-PC-10 Corpus, pseudo relevance feedback does not improve the performance for any of the three levels of obfuscation. This demonstrates that query expansion based on term co-occurrence statistics does not help in improving retrieval performance.

The effect of the WordNet and Paraphrase Lexicon methods for query expansion depends on the level of obfuscation. When the text has not been rewritten (none) performance actually drops for the first few documents retrieved when these approaches are applied. However, query expansion approaches based on knowledge bases achieves a 100% recall for $K = 5$, but this level of recall is reached more quickly when query expansion is not used.

Query expansion based on knowledge bases is more effective in improving performance when texts have been modified (low and high). For low obfuscation, query expansion using Paraphrase Lexicon achieves a recall of 100% for $K = 3$, demonstrating the usefulness of query expansion in detecting modified text. For high obfuscation, none of the approaches achieves a 100% recall, indicating that it is hard to detect heavily paraphrased text. However, the best results for this type of obfuscation are achieved using query expansion. In particular, the best results are achieved using the Paraphrase Lexicon for all but the first retrieved document. The improvement achieved using this approach compared with when no query expansion is used is statistically significant (Wilcoxon signed-rank test, $p < 0.05$) [Wilcoxon et al., 1973].

### 4.6.3 Results for MEDLINE Corpus

Table 4.8 shows the results for the MEDLINE Corpus. Averaged recall figures are reported for the top 1, 5, 10, 15 and 20 candidate documents. Results are presented for the top document because promising results are obtained for this value of $K$ (see Section 3.3). The IR-based approach without query expansion performs well (0.8769 for $K = 1$), however, performance further improves when query expansion is applied (0.9219 for $K = 1$). Improvement in performance is statistically significant for all query expansion approaches (Wilcoxon signed-rank test, $p < 0.05$) [Wilcoxon et al., 1973].

The best results are obtained when the input and expansion terms are used as phrases in the query expansion process (“WSD Phrase” and “Without-WSD Phrase” approaches in Table 4.8). A possible reason for this is that in the biomedical text there are many multi-word phrases which are treated as a single term (see Section 4.4.3). When similarity is computed between a query term and a source document, high sim-

| | Avg. Recall for top K documents | | | | |
|---|---|---|---|---|---|
| Approach | 1 | 5 | 10 | 15 | 20 |
| IR-based Approach | 0.8769 | 0.9173 | 0.9250 | 0.9288 | 0.9288 |
| WSD | 0.9077 | 0.9519 | 0.9558 | 0.9558 | 0.9596 |
| Without-WSD | 0.9035 | 0.9519 | 0.9519 | 0.9558 | 0.9558 |
| WSD Phrase | **0.9219** | **0.9595** | 0.9595 | **0.9652** | 0.9652 |
| Without-WSD Phrase | 0.9115 | 0.9558 | **0.9596** | 0.9634 | **0.9673** |

Table 4.8: Performance with the MEDLINE Corpus

ilarity scores are obtained for matching phrases. Consequently, the source of the plagiarised document is detected.

Expansion term with the "WSD" approach is selected from a smaller set of terms compared to the "Without-WSD" approach and is expected to give better results. However, there is not much difference in performance between the two approaches. A possible reason is that the inappropriate assignment of sense by the automatic WSD system (used by MetaMap for WSD) [Humphrey et al., 2006] adds noisy expansion terms and performance does not improve.

### 4.6.4 Query by Query Analysis

Analysis was carried out to find out the percentage of queries for which the rank of a correctly identified source document is "higher", "lower" or remains "same" when a query expansion approach is applied as compared to no query expansion approach (see Table 4.9). The rank of a query (suspicious document) is considered in the top 5 for the Extended Short Answer Corpus and the top 20 documents for the PAN-PC-10 and the MEDLINE corpora.

On large dataset (PAN-PC-10 Corpus), the percentage of queries with "lower" rank is at a minimum for the Paraphrasing Lexicon (13.38%) and a maximum for pseudo relevance feedback (38.93%). This shows that expansion with lexical equivalents is more suitable than other query expansion approaches. In the Extended Short Answer

|  |  | No. of Queries (%) effecting Rank |  |  |
|---|---|---|---|---|
| Corpus | Approach | Higher | Lower | Same |
| PAN-PC-10 | Pseudo Relevance | 56 (13.62) | 160 (38.93) | 130 (31.63) |
|  | First Sense | 69 (16.79) | 68 (16.54) | 211 (51.34) |
|  | All Senses | 69 (16.79) | 72 (17.52) | 211 (51.34) |
|  | Paraphrase Lexicon | 66 (16.06) | 55 (13.38) | 231 (56.20) |
| Short Answer | Pseudo Relevance | 6 (10.53) | 22 (38.59) | 29 (50.87) |
|  | First Sense | 9 (15.79) | 15 (26.32) | 33 (57.89) |
|  | All Senses | 7 (12.28) | 20 (35.08) | 30 (52.63) |
|  | Paraphrase Lexicon | 10 (17.54) | 15 (26.32) | 32 (56.14) |
| MEDLINE | WSD | 14 (05.38) | 2 (00.77) | 234 (90.00) |
|  | Without-WSD | 17 (06.54) | 5 (01.92) | 230 (88.46) |
|  | WSD Phrase | 13 (05.00) | 4 (01.54) | 234 (90.00) |
|  | Without-WSD Phrase | 15 (05.77) | 4 (01.54) | 233 (89.62) |

Table 4.9: Query by query performance. Number of queries for which the ranking is higher, lower or remained same using a query expansion approach

Corpus (57 queries), the percentage of queries with "lower" rank is higher (26.32%) for Paraphrase Lexicon and first sense, however overall, the Paraphrase Lexicon performs better than other query expansion approaches. A possible reason for the overall low performance is vocabulary mismatch: the plagiarised documents in the Extended Short Answer Corpus are from the Computer Science domain, whereas the Paraphrase Lexicon is generated using parallel documents from the European Parliament (see Section 4.3.2).

For the MEDLINE Corpus, the rank of most of the queries are the same since there is not much difference in performance between the various query expansion methods (see Table 4.8).

### 4.6.5 Exploring Thresholds

Sections 4.6.1, 4.6.2 and 4.6.3 presented the results for three datasets (note that for these results parameters were set automatically using 3-fold cross validation (see Section 4.5.3)). This section aims to explore the most appropriate features for query expansion. Experiments presented in this section are carried out using 10,479 source documents and 411 suspicious documents in the PAN-PC-10 Corpus which only contain simulated obfuscation (see Section 3.4.1.1).

| Parameters | | Avg. Recall for top K documents | | | |
|---|---|---|---|---|---|
| D | T | 5 | 10 | 15 | 20 |
| 5 | 1 | 0.4497 | 0.5182 | 0.5673 | 0.5969 |
| 10 | 1 | 0.4497 | 0.5182 | 0.5673 | 0.5969 |
| 15 | 1 | 0.4497 | 0.5182 | 0.5673 | 0.5969 |
| 20 | 1 | 0.4497 | 0.5182 | 0.5673 | 0.5969 |
| 25 | 1 | 0.4497 | 0.5182 | 0.5673 | 0.5969 |
| 30 | 1 | 0.4497 | 0.5182 | 0.5673 | 0.5969 |
| 5 | 2 | 0.3155 | 0.4112 | 0.4594 | 0.5008 |
| 10 | 2 | 0.2753 | 0.3540 | 0.4335 | 0.4659 |
| 15 | 2 | 0.2753 | 0.3540 | 0.4335 | 0.4659 |
| 20 | 2 | 0.2753 | 0.3540 | 0.4335 | 0.4659 |
| 25 | 2 | 0.2753 | 0.3540 | 0.4335 | 0.4659 |
| 30 | 2 | 0.2753 | 0.3540 | 0.4335 | 0.4659 |
| 5 | 3 | 0.2603 | 0.3086 | 0.3532 | 0.3942 |
| 10 | 3 | 0.2214 | 0.2798 | 0.3313 | 0.3589 |
| 15 | 3 | 0.2214 | 0.2798 | 0.3313 | 0.3589 |
| 20 | 3 | 0.2214 | 0.2798 | 0.3313 | 0.3589 |
| 25 | 3 | 0.2214 | 0.2798 | 0.3313 | 0.3589 |
| 30 | 3 | 0.2214 | 0.2798 | 0.3313 | 0.3589 |

Table 4.10: Query expansion with pseudo relevance feedback method using 411 suspicious plagiarised documents containing only simulated obfuscation from the PAN-PC-10 Corpus

Table 4.10 shows the results for pseudo relevance feedback method. Results are reported for top 5, 10, 15 and 20 candidate documents. In this table, $D$ is the number of top-returned documents to be used for extracting expansion terms and $T$ is the number of expansion terms to be extracted from top ranked documents. Overall, the best result is obtained with $T = 1$ for various values of $D$. Increasing the number of top-returned documents ($D$) does not help to improve performance. As the number of additional search terms ($T$) increases performance decreases. This demonstrates that increasing the value of $D$ and $T$ is likely to add noisy expansion terms which decreases the retrieval performance.

| Parameters | | Avg. Recall for top K documents | | | |
|---|---|---|---|---|---|
| W | S | 5 | 10 | 15 | 20 |
| 1 | 1 | 0.4562 | 0.5272 | 0.5681 | 0.5896 |
| 0.5 | 1 | 0.4980 | 0.5706 | 0.5981 | 0.6184 |
| 0.1 | 1 | **0.5215** | **0.5961** | **0.6294** | **0.6476** |
| 0.05 | 1 | 0.5158 | 0.5852 | 0.6277 | 0.6448 |
| 0.01 | 1 | 0.5158 | 0.5852 | 0.6277 | 0.6472 |
| 1 | 2 | 0.3990 | 0.4672 | 0.4968 | 0.5276 |
| 0.5 | 2 | 0.4826 | 0.5556 | 0.5888 | 0.6071 |
| 0.1 | 2 | 0.5211 | 0.5925 | 0.6204 | 0.6387 |
| 0.05 | 2 | 0.5174 | 0.5937 | 0.6253 | 0.6448 |
| 0.01 | 2 | 0.5199 | 0.5872 | 0.6277 | 0.6472 |
| 1 | 3 | 0.3670 | 0.4290 | 0.4797 | 0.5077 |
| 0.5 | 3 | 0.4712 | 0.5446 | 0.5896 | 0.6119 |
| 0.1 | 3 | 0.5174 | 0.5921 | 0.6253 | 0.6436 |
| 0.05 | 3 | 0.5138 | 0.5925 | 0.6269 | 0.6452 |
| 0.01 | 3 | 0.5195 | 0.5900 | 0.6241 | 0.6448 |

Table 4.11: Query expansion with *first sense* using 411 suspicious plagiarised documents containing only simulated obfuscation from the PAN-PC-10 Corpus

Tables 4.11, 4.12 and 4.13 show the results for first sense, all senses and sense selection after word sense disambiguation approaches based on WordNet and Table 4.14 shows the results for query expansion with the Paraphrase Lexicon. Results are reported for the top 5, 10, 15 and 20 candidate documents. In these tables, $W$ is the weight assigned to an additional search term and $S$ is the number of additional search terms used in the query expansion process.

As can be seen from these tables, in the majority of cases, the best results are obtained with $W = 0.1$ and $S = 1$ indicating that these are the most suitable features for query expansion on this corpus. Overall it can be observed that the number of expansion terms ($S$) and weight assigned to an expansion term ($W$) have a vital effect on performance. When the same weight ($W = 1$) is assigned to an original and expansion word, low results are obtained indicating that the selection of appropriate weight is important to get good performance. In the majority of cases, the most promising

| Parameters | | Avg. Recall for top K documents | | | |
|---|---|---|---|---|---|
| W | S | 5 | 10 | 15 | 20 |
| 1 | 1 | 0.4380 | 0.5146 | 0.5450 | 0.5629 |
| 0.5 | 1 | 0.4903 | 0.5710 | 0.5989 | 0.6196 |
| 0.1 | 1 | **0.5215** | **0.5965** | **0.6249** | 0.6444 |
| 0.05 | 1 | 0.5203 | 0.5904 | 0.6212 | 0.6407 |
| 0.01 | 1 | 0.5191 | 0.5852 | **0.6249** | **0.6456** |
| 1 | 2 | 0.3646 | 0.4290 | 0.4643 | 0.4878 |
| 0.5 | 2 | 0.4643 | 0.5357 | 0.5689 | 0.5904 |
| 0.1 | 2 | 0.5045 | 0.5860 | 0.6156 | 0.6350 |
| 0.05 | 2 | 0.5118 | 0.5807 | 0.6236 | 0.6407 |
| 0.01 | 2 | 0.5154 | 0.5827 | 0.6236 | 0.6431 |
| 1 | 3 | 0.3208 | 0.3706 | 0.4148 | 0.4351 |
| 0.5 | 3 | 0.4363 | 0.5126 | 0.5491 | 0.5661 |
| 0.1 | 3 | 0.5041 | 0.5852 | 0.6131 | 0.6363 |
| 0.05 | 3 | 0.5126 | 0.5815 | 0.6131 | 0.6363 |
| 0.01 | 3 | 0.5118 | 0.5852 | 0.6225 | 0.6431 |

Table 4.12: Query expansion with *all senses* using 411 suspicious plagiarised documents containing only simulated obfuscation from the PAN-PC-10 Corpus

| Parameters | | Avg. Recall for top K documents | | | |
|---|---|---|---|---|---|
| W | S | 5 | 10 | 15 | 20 |
| 1 | 1 | 0.4586 | 0.5426 | 0.5649 | 0.5843 |
| 0.5 | 1 | 0.5020 | 0.5608 | 0.5880 | 0.6071 |
| 0.1 | 1 | **0.5162** | **0.5843** | 0.5985 | **0.6285** |
| 0.05 | 1 | 0.5150 | 0.5746 | 0.5998 | 0.6249 |
| 0.01 | 1 | 0.5126 | 0.5742 | 0.**6058** | 0.6237 |
| 1 | 2 | 0.4238 | 0.4862 | 0.5264 | 0.5446 |
| 0.5 | 2 | 0.4899 | 0.5665 | 0.5843 | 0.5989 |
| 0.1 | 2 | 0.5101 | 0.5779 | 0.5957 | 0.6188 |
| 0.05 | 2 | 0.5138 | 0.5791 | 0.5973 | 0.6212 |
| 0.01 | 2 | 0.5126 | 0.5758 | 0.6010 | 0.6237 |
| 1 | 3 | 0.3986 | 0.4659 | 0.4899 | 0.5199 |
| 0.5 | 3 | 0.4854 | 0.5624 | 0.5831 | 0.5941 |
| 0.1 | 3 | 0.5154 | 0.5722 | 0.6030 | 0.6261 |
| 0.05 | 3 | 0.5154 | 0.5807 | 0.6022 | 0.6249 |
| 0.01 | 3 | 0.5142 | 0.5750 | 0.6022 | 0.6249 |

Table 4.13: Query expansion with *sense selection after word sense disambiguation* method using 411 suspicious plagiarised documents containing only simulated obfuscation from the PAN-PC-10 Corpus

| Parameters | | Avg. Recall for top K documents | | | |
|---|---|---|---|---|---|
| W | S | 5 | 10 | 15 | 20 |
| 1 | 1 | 0.4380 | 0.5057 | 0.5389 | 0.5616 |
| 0.5 | 1 | 0.4984 | 0.5657 | 0.6062 | 0.6221 |
| 0.1 | 1 | **0.5223** | **0.6075** | **0.6358** | **0.6577** |
| 0.05 | 1 | 0.5191 | 0.6034 | 0.6298 | 0.6553 |
| 0.01 | 1 | 0.5191 | 0.5937 | 0.6310 | 0.6517 |
| 1 | 2 | 0.3167 | 0.3958 | 0.4331 | 0.4672 |
| 0.5 | 2 | 0.4627 | 0.5284 | 0.5689 | 0.5921 |
| 0.1 | 2 | 0.5154 | 0.5945 | 0.6208 | 0.6488 |
| 0.05 | 2 | 0.5191 | 0.6014 | 0.6273 | 0.6504 |
| 0.01 | 2 | 0.5215 | 0.5949 | 0.6310 | 0.6529 |
| 1 | 3 | 0.2551 | 0.3074 | 0.3540 | 0.3800 |
| 0.5 | 3 | 0.4323 | 0.5045 | 0.5385 | 0.5543 |
| 0.1 | 3 | 0.5036 | 0.5933 | 0.6172 | 0.6379 |
| 0.05 | 3 | 0.5215 | 0.6038 | 0.6277 | 0.6521 |
| 0.01 | 3 | 0.5215 | 0.5961 | 0.6310 | 0.6529 |

Table 4.14: Query expansion with Paraphrase Lexicon using 411 suspicious plagiarised documents containing only simulated obfuscation from the PAN-PC-10 Corpus

results with 2 and 3 expansion terms are obtained with lowest weight $W = 0.01$. This indicates that expanding an original word with too many expansion words is likely to add noise. To normalize the effect of noise a possible strategy is to assign very small weight to expansion words.

Note that a similar set of experiments was carried out for the MEDLINE Corpus and the Extended Short Answer Corpus.

Regarding optimal parameter values, on all three corpora, the best results are obtained with $S = 1$ and $W = 0.1$ (see Section 4.5.3).

### 4.6.6 Summary of Results

Query expansion is incorporated into the proposed IR-based framework to improve candidate document retrieval performance. Overall, knowledge-based query expansion is helpful in improving the retrieval performance, particularly when the original text has been paraphrased. Query expansion using pseudo relevance feedback does not improve results. This shows that query expansion based on knowledge bases is more accurate than pseudo relevance feedback for text reuse detection (on the three corpora used in this study). Moreover, performance is not harmed when query expansion is applied to verbatim and slightly modified copies of documents. Results using the Paraphrase Lexicon are better than WordNet.

The choice of the number of expansion terms ($S$) and weights ($W$) assigned to them is important to get good results. A large number of expansion terms are likely to add noise.

## 4.7 Chapter Summary

This chapter described various query expansion approaches that were investigated to improve the candidate document retrieval performance, particularly when the source text has been paraphrased. Two widely used approaches for query expansion were explored: (1) pseudo relevance feedback and (2) query expansion using knowledge bases including WordNet, Paraphrase Lexicon and UMLS Metathesaurus. Evaluation was carried out using three benchmark corpora: PAN-PC-10 Corpus, Extended Short Answer Corpus and MEDLINE Corpus.

Results showed that query expansion based on WordNet, Paraphrase Lexicon and UMLS Metathesaurus improves retrieval performance (see Tables 4.6, 4.7 and 4.8 respectively). The selection of suitable expansion terms and assigning appropriate weights to them was found to be important for these methods.

# Chapter 5

# Pairwise Document Comparison using Modified and Weighted N-grams

## 5.1 Introduction

Chapter 3 presented an IR-based approach for the problem of candidate document retrieval. To further improve the performance query expansion was incorporated into the IR-based approach (Chapter 4). The aim was to retrieve a small set of "candidate documents" from large document collections, which includes the source(s) of text reuse. However, the actual source(s) of the reused text in the candidate documents is not known. This chapter describes a system which makes a pairwise document comparison for text reuse detection. The aim of pairwise comparison of documents is to determine whether one document has reused the other. Consequently, the source(s) of text reuse can be identified. The proposed system extends the widely used n-gram overlap approach with modified n-grams. N-grams are also weighted using the language model probability scores obtained by training a language model to assign more weight to rare n-grams and less weight to frequent ones.

The rest of this chapter is organised as follows: Section 5.2 describes the n-gram overlap approach. Section 5.3 describes the proposed approach based on modified n-grams. Section 5.4 presents the language modelling approach for weighting n-grams. Datasets and the evaluation methodology are described in Section 5.5. Finally, results and analysis are presented in Section 5.6.

## 5.2 Overlap of N-grams

Comparison of word and/or character n-grams has proven to be an effective method for detecting text reuse [Chiu et al., 2010; Clough et al., 2002] and plagiarism [Lyon et al., 2001; Potthast et al., 2010b, 2011; Stein et al., 2009] (see Section 2.4). However, a limitation of this approach is that it breaks down when the reused text has been heavily paraphrased. Ceska [2009] mentioned that insertion, deletion or substitution of even a single token in a text results in mismatch of at least one n-gram. Based on this assumption, if every n$^{\text{th}}$ token in a text is altered by employing any of these edit operations then text reuse will not be detected (assuming n-grams of length $n$ are used for comparison). Previously, Chen et al. [2010] used the n-gram overlap approach provided by ROUGE [Lin, 2004] for plagiarism detection. Synonym-based and relationship-based measures (using WordNet) were used to identify semantic similarity between a pair of words (see Section 2.8.2).

For these experiments, the degree of overlap between a document pair is computed using the *containment* similarity measure [Broder, 1997], which is computed as:

$$score_n(A,B) = \frac{|S(A,n) \bigcap S(B,n)|}{|S(B,n)|} \tag{5.1}$$

where $S\left(A,n\right)$ and $S\left(B,n\right)$ are the sets of word n-grams of length $n$ in source and suspicious documents respectively.

This measure has been previously used for measuring text reuse in journalism [Clough et al., 2002] and plagiarism detection [Chong et al., 2010] with promising results.

## 5.3 Modified N-grams

It is difficult to detect text reuse using the standard n-gram overlap approach when the original text has been modified. Consider the following example:

| | |
|---|---|
| Source: | *i ride in a car* |
| Rewrite: | *i* ***drive*** *in a* ***new motorcar*** |

The set of standard bigrams generated for the *source* and *rewrite* texts are {i ride, ride in, in a, a car} and {i drive, drive in, in a, a new, new motorcar} respectively. In this situation, the *source* bigrams “i ride”, “ride in” and “a car” do not match any of the *rewrite* bigrams. The only bigram in common between *source* and *rewrite* is “in a”. The editing of the source text has reduced the similarity between the sets of *source* and *rewrite* n-grams. Consequently, the overall similarity score will be low and text reuse is unlikely to be detected.

To detect text reuse created with paraphrasing, a modified n-gram approach is proposed. Using this approach, new n-grams are created in two ways: (1) Deletions (see Section 5.3.1) and (2) Substitutions (see Section 5.3.2). In the former approach, words in an n-gram are deleted, whereas in the latter approach, words in an n-gram are substituted with synonymous words from a lexical resource. Modified n-grams account for synonym replacement (Substitutions) and word deletion (Deletions), two common text editing operations [Bell, 1991]. N-grams generated using the modified n-gram approach are intended to improve matching with the original set of n-grams even when the original text has been paraphrased. The following subsections give more detail on the two methods used to generate modified n-grams.

### 5.3.1 Deletions (Del)

Modified n-grams are generated by deleting words. Assume that $w_1, w_2, ..., w_n$ is a word n-gram. Then a set of modified n-grams can be created by removing one of $w_2$ ... $w_{n-1}$. The first and last words in the n-gram are not removed. An n-gram of length $n$ will generate $n-2$ deleted n-grams and the length of each deleted n-gram will be $n-1$. No deleted n-grams are generated for unigrams and bigrams.

| Original | `he rides a new car` |
|---|---|
| Deletions | `he rides a car`<br>`he rides new car`<br>`he a new car` |

Table 5.1: Example modified n-grams generated using the Deletions approach

Table 5.1 shows examples of modified n-grams generated using the Deletions approach. Some of the modified n-grams are ungrammatical sequences (e.g. "`he a new car`"), however, this does not cause a problem for the proposed approach since it is very unlikely that these ungrammatical n-grams will occur in documents and therefore do not contribute to the text reuse detection score (see Section 5.3.3).

Each modified n-gram is associated with the n-gram from which it is derived i.e. "original n-gram → `associated modified n-grams`". The association is complex for this approach since the modified n-grams are shorter than the original and the modified n-gram approach compares n-grams of the same length (Section 5.3.3 describes how modified n-grams are compared). Each deleted n-gram is associated with the standard n-grams that can be derived from the original n-gram. For example (see Table 5.1), the original 5-gram "`he rides a new car`" will generate two standard 4-grams; "`he rides a new`" and "`rides a new car`".[1] The three deleted 4-grams will be associated to both standard 4-grams as:

[1] In the Deletions approach, the first and last words in the n-gram are not removed because they will generate standard (or original) n-grams instead of new modified n-grams.

he rides a new → `he rides a car, he rides new car, he a new car`

rides a new car → `he rides a car, he rides new car, he a new car`

### 5.3.2 Substitutions

Further n-grams are created by substituting one of the words in an n-gram with one of its synonyms from: (1) WordNet (WN), (2) Paraphrase Lexicon (Para) and (3) UMLS Metathesaurus (UMLS). The modified n-grams created by substitutions are likely to identify semantic similarity between suspicious-source sets of n-grams. Consequently, the overall similarity score will increase and help in detecting text reuse particularly when the original text has been paraphrased. The three methods used for creating modified n-grams with the substitutions approach are described in the following subsections.

#### 5.3.2.1 WordNet (WN)

Modified n-grams are created by substituting the word in the n-gram with one of its synonyms from WordNet (see Section 4.3.1). If the word belongs to multiple synsets then all are used to generate modified n-grams. Synonymous words are selected from *all senses* because it will generate more modified n-grams as compared to choosing the *first sense* or sense selection after word sense disambiguation. Each word in an n-gram is checked in WordNet. If found, all the synonyms from all senses are extracted (note that for simplicity, words in n-grams are not substituted with multi-word alternatives since these generate n-grams of different length to the original one).

| Original | `he rides a new car` |
|---|---|
| WordNet | `he rides a new motorcar`<br>`he rides a new automobile`<br>`he rides a fresh motorcar`<br>`he rides a fresh automobile` |

Table 5.2: Example modified n-grams generated using WordNet

Table 5.2 shows some of the modified n-grams generated using WordNet. The association of the original and modified n-grams is straightforward since both have the same length. The association for the example shown in Table 5.2 will be: "he rides a new car → `he rides a new motorcar, he rides a new automobile, he rides a fresh motorcar, he rides a fresh automobile`".

#### 5.3.2.2 Paraphrase Lexicon (Para)

Similar to the WordNet approach, n-grams can be created by substituting one of the words with an equivalent term from a Paraphrase Lexicon (which is referred as Para). A Paraphrase Lexicon is generated using an automatic paraphrase generation system [Callison-Burch, 2008] (see Section 4.3.2). Ten lexical equivalents are generated for each word[1] (see Table 5.3 for an example output). Modified n-grams are then created by substituting one of the words in the n-gram with one of the lexical equivalents. Multi-word lexical equivalents are not used for generating modified n-grams, similar to the WordNet approach presented in the previous section,

| Word | Lexical Equivalent |
|---|---|
| accurate | correct |
| accurate | precise |
| accurate | valid |
| accurate | exact |
| accurate | right |

Table 5.3: Example output of the automatic paraphrase generation system [Callison-Burch, 2008] for word "accurate"

Example modified n-grams generated using paraphrase lexicon are shown in Table 5.4. The association of the original and modified n-grams will be: "he rides a new car → `he rides a new vehicle, he drives a new car, he drives a new vehicle, he rides a new cars`".

[1] The default setting of the automatic paraphrase generation system [Callison-Burch, 2008] generates ten lexical equivalents for each input word.

| Original | `he rides a new car` |
|---|---|
| Paraphrase Lexicon | `he rides a new vehicle`<br>`he drives a new car`<br>`he drives a new vehicle`<br>`he drives a new cars` |

Table 5.4: Example modified n-grams generated using Paraphrase Lexicon

#### 5.3.2.3 UMLS Metathesaurus (UMLS)

Using the UMLS Metathesaurus, modified n-grams are generated by first mapping the input terms to UMLS CUIs using MetaMap (see Section 4.3.3.2). Then synonymous terms for all the CUIs mapped to an input term (similar to *all senses* in WordNet) are extracted from the *MRCONSO* table in the UMLS Metathesaurus (see Section 4.3.3.1). Synonymous terms obtained from the *MRCONSO* table are used to substitute words in the n-gram to generate new modified n-grams.

Note that citations (or documents) in the source and suspicious collections of the MEDLINE Corpus are pre-processed by parsing them with the MetaMap (see Section 4.3.3.2). During parsing, MetaMap treats many multi-word phrases as single terms. For simplicity each multi-word phrase is treated as a single term.

| Original | `a renal injury was reported` |
|---|---|
| UMLS Metathesaurus | `a renal injuries was reported`<br>`a kidneys injury was reported`<br>`a kidney injury was reported`<br>`a kidneys injuries was reported` |

Table 5.5: Example modified n-grams generated using UMLS Metathesaurus

Table 5.5 shows example modified n-grams generated using UMLS Metathesaurus. The association of the original and modified n-grams will be: “a renal injury was reported → a renal injuries was reported, a kidneys injury was reported, a kidney injury was reported, a kidneys injuries was reported”.

### 5.3.3 Comparing Modified N-grams

The modified n-grams are applied in the text reuse detection score by generating modified n-grams for the document that is suspected to contain reused text. These n-grams are then compared with the original document to determine the overlap. However, the techniques in Section 5.3 generate a large number of modified n-grams which means that the number of n-grams that overlap with document $A$ (source document) can be greater than the total number of n-grams in $B$ (suspicious document), leading to similarity scores greater than 1. To avoid this the n-gram overlap counts are constrained in a similar way that they are clipped in BLEU [Papineni et al., 2002] and ROUGE [Lin, 2004] (see Section 2.4.3).

For each n-gram in $B$, a set of modified n-grams, $mod(ngram)$, is created.[1] The count for an individual n-gram in $B$, $count(ngram, B)$, can be computed as the number of times any n-gram in $mod(ngram)$ occurs in $A$ (see Equation 5.2).

$$mod\_count(ngram, A) = \begin{cases} count(ngram, A) \text{ if } count(ngram, A) > 0, \\ ARGMAX_{ngram' \in mod(ngram)}\ count(ngram', A) \text{ otherwise.} \end{cases} \tag{5.2}$$

However, the contribution of this count to the text reuse detection score has to be bounded to ensure that the combined count of the modified n-grams appearing in $A$ does not exceed the number of times the original n-gram occurs in $B$. Consequently the text reuse detection score, $score_n(A, B)$, is computed as:

$$score_n(A, B) = \frac{\sum\limits_{\substack{ngram \\ \in B}} min(mod\_count(ngram, A), count(ngram, B))}{\sum\limits_{ngram \in B} count(ngram, B)} \tag{5.3}$$

[1] This is the set of n-grams that could have been created by modifying an n-gram in $B$ and includes the original n-gram itself.

where $mod\_count(ngram, A)$ is the number of times an n-gram $(ngram)$ in the set of modified n-grams $mod(ngram)$ occurs in $A$ and $count(ngram, B)$ is the number of times $ngram$ occurs in B.

The example shown in Table 5.6 explains the clipping of n-gram counts and restriction of the containment similarity score between 0 and 1.

| Document | Set of Unigrams |
|---|---|
| Suspicious | the, the, boy, in, in, the, park |
| Modified Suspicious | the, the, boy→`child, teenager`, in, in, the, park→`playground, ground` |
| Source | the, the, the, the, the, boy, child, ground, in, in, in, playground |

Table 5.6: Example showing comparison of modified n-grams

*Suspicious* and *Source* represent the set of unigrams in the suspicious and source documents respectively. *Modified Suspicious* represents the set of unigrams created by associating modified unigrams to the original unigrams in the suspicious document. For example, in "boy→`child, teenager`", "boy" is the original unigram and {`child, teenager`} are the modified unigrams.

Consider the first scenario in which *Suspicious* is compared with *Source* (this comparison can be made using the standard n-gram overlap approach described in Section 5.2). Since the containment similarity score is normalised by the number of n-grams in the *Suspicious* document (see Equation 5.1), the count of an n-gram $count(ngram, Suspicious)$ is limited to its count in the *Suspicious* document. In the above example, $count(ngram, Source)$ for the unigram "the" is 5 in the *Source* document and 3 in the *Suspicious* document. So, $count(ngram, Suspicious)$ for this unigram will be 3. The containment similarity score between *Suspicious* and *Source* document pair will be: $score_n(\text{Source, Suspicious}) = 6/7 = 0.857$.

The situation is more complex when comparing *Modified Suspicious* with *Source* (Section 5.3). To restrict the similarity score between 0 and 1, first Equation 5.2 is used to compute $mod\_count(ngram, A)$ and then Equation 5.3 is used to compute the text reuse detection score.

If an original n-gram in the *Modified Suspicious* matches a *Source* n-gram then $mod\_count(ngram, A) = count(ngram, A)$, and the modified n-grams associated with it are not checked for matching (see Equation 5.2). Otherwise, associated modified n-grams are checked for matching and the maximum 'count' value for a modified n-gram is returned as the $mod\_count(ngram, A)$ value. After computing the $mod\_count(ngram, A)$, Equation 5.3 is used to compute the text reuse detection score.

In the above example, the original unigram "boy" matches the *Source* unigram, so its associated modified unigrams (`child` and `teenager`) will not be checked (although the associated modified unigram "child" matches the *Source* unigram). The 'count' of "boy" is 1 so $mod\_count(ngram, A) = 1$. The original unigram "park" does not match any of the *Source* unigrams, therefore, its associated modified unigrams are checked to determine whether they match. Both the modified unigrams ("playground" and "ground") match the *Source* unigrams and both of them have a 'count' of 1, so $mod\_count(ngram, A) = 1$. The containment similarity score between *Modified Suspicious* and *Source* will be: $score_n(\text{Source, Modified Suspicious}) = 7/7 = 1$.

## 5.4 Weighting N-grams

In a document (or document collection), some words (or phrases) are likely to be more important than others and will be more useful in determining whether texts have been reused. For example, rare words (or phrases) are likely to be more important than frequent ones. The standard n-gram overlap approach assigns equal weight to all the n-grams and relative importance of each n-gram is not taken into account. To give more importance to rare n-grams and decrease the contribution of frequent ones, n-grams are weighted with probability scores obtained by training a statistical n-gram language model. Previously Errami et al. [2010] weighted word 6-grams using language model probability scores (see Section 2.4.1).

The main task of a language model is to assign a probability to a sequence of words: $w_1, w_2, ..., w_{n-1}, w_n$. N-gram language models have been used in a variety of applications including speech recognition, handwriting recognition, statistical machine translation and spelling correction [Jurafsky and Martin, 2008].

N-gram language models are a widely used type of language model that estimate the probability of a sequence of words based on the probabilities of the n-grams they contain. N-gram probabilities can be estimated by counting their frequencies in a corpus. However, as the length of $n$ increases, the problem of data sparseness increases. To avoid this problem, the probabilities of lower order n-grams ($n \leq 2$) can be used to estimate the probabilities of higher order n-grams ($n \geq 3$). For example, probability estimates of higher order n-grams ($n \geq 3$) can be calculated using a bigram language model. An n-gram is broken into successive bigrams and the probability of the n-gram will be the product of the probabilities of the bigrams in the n-gram, which is computed as:

$$P = (w_n | w_{n-1}, w_{n-2}, ..., w_1) = \prod_{i=2}^{n} P\left(w_i | w_{i-1}\right) \tag{5.4}$$

where $w_1, w_2, ..., w_n$ is an n-gram ($n \geq 3$).

Due to data sparseness, n-grams not appearing in the training data are assigned a 0 probability score. In addition, the probability estimates of the bi- or tri-grams occurring only a very few times are not reliable. Approaches have been proposed to assign some probability to the unseen bi- or tri-grams and also to those which have very few occurrences (called *smoothing*) [Jurafsky and Martin, 2008].

For these experiments, probabilities for each n-gram are obtained using a language model and used to increase the contribution of rare n-grams and decrease the importance of common ones. Word uni-gram and bi-gram probabilities are computed using the SRILM language modeling toolkit [Stolcke, 2002]. The probabilities of higher order n-grams ($n > 2$) are computed using the bigram probabilities (see Equation 5.4). Good-Turing smoothing [Wang et al., 2007] is used to smooth word n-grams.

The score for each n-gram is computed as its Information Content (IC) [Cover and Thomas, 1991], i.e. $-log(P)$ (referred as '$LM(ngram)$'). The $-log$ is taken instead of $log$ because it will assign higher weights to more important (or rare) n-grams and lower weights to less important (or frequent) n-grams.

Note that when the Language Model (LM) approach is applied the $mod\_count(ngram, A)$ is computed using Equation 5.2. The overall text reuse detection score, $score_{LM}(A, B)$, is computed by combining the language model probability score for an n-gram ($LM(ngram)$) with an associated 'count' value as:

$$score_{LM}(A,B) = \frac{\sum\limits_{\substack{ngram \\ \in B}} min(mod\_count(ngram, A), count(ngram, B)) \cdot LM(ngram)}{\sum\limits_{ngram \in B} count(ngram, B)} \tag{5.5}$$

where $mod\_count(ngram, A)$ is the number of times an n-gram ($ngram$) in the set of modified n-grams $mod(ngram)$ occurs in $A$ and $count(ngram, B)$ is the number of times $ngram$ occurs in B.

Table 5.7 shows an example of bigrams, their probabilities and scores assigned to them when computing the containment similarity score using Equations 5.1, 5.2 and 5.3. The bigrams which frequently occur in the training dataset have high probability but low score and situation is opposite for the rare bigrams (low probability but high score). Consequently, when bigrams are weighted with language model probability scores, it will decrease the contribution of frequent bigrams and increase the contribution of rare ones in the overall text reuse detection score.

| Bigram | Probability ($P$) | Score ($-log(P)$) |
|---|---|---|
| among nationals | 0.0120 | 4.4228 |
| among operating | 0.0139 | 4.2758 |
| among sectors | 0.0371 | 3.2941 |
| among special | 0.0515 | 2.9661 |
| among those | 0.2442 | 1.4098 |
| among other | 0.2827 | 1.2633 |

Table 5.7: Example of bigrams probabilities and scores

## 5.5 Experimental Setup

This section describes three benchmark corpora, data used to train language models and the evaluation methodology used to evaluate the proposed approach.

### 5.5.1 Datasets

The proposed approach is evaluated using three datasets: (1) METER Corpus, (2) Short Answer Corpus and (3) MEDLINE Corpus.

For these experiments, the entire METER Corpus (see Section 2.12.3) and the Short Answer Corpus (see Section 2.12.4) are used. However, in case of the MEDLINE Corpus (see Section 2.12.2), the subset of 260 manually examined and verified duplicate citation pairs with no shared author is used as the "Plagiarised" set of documents. The *Deja vu* database[1] does not contain independently written (or non-plagiarised) citation pairs, so a set of 260 citation pairs is randomly selected from MEDLINE to make a collection of "Non-Plagiarised" documents. In total, there are 520 citation pairs (half plagiarised and half non-plagiarised).

### 5.5.2 Training Language Models

To assign appropriate weights to n-grams, three different language models are trained for three evaluation corpora: Reuters Language Model (RLM) for the METER Corpus,

[1] http://dejavu.vbi.vt.edu/dejavu/duplicate/ Last visited: 03-07-2012

(2) Wikipedia Language Model (WLM) for the Short Answer Corpus and (3) MEDLINE Language Model (MLM) for the MEDLINE Corpus. Separate language models are build for each corpus since these corpora contain documents from different domains.

The choice of data collection for training a language model is based on the domain of the documents in the corpus. The METER corpus is a collection of news stories on two topics: (1) law and court stories and (2) showbusiness. The Reuters Corpus [Rose et al., 2002] is a large collection of news articles, which are labeled with topic codes. Five topic codes indicate documents similar to those in the METER Corpus: (1) legal/judicial, (2) current news - entertainment, (3) crime, law enforcement, (4) arts, culture, entertainment and (5) fashion. In total 806,791 news articles from the Reuters Corpus are used to train the Reuters Language Model.

The Short Answer corpus contains answers to five questions in the Computer Science domain (see Section 2.12.4). Wikipedia articles which contain any of the keywords of these five topics (including the keyword "computer science") are selected to train the Wikipedia Language Model. Total 500,000 articles are selected from the Wikipedia dump.[1]

For the Medline Language Model, 344,000 citations from the 2011 MEDLINE/PubMed Baseline Repository[2] are randomly selected. All citations are parsed using the MetaMap (see Section 4.3.3.2). MetaMap treats many multi-word phrases as single terms. For simplicity, each multi-word phrase is treated as a single term while training the language model.

To train a language model, text is pre-processed by removing all punctuation marks and converting to lower case.

[1] For these experiments, April 2011 Wikipedia dump was used.

[2] The 2011 MEDLINE/PubMed Baseline Repository was downloaded from http://mbr.nlm.nih.gov/ on 25-04-2011

### 5.5.3 Evaluation Methodology

The set of experiments presented in this chapter aims to distinguish between different levels of text reuse or plagiarism in a corpus. The category of a document corresponds to the level of text reuse or plagiarism contained in it. The METER corpus has three levels of text reuse: (1) Wholly Derived (WD), (2) Partially Derived (PD) and (3) Non Derived (ND); the Short Answer Corpus has four levels of plagiarism: (1) near copy, (2) light revision, (3) heavy revision and (4) non-plagiarised; and the MEDLINE Corpus has two levels of plagiarism: (1) plagiarised and (2) non-plagiarised.

The problem of distinguishing between different levels of text reuse or plagiarism is treated as a supervised document classification task. Two versions of the task are used: (1) binary classification and (2) multi-classification. In the former case, the aim is to distinguish between two categories. For the METER Corpus, WD and PD documents are combined to make the "Derived" class and ND documents make the "Non Derived" class. For the Short Answer Corpus, documents categorised as near copy, light revision and heavy revision are combined to make the "Plagiarised" class and the set of non-plagiarised documents make the "Non-Plagiarised" class. In the latter case, the goal is to distinguish between various levels of text reuse or plagiarism.

$N$-fold cross-validation is used to better estimate the performance of the proposed approach. 10-fold cross-validation is used for experiments with the METER and MEDLINE corpora. However, due to the small number of instances in the Short Answer Corpus, 3-fold cross-validation is applied.

For these experiments, the Naive Bayes classifier, a simple probabilistic classification algorithm based on Bayes' theorem, is used. This classifier uses the set of training examples to create a probabilistic model, which can be further used for the classification of unseen examples.

The WEKA[1] implementation of the Naive Bayes classifier is used. It is appropriate

[1] http://www.cs.waikato.ac.nz/ml/weka/ Last visited: 12-08-2012

for these experiments because it can operate on *numeric* features and the features generated by the n-gram overlap approach (see Section 5.2) and the modified n-gram approach (Section 5.3) are also *numeric*.

Containment similarity scores for each suspicious-source document pair are computed for word unigrams, bigrams, trigrams, fourgrams and fivegrams. These five similarity score are used as features for the Naive Bayes classifier. Note that for the METER Corpus, containment similarity scores between all PA source texts and news articles on the same story are calculated.

### Evaluation Measures

Precision, recall and $F_1$ (see Section 2.13) scores are computed for each class (or level of text reuse) using $n$-fold cross-validation. Macro-averaged precision, recall and $F_1$ scores computed across all classes are reported.

## 5.6 Results and Analysis

In classification tasks, the performance of a machine learning algorithm can be compared with a simple baseline approach called the Most Common Category (MCC). Using this approach, the accuracy of a machine learning algorithm can be computed by assuming that it will assign the most common category to all the examples in the dataset.

For the METER Corpus (see Section 2.12.3), the performance with the MCC approach is 0.78 and 0.46 for the binary and ternary classification tasks respectively. For the Short Answer Corpus (see Section 2.12.4), the performance with the MCC approach is 0.60 and 0.40 for the binary and multi classification tasks respectively. For the binary classification task in the MEDLINE Corpus (see Section 2.12.2) performance using the MCC approach is 0.50.

### 5.6.1 Results for METER Corpus

Tables 5.8 and 5.9 show the results for both binary- and multi-classification. "NG" refers to the comparison of n-grams in each document (Section 5.2), while "Del", "WN" and "Para" refer to the modified n-grams created using Deletions, WordNet and Paraphrase Lexicon respectively (Section 5.3). The prefix "LM" (e.g. "LM-NG") indicates that the n-grams are weighted using the language model probability scores (Section 5.4).

"Unigrams" means that containment similarity score obtained using word unigrams is used as a single feature for the classification task. Similarly "Bigrams", "Trigrams", "Fourgrams" and "Fivegrams" mean that the containment similarity score obtained using word bigrams, trigrams, fourgrams and fivegrams respectively is used as a single feature. "Combined" means that containment similarity scores obtained for word unigrams, bigrams, trigrams, fourgrams and fivegrams are used as a set of features ('five' features) for the classification task. Since no deleted n-grams are generated for unigrams (see Section 5.3.1), no results are reported (marked as "—").

For the binary classification task (Table 5.8) including modified n-grams improves performance. This improvement is observed when each of the three types of modified n-grams is applied individually, with a greater increase being observed for the n-grams created using the WordNet and Paraphrase Lexicon approaches. Further improvement is observed when different types of modified n-grams are combined with the best performance obtained when all three types are used. All improvements over the baseline approach (NG) are statistically significant (Wilcoxon signed-rank test, $p < 0.05$). These results demonstrate that the various types of modified n-grams all contribute to identifying when text is being reused.

Performance consistently improves when n-grams are weighted using language model scores. This demonstrates that the information provided by the language model is useful in determining the relative importance of n-grams. The improvement is statisti-

| | Feature | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Unigrams | | | Bigrams | | | Trigrams | | |
| Approach | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| NG | 0.758 | 0.786 | 0.764 | 0.791 | 0.797 | 0.794 | 0.794 | 0.767 | 0.777 |
| LM-NG | 0.808 | 0.822 | 0.811 | 0.831 | 0.823 | 0.826 | 0.811 | 0.767 | 0.781 |
| Del | — | — | — | 0.836 | 0.832 | 0.833 | 0.826 | 0.793 | 0.804 |
| LM-Del | — | — | — | 0.870 | 0.862 | 0.865 | 0.843 | 0.799 | 0.811 |
| WN | 0.904 | 0.906 | 0.905 | 0.841 | 0.837 | 0.839 | 0.825 | 0.778 | 0.792 |
| LM-WN | 0.908 | 0.908 | 0.908 | 0.865 | 0.857 | 0.860 | 0.815 | 0.789 | 0.799 |
| Para | 0.910 | 0.910 | 0.910 | 0.833 | 0.828 | 0.830 | 0.816 | 0.787 | 0.797 |
| LM-Para | 0.917 | 0.917 | 0.917 | 0.865 | 0.857 | 0.860 | 0.824 | 0.781 | 0.794 |
| Del+WN | — | — | — | 0.872 | 0.868 | 0.869 | 0.832 | 0.805 | 0.814 |
| LM-Del+WN | — | — | — | 0.890 | 0.885 | 0.887 | 0.848 | 0.810 | 0.821 |
| Del+Para | — | — | — | 0.871 | 0.868 | 0.869 | 0.838 | 0.811 | 0.820 |
| LM-Del+Para | — | — | — | 0.889 | 0.882 | 0.885 | 0.857 | 0.813 | 0.825 |
| WN+Para | 0.926 | 0.925 | 0.925 | 0.839 | 0.836 | 0.837 | 0.815 | 0.784 | 0.795 |
| LM-WN+Para | **0.938** | **0.938** | **0.938** | 0.868 | 0.859 | 0.863 | 0.827 | 0.781 | 0.795 |
| Del+WN+Para | — | — | — | 0.871 | 0.868 | 0.869 | 0.845 | 0.815 | 0.824 |
| LM-Del+WN+Para | — | — | — | **0.892** | **0.887** | **0.889** | **0.852** | **0.820** | **0.830** |

| | Feature | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Fourgrams | | | Fivegrams | | | Combined | | |
| Approach | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| NG | 0.812 | 0.720 | 0.743 | 0.791 | 0.704 | 0.728 | 0.830 | 0.681 | 0.709 |
| LM-NG | 0.795 | 0.732 | 0.751 | 0.818 | 0.710 | 0.735 | 0.840 | 0.703 | 0.729 |
| Del | 0.824 | 0.738 | 0.760 | 0.811 | 0.713 | 0.737 | 0.840 | 0.720 | 0.745 |
| LM-Del | 0.815 | 0.744 | 0.763 | 0.825 | 0.722 | 0.745 | 0.850 | 0.738 | 0.761 |
| WN | 0.816 | 0.727 | 0.749 | 0.823 | 0.730 | 0.752 | 0.864 | 0.779 | 0.797 |
| LM-WN | 0.806 | 0.730 | 0.751 | 0.827 | 0.729 | 0.762 | 0.869 | 0.791 | 0.808 |
| Para | 0.805 | 0.728 | 0.749 | 0.821 | 0.728 | 0.750 | 0.867 | 0.788 | 0.805 |
| LM-Para | 0.812 | 0.727 | 0.749 | 0.838 | 0.731 | 0.755 | 0.876 | 0.804 | 0.819 |
| Del+WN | 0.817 | 0.751 | 0.769 | 0.829 | 0.727 | 0.760 | 0.874 | 0.809 | 0.824 |
| LM-Del+WN | **0.825** | **0.754** | **0.773** | 0.814 | 0.738 | 0.759 | 0.882 | 0.822 | 0.835 |
| Del+Para | 0.821 | 0.749 | 0.768 | 0.821 | 0.734 | 0.756 | 0.879 | 0.816 | 0.829 |
| LM-Del+Para | 0.823 | 0.748 | 0.767 | 0.835 | 0.740 | 0.764 | 0.884 | 0.822 | 0.835 |
| WN+Para | 0.806 | 0.728 | 0.750 | 0.82 | 0.732 | 0.754 | 0.878 | 0.814 | 0.828 |
| LM-WN+Para | 0.812 | 0.730 | 0.752 | 0.826 | 0.747 | 0.767 | 0.884 | 0.826 | 0.839 |
| Del+WN+Para | 0.823 | 0.748 | 0.767 | 0.831 | 0.750 | 0.770 | 0.887 | 0.828 | 0.841 |
| LM-Del+WN+Para | 0.821 | 0.749 | 0.768 | **0.831** | **0.750** | **0.770** | **0.893** | **0.846** | **0.857** |

Table 5.8: Results for binary classification using METER Corpus

cally significant for the “LM-NG”, “LM-Del”, “LM-WN” and “LM-Para” approaches compared to the “NG”, “Del”, “WN” and “Para” approaches respectively (Wilcoxon signed-rank test, $p < 0.05$).

Performance decreases as the length of n-gram increases. This is likely to happen because the news stories created using the PA source text involve substantial rewriting which makes it difficult to find longer matching n-grams. The best results are obtained with word unigrams ($F_1 = 0.938$) and the second highest with bigrams ($F_1 = 0.889$). This indicates that smaller n-grams ($n \leq 2$) can capture the similarity when the original text has been rewritten (or paraphrased) better than longer n-grams ($n \geq 3$). This finding is also consistent with the previous studies [Barrón-Cedeño et al., 2009; Clough, 2003b] which showed that word unigrams and bigrams are the best features on this corpus. Combining all five features ($F_1 = 0.857$) does not improve results compared with word unigrams ($F_1 = 0.938$).

It can also be noted that results with “Bigrams” is higher than “Unigrams” for the “NG” (Unigrams: $F_1 = 0764$, Bigrams: $F_1 = 0.794$) and “LM-NG” (Unigrams: $F_1 = 0.811$, Bigrams: $F_1 = 0.826$) approaches. However, performance with “Unigrams” is higher than “Bigrams” when modified n-grams are added to the original set of n-grams.

For all lengths of n-grams ($n = 1 - 5$), the best results are obtained when n-grams are weighted with language model probability scores, highlighting the usefulness of the language model. For the majority of n-gram lengths, the best results are obtained when all types of modified n-grams are combined and weighted i.e. “LM-Del+WN+Para” approach (except for the “Fourgrams”) indicating that modified n-grams and weighting n-grams help to improve performance.

Results for the ternary classification are shown in Table 5.9. Overall, results show a similar pattern to those observed for the binary classification and the best result is also obtained when all three types of modified n-grams are included and n-grams

| | Feature | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Unigrams | | | Bigrams | | | Trigrams | | |
| Approach | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| NG | 0.552 | 0.560 | 0.546 | 0.607 | 0.600 | 0.601 | 0.607 | 0.582 | 0.584 |
| LM-NG | 0.614 | 0.613 | 0.607 | 0.634 | 0.626 | 0.627 | 0.613 | 0.583 | 0.585 |
| Del | — | — | — | 0.635 | 0.631 | 0.632 | 0.618 | 0.599 | 0.599 |
| LM-Del | — | — | — | 0.661 | 0.658 | 0.657 | 0.628 | 0.605 | 0.604 |
| WN | 0.644 | 0.623 | 0.614 | 0.635 | 0.631 | 0.631 | 0.619 | 0.593 | 0.595 |
| LM-WN | 0.662 | 0.645 | 0.639 | 0.663 | 0.660 | 0.658 | 0.628 | 0.602 | 0.603 |
| Para | 0.623 | 0.602 | 0.584 | 0.631 | 0.626 | 0.626 | 0.614 | 0.589 | 0.590 |
| LM-Para | 0.634 | 0.610 | 0.600 | 0.658 | 0.654 | 0.652 | 0.630 | 0.605 | 0.605 |
| Del+WN | — | — | — | 0.644 | 0.644 | 0.643 | 0.636 | 0.620 | 0.620 |
| LM-Del+WN | — | — | — | 0.663 | 0.660 | 0.658 | **0.642** | **0.623** | **0.621** |
| Del+Para | — | — | — | 0.650 | 0.649 | 0.648 | 0.634 | 0.618 | 0.616 |
| LM-Del+Para | — | — | — | 0.666 | 0.665 | 0.663 | 0.631 | 0.610 | 0.609 |
| WN+Para | 0.661 | 0.643 | 0.633 | 0.632 | 0.628 | 0.627 | 0.617 | 0.591 | 0.592 |
| LM-WN+Para | **0.679** | **0.661** | **0.658** | 0.658 | 0.655 | 0.652 | 0.630 | 0.605 | 0.605 |
| Del+WN+Para | — | — | — | 0.650 | 0.650 | 0.649 | 0.636 | 0.615 | 0.614 |
| LM-Del+WN+Para | — | — | — | **0.675** | **0.675** | **0.673** | 0.638 | 0.619 | 0.617 |

| | Feature | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Fourgrams | | | Fivegrams | | | Combined | | |
| Approach | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| NG | 0.598 | 0.549 | 0.551 | 0.586 | 0.523 | 0.525 | 0.585 | 0.540 | 0.531 |
| LM-NG | 0.602 | 0.555 | 0.554 | 0.604 | 0.544 | 0.544 | 0.598 | 0.558 | 0.551 |
| Del | 0.605 | 0.564 | 0.564 | 0.594 | 0.537 | 0.537 | 0.606 | 0.572 | 0.563 |
| LM-Del | 0.615 | 0.568 | 0.567 | 0.600 | 0.555 | 0.554 | 0.620 | 0.595 | 0.589 |
| WN | 0.594 | 0.549 | 0.550 | 0.610 | 0.542 | 0.546 | 0.624 | 0.613 | 0.609 |
| LM-WN | 0.599 | 0.555 | 0.556 | 0.605 | 0.550 | 0.551 | 0.637 | 0.625 | 0.620 |
| Para | 0.589 | 0.543 | 0.545 | 0.601 | 0.548 | 0.548 | 0.634 | 0.621 | 0.616 |
| LM-Para | 0.604 | 0.556 | 0.557 | 0.600 | 0.552 | 0.550 | 0.644 | 0.632 | 0.627 |
| Del+WN | 0.614 | 0.576 | 0.577 | 0.611 | 0.554 | 0.552 | 0.634 | 0.628 | 0.622 |
| LM-Del+WN | **0.623** | **0.579** | **0.579** | 0.619 | 0.552 | 0.556 | 0.632 | 0.622 | 0.617 |
| Del+Para | 0.611 | 0.570 | 0.570 | 0.604 | 0.552 | 0.551 | 0.642 | 0.633 | 0.628 |
| LM-Del+Para | 0.621 | 0.577 | 0.576 | 0.623 | 0.560 | 0.559 | 0.643 | 0.637 | 0.631 |
| WN+Para | 0.589 | 0.543 | 0.545 | 0.606 | 0.557 | 0.556 | 0.647 | 0.640 | 0.635 |
| LM-WN+Para | 0.604 | 0.556 | 0.557 | 0.614 | 0.569 | 0.567 | 0.654 | 0.650 | 0.644 |
| Del+WN+Para | 0.611 | 0.570 | 0.570 | 0.613 | 0.563 | 0.562 | 0.648 | 0.642 | 0.637 |
| LM-Del+WN+Para | 0.621 | 0.577 | 0.576 | **0.619** | **0.576** | **0.576** | **0.658** | **0.651** | **0.646** |

Table 5.9: Results for ternary classification using METER Corpus

are weighted with probability scores (except for 3-grams and 4-grams - the best result in these cases is when Para isn't used but other approaches are). The improvement in performance is statistically significant for all approaches compared to the baseline approach "NG" (Wilcoxon signed-rank test, $p < 0.05$).

Interestingly, the best results are obtained with word bigrams ($F_1 = 0.673$) and second highest with unigrams ($F_1 = 0.658$). This demonstrates that bigrams are more appropriate then unigrams in discriminating between three rewrite levels (WD, PD and ND) in the METER Corpus.

Once again weighting n-grams with language model scores improves results for all types of n-gram and this improvement is statistically significant for the "LM-NG", "LM-Del" and "LM-WN+Para" approaches compared to the "NG", "Del" and "WN+Para" approaches respectively (Wilcoxon signed-rank test, $p < 0.05$). Low performance is obtained for longer n-grams ($n \geq 3$), indicating that when original text is modified it becomes difficult to find long matches. Similar to the binary classification, combining features ("Combined") does not give a better result ($F_1 = 0.646$) than the best results using single n-grams ("Bigrams", $F_1 = 0.673$). For all lengths of n-grams, adding modified n-grams to the original n-grams and weighting them gives the best results.

Results for all approaches are lower for ternary classification than binary classification. This is because the binary classification task involves distinguishing between two classes of documents which are relatively distinct ("Derived" and "Non Derived") while the ternary task divides the "Derived" class into two (WD and PD) which are more difficult to separate [Clough et al., 2002]. Table 5.10 shows the confusion matrix for the "NG" approach with "Combined" features. It can be noted that it is difficult to distinguish PD class from other two classes (WD and ND), particularly it is more difficult to distinguish between WD and PD classes. Consequently, overall performance decreases for the ternary classification.

| Classified as | WD | PD | ND |
|---|---|---|---|
| WD | 182 | 72 | 47 |
| PD | 85 | 155 | 198 |
| ND | 2 | 30 | 173 |

Table 5.10: Confusion matrix for the ternary classification when “NG” approach is used with “Combined” features

### 5.6.2 Results for Short Answer Corpus

Table 5.11 shows the results for the binary classification. Overall, results show a similar pattern to that of the METER Corpus (see Table 5.8). As the length of n-gram increases performance decreases. The “Unigrams” feature gives the best results ($F_1 = 0.989$). The combination of different features (“Combined” approach) is not helpful in improving results. For different lengths of n-grams, the best results are obtained when all types of modified n-grams are combined and weighted with probability scores indicating that combination of different resources (WN and Para) for creating modified n-grams and weighting them with language model probability scores is effective in improving performance. Compared to the baseline approach (NG) improvement with all other approaches is statistically significant (Wilcoxon signed-rank test, $p < 0.05$).

Performance with the baseline approach (NG) is high compared to the METER Corpus for different lengths of n-grams. A possible reason is the clear distinction between plagiarised and non-plagiarised documents in the Short Answer Corpus compared to the METER Corpus (see confusion matrix in Table 5.13, which shows a clear distinction between plagiarised and non-plagiarised documents). This corpus contains documents which were intentionally created as plagiarised and non-plagiarised. The plagiarised documents were created using the source text (with three rewrite levels) and non-plagiarised documents were written without using the source text. On the other hand, in the METER Corpus, the news stories published by nine British newspapers were collected and manually annotated by journalists as WD, PD and ND, depending on the

|  | Feature |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | Unigrams |  |  | Bigrams |  |  | Trigrams |  |  |
| Approach | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| NG | 0.864 | 0.863 | 0.863 | 0.950 | 0.947 | 0.948 | 0.941 | 0.937 | 0.937 |
| LM-NG | 0.908 | 0.905 | 0.906 | 0.950 | 0.947 | 0.948 | 0.948 | 0.947 | 0.947 |
| Del | — | — | — | 0.950 | 0.947 | 0.948 | 0.948 | 0.947 | 0.947 |
| LM-Del | — | — | — | 0.950 | 0.947 | 0.948 | **0.959** | **0.958** | **0.958** |
| WN | 0.979 | 0.979 | 0.979 | 0.959 | 0.958 | 0.958 | **0.959** | **0.958** | **0.958** |
| LM-WN | 0.980 | 0.979 | 0.979 | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |
| Para | **0.990** | **0.989** | **0.989** | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |
| LM-Para | **0.990** | **0.989** | **0.989** | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |
| Del+WN | — | — | — | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |
| LM-Del+WN | — | — | — | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |
| Del+Para | — | — | — | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |
| LM-Del+Para | — | — | — | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |
| WN+Para | **0.990** | **0.989** | **0.989** | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |
| LM-WN+Para | **0.990** | **0.989** | **0.989** | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |
| Del+WN+Para | — | — | — | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |
| LM-Del+WN+Para | — | — | — | **0.971** | **0.968** | **0.969** | **0.959** | **0.958** | **0.958** |

|  | Feature |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | Fourgrams |  |  | Fivegrams |  |  | Combined |  |  |
| Approach | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| NG | 0.929 | 0.926 | 0.927 | 0.904 | 0.895 | 0.896 | 0.950 | 0.947 | 0.948 |
| LM-NG | **0.941** | **0.937** | **0.937** | 0.904 | 0.895 | 0.896 | 0.950 | 0.947 | 0.948 |
| Del | **0.941** | **0.937** | **0.937** | 0.904 | 0.895 | 0.896 | 0.950 | 0.947 | 0.948 |
| LM-Del | **0.941** | **0.937** | **0.937** | 0.904 | 0.895 | 0.896 | 0.950 | 0.947 | 0.948 |
| WN | **0.941** | **0.937** | **0.937** | 0.920 | 0.916 | 0.916 | 0.959 | 0.958 | 0.958 |
| LM-WN | **0.941** | **0.937** | **0.937** | 0.920 | 0.916 | 0.916 | 0.959 | 0.958 | 0.958 |
| Para | **0.941** | **0.937** | **0.937** | 0.920 | 0.916 | 0.916 | 0.959 | 0.958 | 0.958 |
| LM-Para | **0.941** | **0.937** | **0.937** | 0.920 | 0.916 | 0.916 | 0.959 | 0.958 | 0.958 |
| Del+WN | **0.941** | **0.937** | **0.937** | 0.920 | 0.916 | 0.916 | 0.959 | 0.958 | 0.958 |
| LM-Del+WN | **0.941** | **0.937** | **0.937** | 0.920 | 0.916 | 0.916 | 0.959 | 0.958 | 0.958 |
| Del+Para | **0.941** | **0.937** | **0.937** | 0.920 | 0.916 | 0.916 | 0.959 | 0.958 | 0.958 |
| LM-Del+Para | **0.941** | **0.937** | **0.937** | 0.920 | 0.916 | 0.916 | 0.959 | 0.958 | 0.958 |
| WN+Para | **0.941** | **0.937** | **0.937** | 0.920 | 0.916 | 0.916 | 0.959 | 0.958 | 0.958 |
| LM-WN+Para | **0.941** | **0.937** | **0.937** | 0.920 | 0.916 | 0.916 | **0.971** | **0.968** | **0.969** |
| Del+WN+Para | **0.941** | **0.937** | **0.937** | **0.929** | **0.926** | **0.927** | 0.959 | 0.958 | 0.958 |
| LM-Del+WN+Para | **0.941** | **0.937** | **0.937** | **0.929** | **0.926** | **0.927** | **0.971** | **0.968** | **0.969** |

Table 5.11: Results for binary classification using Short Answer Corpus

| | Feature | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Unigrams | | | Bigrams | | | Trigrams | | |
| Approach | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| NG | 0.602 | 0.600 | 0.586 | 0.680 | 0.684 | 0.677 | 0.622 | 0.642 | 0.628 |
| LM-NG | 0.628 | 0.621 | 0.598 | 0.676 | 0.684 | 0.677 | 0.634 | 0.653 | 0.639 |
| Del | — | — | — | 0.684 | 0.684 | 0.680 | 0.620 | 0.650 | 0.634 |
| LM-Del | — | — | — | 0.683 | 0.689 | 0.683 | 0.642 | 0.659 | 0.644 |
| WN | 0.635 | 0.642 | 0.633 | 0.699 | 0.693 | 0.690 | 0.674 | 0.684 | 0.677 |
| LM-WN | 0.656 | 0.663 | 0.656 | 0.708 | 0.715 | 0.705 | 0.694 | 0.695 | 0.685 |
| Para | 0.684 | 0.663 | 0.656 | 0.697 | 0.705 | 0.696 | 0.690 | 0.685 | 0.673 |
| LM-Para | 0.680 | 0.663 | 0.662 | 0.709 | 0.715 | 0.707 | 0.698 | 0.689 | 0.676 |
| Del+WN | — | — | — | 0.714 | 0.714 | 0.710 | 0.689 | 0.694 | 0.682 |
| LM-Del+WN | — | — | — | 0.729 | 0.724 | 0.719 | 0.692 | 0.700 | 0.689 |
| Del+Para | — | — | — | 0.717 | 0.715 | 0.710 | 0.690 | 0.696 | 0.686 |
| LM-Del+Para | — | — | — | 0.715 | 0.715 | 0.715 | 0.692 | 0.690 | 0.690 |
| WN+Para | 0.690 | 0.682 | 0.664 | 0.725 | 0.725 | 0.719 | 0.679 | 0.684 | 0.676 |
| LM-WN+Para | **0.685** | **0.673** | **0.672** | 0.730 | 0.736 | 0.726 | 0.693 | 0.695 | 0.680 |
| Del+WN+Para | — | — | — | 0.739 | 0.734 | 0.729 | 0.736 | 0.716 | 0.702 |
| LM-Del+WN+Para | — | — | — | **0.738** | **0.745** | **0.738** | **0.711** | **0.716** | **0.707** |

| | Feature | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Fourgrams | | | Fivegrams | | | Combined | | |
| Approach | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| NG | 0.633 | 0.653 | 0.636 | 0.674 | 0.674 | 0.665 | 0.658 | 0.663 | 0.658 |
| LM-NG | 0.643 | 0.653 | 0.642 | 0.680 | 0.684 | 0.677 | 0.675 | 0.674 | 0.669 |
| Del | 0.640 | 0.650 | 0.639 | 0.690 | 0.684 | 0.670 | 0.658 | 0.663 | 0.658 |
| LM-Del | 0.649 | 0.657 | 0.646 | 0.708 | 0.693 | 0.679 | 0.675 | 0.674 | 0.669 |
| WN | 0.688 | 0.676 | 0.660 | 0.713 | 0.695 | 0.677 | 0.679 | 0.684 | 0.671 |
| LM-WN | 0.694 | 0.670 | 0.663 | 0.719 | 0.695 | 0.680 | 0.684 | 0.695 | 0.679 |
| Para | 0.679 | 0.681 | 0.661 | 0.698 | 0.694 | 0.676 | 0.674 | 0.684 | 0.673 |
| LM-Para | 0.679 | 0.678 | 0.662 | 0.698 | 0.699 | 0.676 | 0.678 | 0.695 | 0.675 |
| Del+WN | 0.686 | 0.684 | 0.673 | 0.719 | 0.690 | 0.682 | 0.679 | 0.684 | 0.671 |
| LM-Del+WN | 0.693 | 0.695 | 0.675 | 0.713 | 0.688 | 0.683 | 0.684 | 0.695 | 0.679 |
| Del+Para | 0.701 | 0.690 | 0.676 | 0.708 | 0.700 | 0.680 | 0.674 | 0.684 | 0.673 |
| LM-Del+Para | 0.701 | 0.689 | 0.678 | 0.711 | 0.695 | 0.685 | 0.678 | 0.695 | 0.675 |
| WN+Para | 0.704 | 0.694 | 0.676 | 0.718 | 0.710 | 0.691 | 0.699 | 0.716 | 0.695 |
| LM-WN+Para | 0.707 | 0.690 | 0.680 | 0.724 | 0.705 | 0.693 | **0.708** | **0.716** | **0.705** |
| Del+WN+Para | 0.691 | 0.695 | 0.681 | 0.745 | 0.716 | 0.702 | 0.699 | 0.716 | 0.695 |
| LM-Del+WN+Para | **0.708** | **0.705** | **0.690** | **0.751** | **0.720** | **0.712** | **0.708** | **0.716** | **0.705** |

Table 5.12: Results for multi-classification using Short Answer Corpus

| Classified as | Near Copy | Light Revision | Heavy Revision | Non-Plagiarised |
|---|---|---|---|---|
| Near Copy | 10 | 2 | 5 | 2 |
| Light Revision | 5 | 7 | 7 | 0 |
| Heavy Revision | 0 | 8 | 9 | 2 |
| Non-Plagiarised | 0 | 0 | 1 | 37 |

Table 5.13: Confusion matrix for the multi-classification when “NG” approach is used with “Combined” features

amount of reused PA source text. Therefore, distinction between “Derived” and “Non Derived” documents is not clear and it is more difficult to identify reused documents.

Performance does not improve with deleted n-grams in the majority of cases. In addition, smaller improvements are obtained using the WN and Para approaches than with the METER Corpus. The reason is that performance of the baseline approach (NG) in this corpus is quite high which makes it difficult to get further improvement.

Similar to the METER Corpus, performance with unigrams is lower than using bigrams for the “NG” and “LM-NG” approaches. However, unigrams perform better than bigrams when the modified n-grams are added. Again this highlights the fact that unigrams can better identify the semantic relationship between a document pair compared to longer n-grams ($n \geq 2$).

Table 5.12 shows the results for multi-classification (four levels of plagiarism). A similar pattern of results is obtained compared to the multi-classification of the METER Corpus (see Table 5.9). The best result is obtained with the “LM-Del+WN+Para” approach, demonstrating the usefulness of using modified n-grams and assigning appropriate weights to n-grams. The highest $F_1$ score is obtained with bigrams ($F_1 = 0.738$). A combination of different lengths of n-grams (“Combined” approach) does not improve results compared to the best results with bigrams.

In contrast to binary classification (see Table 5.11), applying deleted n-grams improve the $F_1$ score. This is because results with the baseline approach (NG) are not high ($F_1 = 0.677$ for “Bigrams”) so there is room for improvement. Also, performance increases when modified n-grams are added using WordNet and Paraphrase Lexicon approaches.

Overall results for the multi-classification are lower than the binary classification for all approaches because it is difficult to differentiate between four levels of plagiarism compared to two (binary classification). Table 5.13 shows the confusion matrix for the multi-classification using "NG" approach with "Combined" features. It can be observed that it is hard to differentiate between three levels of plagiarism (near copy, light revision and heavy revision), although distinguishing between plagiarised and non-plagiarised is a simpler problem.

### 5.6.3 Results for MEDLINE Corpus

Table 5.14 shows results for binary classification with the MEDLINE corpus. Overall, results show a similar pattern to those observed for the binary classification for the METER (see Table 5.8) and the Short Answer (see Table 5.11) corpora. The best result is obtained with unigrams when modified n-grams generated with "UMLS" approach are used and weighted with language model probability scores ($F_1 = 0.907$). As the length of n-gram increases performance decreases and the combination of different lengths of n-grams ("Combined" approach) does not improve results. For all approaches, weighting n-grams gives better results than assigning equal weights to them. Improvement in performance is statistically significant for all approaches (except for the "Del" approach) compared to the baseline approach "NG" (Wilcoxon signed-rank test, $p < 0.05$).

Similar to the METER and the Short Answer corpora, modified n-grams generated with both the "Substitutions" and "Deletions" approaches improve results with a greater improvement observed with the "Substitutions" approach. A combination of these two approaches with n-gram weighting gives the best results for all cases apart except from unigrams.

|  | Feature |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | Unigrams |  |  | Bigrams |  |  | Trigrams |  |  |
| Approach | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| NG | 0.848 | 0.833 | 0.831 | 0.830 | 0.825 | 0.824 | 0.818 | 0.817 | 0.817 |
| LM-NG | 0.855 | 0.840 | 0.839 | 0.839 | 0.837 | 0.836 | 0.823 | 0.823 | 0.823 |
| Del | — | — | — | 0.832 | 0.825 | 0.824 | 0.820 | 0.819 | 0.819 |
| LM-Del | — | — | — | 0.842 | 0.838 | 0.838 | 0.833 | 0.833 | 0.833 |
| UMLS | 0.908 | 0.896 | 0.895 | 0.843 | 0.838 | 0.838 | 0.830 | 0.829 | 0.829 |
| LM-UMLS | **0.917** | **0.908** | **0.907** | 0.852 | 0.846 | 0.845 | 0.831 | 0.831 | 0.831 |
| Del+UMLS | — | — | — | 0.860 | 0.856 | 0.855 | 0.841 | 0.840 | 0.840 |
| LM-Del+UMLS | — | — | — | **0.864** | **0.860** | **0.859** | **0.845** | **0.844** | **0.844** |

|  | Feature |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | Fourgrams |  |  | Fivegrams |  |  | Combined |  |  |
| Approach | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| NG | 0.797 | 0.796 | 0.796 | 0.789 | 0.783 | 0.782 | 0.828 | 0.825 | 0.825 |
| LM-NG | 0.815 | 0.813 | 0.813 | 0.807 | 0.800 | 0.799 | 0.836 | 0.835 | 0.834 |
| Del | 0.802 | 0.802 | 0.802 | 0.787 | 0.785 | 0.784 | 0.833 | 0.829 | 0.828 |
| LM-Del | 0.831 | 0.831 | 0.831 | 0.806 | 0.802 | 0.801 | 0.842 | 0.840 | 0.840 |
| UMLS | 0.820 | 0.819 | 0.819 | 0.807 | 0.804 | 0.803 | 0.866 | 0.862 | 0.861 |
| LM-UMLS | 0.834 | 0.837 | 0.835 | 0.809 | 0.804 | 0.803 | 0.876 | 0.873 | 0.873 |
| Del+UMLS | 0.822 | 0.823 | 0.823 | 0.821 | 0.817 | 0.817 | 0.875 | 0.871 | 0.871 |
| LM-Del+UMLS | **0.839** | **0.837** | **0.839** | **0.825** | **0.821** | **0.821** | **0.887** | **0.885** | **0.884** |

Table 5.14: Results for binary classification using MEDLINE corpus

### 5.6.4 Summary of Results

The previous sections show that there are similarities between the results of the three corpora. The main findings of these results can be summarised as follows.

For both the binary- and multi-classification tasks, modified n-grams generated with the Substitutions and Deletions approaches contribute to improving results. The improvement observed using the Substitutions approach is higher than the Deletions approach. Further improvement is obtained by weighting n-grams with language model probability scores, indicating that assigning appropriate weights to n-grams helps to better discriminate between different levels of text reuse. For the majority of cases, the best result is obtained when the combination of deleted and substituted n-grams is used with n-gram weighting. This shows that assigning appropriate weights to n-grams and including modified n-grams are helpful in detecting reuse/plagiarism even when the text has been paraphrased.

For the binary classification task, the best results are obtained using unigrams for all three corpora. The improvement in performance is much higher for word unigrams than longer n-grams ($n \geq 2$) when the Substitutions approach is applied. This highlights the fact that this feature can capture the semantic similarity between suspicious-source document pairs better than longer n-grams ($n \geq 2$). It was also observed that different results are obtained with the baseline approach (NG) depending on the nature of the corpus. The $F_1$ score with the Short Answer Corpus is quite high compared to the other two corpora.

For the multi-classification task in the METER and the Short Answer corpora, the best results are obtained with bigrams indicating that this is the most appropriate feature to distinguish between different levels of reuse/plagiarism in these corpora. Results for multi-classification are lower than binary classification for all approaches.

Regarding the n-gram length, unigrams and bigrams give the best results for the binary- and multi-classification tasks respectively. Performance decreases as the length of n-gram increases. For both the binary- and multi-classification tasks, combining all five features (unigrams, bigrams, trigrams, fourgrams and fivegrams) does not improve performance. For all lengths of n-grams, the best results are obtained when n-grams are weighted and modified n-grams are applied.

## 5.7 Chapter Summary

A limitation of the standard n-gram overlap approach is it fails to identify paraphrased text. To overcome this a modified n-gram approach is proposed. Modified n-grams are generated by substituting words with synonyms and deleting words to capture different text editing operations. To give more weight to rare n-grams (containing more information) and less weight to frequent n-grams (containing less information) n-grams were weighted with language model probability scores. The problem of discriminating between different levels of text reuse (or plagiarism) was treated as a supervised document classification task. Two versions of the classification task were used: (1) binary classification and (2) multi-classification. Three standard datasets were used for evaluation: (1) METER Corpus, (2) Short Answer Corpus and (3) MEDLINE Corpus. Performance was measured using precision, recall and $F_1$ measures with macro-averaged reported across all classes.

Results showed modified n-grams generated using Deletions and Substitutions methods improve performance. Weighting n-grams using language model probability scores further improves performance. Best results are obtained when all types of modified n-grams are combined and weighted with probability scores. For the binary- and multi-classification tasks the best results were obtained with word unigrams and bigrams respectively.

# Chapter 6

# Conclusions

Text reuse is the process of creating new document(s) using text from existing document(s). Text reuse is standard practice in some situations, such as journalism, while it is not acceptable in others. Plagiarism, the unattributed reuse of text, is acknowledged as a serious problem in academia and in recent years cases of plagiarism have been reported to be on rise [Judge, 2008; McCabe, 2005; Park, 2003]. Consequently, the research community has explored the development of systems that can efficiently detect plagiarism.

The main aim of this research was to develop algorithms and techniques for mono-lingual text reuse detection. The particular focus was on detecting mono-lingual extrinsic plagiarism when the original text has been paraphrased. The detection of text reuse created with heavy paraphrasing is still in its infancy and an open challenge.

## 6.1 Thesis Summary

An IR-based framework was developed to retrieve candidate documents from large document collections (see Chapter 3). It was compared with a state-of-the-art approach, Kullback Leibler Distance, which gave promising results for reducing the plagiarism detection search space [Barrón-Cedeño et al., 2009]. Evaluation was carried out on three benchmark corpora: (1) PAN-PC-10 Corpus, (2) MEDLINE Corpus and (3) Extended Short Answer Corpus. Results showed that it is relatively straightforward to detect verbatim and slightly modified copies of texts but detecting paraphrased cases of text reuse is a difficult task. The proposed approach outperformed the baseline approach on all three corpora. It was also observed that the IR-based approach is more robust in identifying modified text than the baseline approach. Using the IR-based approach, the most appropriate length of the query was found to be a single sentence on all three corpora.

To identify text reuse when the original text has been paraphrased, query expansion was incorporated into the IR-based framework (see Chapter 4). Content words in the document which is suspected to contain reused text were expanded using two methods: (1) pseudo relevance feedback based on term co-occurrence statistics and (2) knowledge-based query expansion. For the latter case three lexicons were used: (*i*) WordNet, (*ii*) Paraphrase Lexicon and (*iii*) UMLS Metathesaurus. Results showed that pseudo relevance feedback does not improve results but query expansion based on knowledge bases improves candidate document retrieval performance on all three corpora. It was also observed that the choice of number of expansion terms and weights assigned to them effects retrieval performance.

The IR-based approach aims to quickly reduce the search space by identifying candidate documents from large reference collections. To carry out an exhaustive comparison of documents, a system was developed for pairwise comparison of documents to determine whether one document reused the other (see Chapter 5). In addition, this system could also be used to discriminate between different levels of text reuse/rewrite. The problem was cast as a supervised document classification task. Two versions of classification were used: binary classification (distinguish between two levels of reuse/rewrite) and multi-classification (distinguish between more than two levels of reuse/rewrite).

The proposed approach augments an n-gram overlap approach with modified n-grams generated by: (1) Substitutions (substituting a word with one of its synonymous words from WordNet, Paraphrase Lexicon or UMLS Metathesaurus) and (2) Deletions (delete word in an n-gram). Evaluation was carried out on three benchmark corpora: (1) METER Corpus, (2) Short Answer Corpus and (3) MEDLINE Corpus. To assign appropriate weights, n-grams were weighted with probability scores obtained by training a language model. Results showed that using modified n-grams improves performance. Weighting n-grams using language model probability scores further improves performance. The best results were obtained when all types of modified n-grams were combined and weighted with probability scores using word unigrams feature. Word unigrams and bigrams gave best results for the binary- and multi-classifications respectively.

### 6.1.1 Thesis Contributions

The main contributions of this thesis are:

1. Development of an IR-based framework for retrieving candidate documents from large document collections.
2. Incorporation of query expansion into the IR-based framework to deal with paraphrased cases of text reuse.
3. Exploration of lexical resources for text reuse detection.
4. Evaluation on a variety of benchmark corpora.
5. Development of a corpus for evaluating systems for the candidate document retrieval task.
6. Development of a system for pairwise document comparison using an n-gram overlap approach extended with modified n-grams.

## 6.2 Future Work

This thesis focussed on one area of text reuse/plagiarism detection (namely the detection of mono-lingual text reuse/plagiarism when the source has been paraphrased). The detection of text reuse is a wide area and there are a number of problems which still need to be addressed. Possible continuations or future avenues for this research include the following.

- **N-gram modeling for retrieving candidate documents.**

  The IR-based approach described in Chapters 3 and 4 is based on word unigram features. A possible extension would be to model n-grams, which may help to push the relevant source documents on top of the ranked list of retrieved documents. In addition, in ranking, multiple features can be used for computing the similarity score, for example, methods based on *rank fusion* can be explored [Fox and Shaw, 1994].

- **New approaches for generating modified n-grams and weighting them.**

  The modified n-gram approach generates a large number of n-grams and a reasonable number of modified n-grams are ungrammatical and unlikely to occur in the source text. In the current work, modified n-grams are generated based on “Substitutions” and “Deletions”. It would be interesting to explore ways to modify n-grams that are more linguistically motivated or simulate rewriting techniques people use. In addition, it might be useful to further investigate the effect of n-gram weighting on performance, for example, using Google N-gram Corpus[1] to assign weights to n-grams.

  Ungrammatical n-grams could be filtered by associating POS tag information to each word in an n-gram. If the POS tag sequence of the n-gram represents a

[1] http://googleresearch.blogspot.co.uk/2006/08/all-our-n-gram-are-belong-to-you.html Last Visited: 25-06-2012

valid grammatical sequence then it could be kept otherwise it would be discarded. Filtration of ungrammatical n-grams is likely to reduce the storage space and running time of the modified n-gram approach.

- **Editing operations used by people in rewriting.**

  It would be useful to carry out a study which investigates the kinds of editing operations (e.g. insertions, deletions, substitutions) used by people when they reuse text. The results of this analysis would allow us to model what people do in practice when they rewrite and then develop algorithms/techniques which could capture these.

- **Detailed comparison to identify suspicious-source section pairs.**

  The approaches explored in this thesis identify text reuse at document level. Another avenue of future work could be to make detailed comparison of the suspicious-source document pairs to identify the sections of text that are reused and their corresponding source sections.

  The paper [Nawab et al., 2010] describes our entry for the PAN 2010 Competition [Potthast et al., 2010b]. The Running Karp-Rabin Greedy String Tiling (RKR-GST) [Wise, 1993] algorithms (see Section 2.6.4) was used to identify suspicious-source section pairs (the detailed analysis stage). Results showed that this approach gave the best results in detecting simulated (manually paraphrased) cases of plagiarism in the PAN-PC-10 Corpus [Potthast et al., 2010a], which were most difficult to detect [Barrón-Cedeño, 2012]. However, this approach was not explored in-depth. A possible future direction will be to explore the RKR-GST approach more thoroughly for the detailed analysis of documents.

# References

E. Agirre and A. Soroa. Personalizing Pagerank for Word Sense Disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association of Computational Linguistics*, pages 33–41, 2009.

S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

S.M. Alzahrani, N. Salim, and A. Abraham. Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(2):133–149, 2011.

G. Amati and C.J. Van Rijsbergen. Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness. *ACM Transactions on Information Systems*, 20(4):357–389, 2002.

A.R. Aronson and F.M. Lang. An Overview of MetaMap: Historical Perspective and Recent Advances. *Journal of the American Medical Informatics Association*, 17(3): 229–236, 2010.

A.R. Aronson and T.C. Rindflesch. Query Expansion using the UMLS Metathesaurus. In *Proceedings of the AMIA Annual Fall Symposium*, pages 485–489. American Medical Informatics Association, 1997.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval - The Concepts and Technology Behind Search, Second Edition.* Pearson Education Ltd., Harlow, England, 2011.

J. Bagdis. Plagiarism Detection in Natural Language. Bachelor's Thesis, Princeton University, USA, 2008.

J. Bai, D. Song, P. Bruza, J.Y. Nie, and G. Cao. Query Expansion using Term Relationships in Language Models for Information Retrieval. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 688–695. ACM, 2005.

A. Barron-Cedeno, P. Rosso, D. Pinto, and A. Juan. On Cross-lingual Plagiarism Analysis Using a Statistical Model. In *ECAI 08 PAN Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 9–13, 2008.

A. Barrón-Cedeño, P. Rosso, and J.M. Benedi. Reducing the Plagiarism Detection Search Space on the Basis of the Kullback-Leibler Distance. In *Proceedings of 10th International Conference on Computational Linguistics and Intelligent Text Processing.*, pages 523–534. Springer, 2009.

L.A. Barrón-Cedeño. *On the Mono- and Cross-Language Detection of Text Re-Use and Plagiarism.* PhD Dissertation, Technical University of Valencia, Spain, 2012.

A. Bell. *The Language of News Media.* Blackwell Oxford, 1991.

M. Bendersky and W.B. Croft. Finding Text Reuse on the Web. In *Proceedings of* *the Second ACM International Conference on Web Search and Data Mining*, pages 262–271. ACM, 2009.

S. Bergsma, D. Lin, and R. Goebel. Web-scale N-gram Models for Lexical Disambiguation. In *Proceedings for the 21st International Joint Conference on Artificial Intelligence*, pages 1507–1512, 2009.

S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python.* O'Reilly Media, 2009.

R.F. Boisvert and M.J. Irwin. Plagiarism on the Rise. *Communications of the ACM*, 49(6):23–24, 2006.

S. Brin, J. Davis, and H. Garcia-Molina. Copy Detection Mechanisms for Digital Documents. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of Data, ACM*, pages 398–409, 1995.

A. Broder. On the Resemblance and Containment of Documents. In *Proceedings of the Compression and Complexity of Sequences*, pages 21–29. IEEE Computer Society, 1997.

S. Burrows, S.M.M. Tahaghoghi, and J. Zobel. Efficient and Effective Plagiarism Detection for Large Code Repositories. In *Proceedings of the Second Australian Undergraduate Students Computin Conference*, pages 8–15, 2004.

C. Callison-Burch. Syntactic Constraints on Paraphrases Extracted from Parallel Corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACM*, pages 196–205, 2008.

C. Callison-Burch, P. Koehn, and M. Osborne. Improved Statistical Machine Translation using Paraphrases. In *Proceedings of the Human Language Technology - North*

*American Chapter of the Association of Computational Linguistics*, pages 17–24. ACL, 2006.

C. Campbell. Writing with other's words: Using background reading text in academic compositions. In *In B. Kroll (Ed.) Second language writing: research insights for the classroom*, pages 211–230. Cambridge: Cambridge University Press, 1990.

Z. Ceska. *Automatic Plagiarism Detection Based on Latent Semantic Analysis.* PhD thesis, University of West Bohemia, Czech Republic, 2009.

Z. Ceska and C. Fox. The Influence of Text Pre-processing on Plagiarism Detection. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 55–59, 2009.

C.Y. Chang and S. Clark. Linguistic Steganography using Automatically Generated Paraphrases. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association of Computational Linguistics*, pages 591–599. ACL, 2010.

C.Y. Chen, J.Y. Yeh, and H.R. Ke. Plagiarism Detection using ROUGE and WordNet. *Journal of Computing*, 2:34–44, 2010.

S. Chiu, I. Uysal, and W.B. Croft. Evaluating Text Reuse Discovery on the Web. In *Proceeding of the Third Symposium on Information Interaction in Context*, pages 299–304. ACM, 2010.

M. Chong and L. Specia. Lexical Generalisation for Word-level Matching in Plagiarism Detection. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP)*, pages 704–709, 2011.

M. Chong, L. Specia, and R. Mitkov. Using Natural Language Processing for Automatic Detection of Plagiarism. In *Proceedings of the 4th International Plagiarism Conference (IPC-2010)*, 2010.

T.W.S. Chow and M.K.M. Rahman. Multilayer SOM with Tree-structured Data for Efficient Document Retrieval and Plagiarism Detection. *IEEE Transactions on Neural Networks*, 20(9):1385–1402, 2009.

P. Clough. Old and New Challenges in Automatic Plagiarism Detection: National Plagiarism Advisory Service, 2003a.

P. Clough. *Measuring Text Reuse.* PhD Dissertation, University of Sheffield, UK, 2003b.

P. Clough and R. Gaizauskas. *Corpora and Text Re-Use.* Corpus Linguistics (Series: Handbooks of Linguistics and Communication Science), Mouton de Gruyter, 2009.

P. Clough and M. Stevenson. Developing a Corpus of Plagiarised Short Answers. *Language Resources and Evaluation: Special Issue on Plagiarism and Authorship Analysis*, 45(1):5–24, 2011.

P. Clough, R. Gaizauskas, S. Piao, and Y. Wilks. Measuring Text Reuse. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics*, pages 152–159, 2002.

N. Cooke, L. Gillam, P. Wrobel, H. Cooke, and F. Al-Obaidli. A High Performance Plagiarism Detection System. In *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Notebook Papers of CLEF 11 Labs and Workshops, 2011.

M.R.J. Costa, R.E Banchs, J. Grivolla, and J. Codina. Plagiarism Detection Using Information Retrieval and Similarity Measures Based on Image Processing Techniques. In *Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*. Lab Report for PAN at CLEF 10, 2010.

T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

M.N. Do and M. Vetterli. Texture Similarity Measurement using Kullback-Leibler Distance on Wavelet Subbands. In *Proceedings of the International Conference on Image Processing*, pages 730–733. IEEE, 2000.

E.N. Efthimiadis. Query expansion. *Annual Review of Information Systems and Technology (ARIST)*, 31:121–187, 1996.

M. Elhadi and A. Al-Tobi. Duplicate Detection in Documents and Webpages using Improved Longest Common Subsequence and Documents Syntactical Structures. In *Fourth International Conference on Computer Sciences and Convergence Information Technology*, pages 679–684. IEEE, 2009.

M. Errami, J.D. Wren, J.M. Hicks, and H.R. Garner. eTBLAST: A Web Server to Identify Expert Reviewers, Appropriate Journals and Similar Publications. *Nucleic Acids Research*, 35:W12–W15, 2007.

M. Errami, J.M. Hicks, W. Fisher, D. Trusty, J.D. Wren, T.C. Long, and H.R. Garner. Déjà vu - A Study of Duplicate Citations in MEDLINE. *Bioinformatics*, 24(2): 243–249, 2008.

M. Errami, Z. Sun, A.C. George, T.C. Long, M.A. Skinner, J.D. Wren, and H.R. Garner. Identifying Duplicate Content using Statistically Improbable Phrases. *Bioinformatics*, 26(11):1453–1457, 2010.

H. Fang. A Re-examination of Query Expansion using Lexical Resources. In *Proceedings of Association for Computational Linguistics*, pages 139–147, 2008.

E. A. Fox and J. A. Shaw. Combination of Multiple Searches. *In Proceedings of TREC-2*, pages 243–249, 1994.

R. Gaizauskas, J. Foster, Y. Wilks, J. Arundel, P. Clough, and S. Piao. The METER Corpus: A Corpus for Analysing Journalistic Text Reuse. In *Proceedings of the Conference on Corpus Linguistics*, pages 214–223, 2001.

A. Ghosh, P. Bhaskar, S. Pal, and S. Bandyopadhyay. Rule Based Plagiarism Detection using Information Retrieval. In *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Notebook Papers of CLEF 11 Labs and Workshops, 2011.

Z. Gong, C. Cheang, and L. Hou U. Multi-term Web Query Expansion using WordNet. In *Database and Expert Systems Applications*, pages 379–388. Springer, 2006.

J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. Indexing with WordNet Synsets can Improve Text Retrieval. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, pages 647–678. Association for Computational Linguistics, 1998.

G. Grefenstette. *Explorations in Automatic Thesaurus Discovery.* Kluwer Academic Press, Boston, MA, 1994.

J. Grman and R. Ravas. Improved implementation for Finding Text Similarities in Large Collections of Data. In *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Notebook Papers of CLEF 11 Labs and Workshops, 2011.

C. Grozea and M. Popescu. Encoplot - Performance in the Second International Plagiarism Detection Challenge. In *Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Lab Report for PAN at CLEF 2010, 2010.

C. Grozea and M. Popescu. The Encoplot Similarity Measure for Automatic Detection of Plagiarism. In *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Notebook Papers of CLEF 11 Labs and Workshops, 2011.

C. GuangZhou, W. Long, and Z. Ling. A Cluster-Based Plagiarism Detection Method. In *Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Lab Report for PAN at CLEF 2010, 2010.

N. Gustafson, M.S. Pera, and Y.K. Ng. Nowhere to Hide: Finding Plagiarized Documents Based on Sentence Similarity. In *International Conference on Web Intelligence and Intelligent Agent Technology*, pages 690–696, 2008.

E. Hatcher, O. Gospodnetic, and M. McCandless. *Lucene in Action.* Manning Publications, 2004.

T.C. Hoad and J. Zobel. Methods for Identifying Versioned and Plagiarized Documents. In *Journal of the American Society for Information Science and Technology*, volume 54, pages 203–215, 2003.

S.M. Humphrey, W.J. Rogers, H. Kilicoglu, D. Demner-Fushman, and T.C. Rindflesch. Word Sense Disambiguation by Selecting the Best Semantic Type Based on Journal Descriptor Indexing: Preliminary Experiment. *Journal of the American Society for Information Science and Technology*, 57(1):96–113, 2006.

R.W. Irving. Plagiarism and Collusion Detection using the Smith-Waterman Algorithm. Computing Science Department, University of Glasgow, Research Report, TR-2004-164, 2004.

A. Johns and P. Myers. An Analysis of Summary Protocols of University ESL Students. *Applied Linguistics*, 11:253–271, 1990.

G. Judge. Plagiarism: Bringing Economics and Education Together (With a Little Help from IT). *Computers in Higher Education Economics Review*, 20(1):21–26, 2008.

D. Jurafsky and J.H. Martin. *Speech and Language Processing.* Prentice Hall, 2008.

J. Kasprzak and M. Brandejs. Improving the Reliability of the Plagiarism Detection System. In *Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Lab Report for PAN at CLEF 2010, 2010.

D. Kauchak and R. Barzilay. Paraphrasing for Automatic Evaluation. In *Proceedings of the North American Chapter of the Association of Computational Linguistics*, pages 455–462. ACL, 2006.

C. Keck. The use of paraphrase in summary writing: A comparison of l1 and l2 writers. *Journal of Second Language Writing*, 15:261–278, 2006.

P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of MT Summit*, 2005.

S. Kullback and R.A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

P.C.R. Lane, C. Lyon, and J.A. Malcolm. Demonstration of the Ferret Plagiarism Detector. In *Proceedings of the 2nd International Plagiarism Conference, Newcastle, UK*, 2006.

V.I. Levenshteiti. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. In *Soviet Physics-Doklady*, volume 10, 1966.

J. Lewis, S. Ossowski, J. Hicks, M. Errami, and H.R. Garner. Text Similarity: An Alternative Way to Search MEDLINE. *Bioinformatics*, 22(18):2298–2304, 2006.

C.Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Workshop on Text Summarization, Post-Conference Workshop of ACL*, pages 74–81, 2004.

S. Liu, F. Liu, C. Yu, and W. Meng. An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 266–272. ACM, 2004.

E. Loper and S. Bird. NLTK: The Natural Language ToolKit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, 2002.

K. Lu and X. Mu. Query Expansion using UMLS Tools for Health Information Retrieval. *Journal of the American Society for Information Science and Technology*, 46 (1):1–16, 2009.

C. Lyon, J. Malcolm, and B. Dickerson. Detecting Short Passages of Similar Text in Large Document Collections. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 118–125, 2001.

C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

C.D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*, volume 1. Cambridge University Press, UK, 2008.

B. Martin. Plagiarism: A Misplaced Emphasis. *Journal of Information Ethics*, 3(2): 36–47, 1994.

H. Maurer, F. Kappe, and B. Zaka. Plagiarism - A Survey. *Journal of Universal Computer Science*, 12(8):1050–1084, 2006.

D. McCabe. Research Report of the Center for Academic Integrity, 2005.

D.L. McCabe, K.D. Butterfield, and L.K. Trevino. Academic Dishonesty in Graduate Business Programs: Prevalence, Causes, and Proposed Action. *Academy of Management Learning and Education*, 5(3):1–294, 2006.

G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–312, 1990.

D.M. Mount. *Bioinformatics: Sequence and Genome Analysis, 2nd Edition.* Cold Spring Harbor Laboratory Press: Cold Spring Harbor, New York, 2004.

M. Mozgovoy, K. Fredriksson, D. White, M. Joy, and E. Sutinen. Fast Plagiarism Detection System. In *String Processing and Information Retrieval*, pages 267–270. Springer, 2005.

M. Mozgovoy, T. Kakkonen, and E. Sutinen. Using Natural Language Parsers in Plagiarism Detection. In *Proceedings of SLaTE'07 Workshop*, 2007.

M. Muhr, R. Kern, M. Zechner, and M. Granitzer. External and Intrinsic Plagiarism Detection using a Cross-Lingual Retrieval and Segmentation System. In *Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Lab Report for PAN at CLEF 2010, 2010.

R.M.A. Nawab, M. Stevenson, and P. Clough. University of Sheffield. In *Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Lab Report for PAN at CLEF 10, 2010.

R.M.A. Nawab, M. Stevenson, and P. Clough. External Plagiarism Detection using Information Retrieval and Sequence Alignment. In *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Notebook Papers of CLEF 11 Labs and Workshops, 2011.

R.M.A. Nawab, M. Stevenson, and P. Clough. Detecting Text Reuse with Modified and Weighted N-grams. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics*, pages 54–58. Association for Computational Linguistics, 2012a.

R.M.A. Nawab, M. Stevenson, and P. Clough. Retrieving Candidate Plagiarised Documents Using Query Expansion. In *Proceedings of the 34th European Conference on Information Retrieval (ECIR)*, pages 207–218. Springer, 2012b.

S.B. Needleman and C.D. Wunsch. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

G. Oberreuter, G. LHuillier, S.A. Ríos, and J.D. Velásquez. FASTDOCODE: Finding Approximated Segments of N-Grams for Document Copy Detection. In *Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*. Lab Report for PAN at CLEF 2010, 2010.

G. Oberreuter, G. LHuillier, S.A. Ríos, and J.D. Velásquez. Approaches for Intrinsic and External Plagiarism Detection. In *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*. Notebook Papers of CLEF 11 Labs and Workshops, 2011.

I. Ounis, G. Amati, Plachouras V., B. He, C. Macdonald, and Johnson. Terrier Information Retrieval Platform. In *Proceedings of the 27th European Conference on Information Retrieval*, pages 517–519. Springer, 2005.

Y. Palkovskii, A. Belov, and I. Muzyka. Using wordnet-based semantic similarity measurement in external plagiarism detection. In *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*. Notebook Papers of CLEF 11 Labs and Workshops, 2011.

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

C. Park. In Other (People's) Words: Plagiarism by University Students – Literature and Lessons. *Assessment & Evaluation in Higher Education*, 28(5):471–488, 2003.

W.R. Pearson. Rapid and Sensitive Sequence Comparison With FASTP and FASTA. *Methods in Enzymology*, 183:63–98, 1990.

H.J. Peat and P. Willett. The Limitations of Term Co-occurrence Data for Query Expansion in Document Retrieval Systems. *Journal of the American Society for Information Science*, 42(5):378–383, 1991.

D. Pinto, J.M. Benedí, and P. Rosso. Clustering Narrow-Domain Short Texts by Using the Kullback-Leibler Distance. In *Computational Linguistics and Intelligent Text Processing*, pages 611–622. Springer, 2007.

M.F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980.

M. Potthast. *Technologies for Reusing Text from the Web*. PhD Dissertation, Bauhaus-Universität Weimar, 2011.

M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso. An Evaluation Framework for Plagiarism Detection. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 997–1005, 2010a.

M. Potthast, B. Stein, A. Eiselt, A. Barrón-Cedeño, and P. Rosso. Overview of the 2nd International Competition on Plagiarism Detection. In *Proceedings of the CLEF10 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, 2010b.

M. Potthast, A. Eiselt, A. Barrón-Cedeño, B. Stein, and P. Rosso. Overview of the 3rd

International Competition on Plagiarism Detection. In *Notebook Papers of CLEF 11 Labs and Workshops*, 2011.

Y. Qiu and H.P. Frei. Concept Based Query Expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 160–169. ACM, 1993.

S. Rao, P. Gupta, K. Singhal, and P. Majumder. External & Intrinsic Plagiarism Detection: VSM & Discourse Markers based Approach. In *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse.* Notebook Papers of CLEF 11 Labs and Workshops, 2011.

S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. Statistical Machine Translation for Query Expansion in Answer Retrieval. In *Proceedings of the 45th Annual Meeting of Association of Computational Linguistics*, pages 464–471, 2007.

R. Rivest. The MD5 Message-Digest Algorithm. *Request for Comments 1321, Network Working Group, ISI*, 1992.

J.J. Rocchio. Relevance feedback in Information Retrieval. In *The SMART retrieval system: experiments in automatic document processing*, pages 313–323, 1971.

T. Rose, M. Stevenson, and M. Whitehead. The Reuters Corpus Volume 1 - from Yesterday's News to Tomorrow's Language Resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, pages 827–832, 2002.

P. Runeson, M. Alexandersson, and O. Nyholm. Detection of Duplicate Defect Reports using Natural Language Processing. In *Proceedings of the 29th international conference on Software Engineering*, pages 499–510. IEEE Computer Society, 2007.

G. Salton, A. Wong, and C.S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, 1975.

D. Sankoff and J.B. Kruskal. Time Warps, String Edits, and Macro Molecules: The Theory and Practice of Sequence Comparison. *Addison-Wesley Publication*, 1983.

V. Scherbinin and S. Butakov. Using Microsoft SQL Server Platform for Plagiarism Detection. In *25th Annual Conference of the Spanish Society for Natural Language Processing (SEPLN)*, 2009.

J. Seo and W.B. Croft. Local Text Reuse Detection. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 571–578. ACM, 2008.

J. Sheard, M. Dick, S. Markham, I. Macdonald, and M. Walsh. Cheating and Plagiarism: Perceptions and Practices of First Year IT Students. In *ACM SIGCSE Bulletin*, volume 34, pages 183–187. ACM, 2002.

N. Shivakumar and H. Garcia-Molina. SCAM: A Copy Detection Mechanism for Digital Documents. In *Proceedings of the 2nd Annual Conference on the Theory and Practice of Digital Libraries, Texas, USA*, 1995.

A.F. Smeaton and C.J. Van-Rijsbergen. The Retrieval Effects of Query Expansion on a Feedback Document Retrieval System. *The Computer Journal*, 26(3):239–246, 1983.

T.F. Smith and M.S. Waterman. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

E. Stamatatos. A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556, 2009.

B. Stein. Fuzzy-Fingerprints for Text-Based Information Retrieval. In *Proceedings of the 5th International Conference on Knowledge Management*, pages 572–579, 2005.

B. Stein, S.M. Eissen, and M. Potthast. Strategies for Retrieving Plagiarized Documents. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 825–826, 2007.

B. Stein, P. Rosso, E. Stamatatos, M. Koppel, and E. Agirre. 3rd PAN Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. In *25th Annual Conference of the Spanish Society for Natural Language Processing (SEPLN)*, pages 1–77, 2009.

A. Stolcke. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, 2002.

Z. Su, B.R. Ahn, K.Y. Eom, M.K. Kang, J.P. Kim, and M.K. Kim. Plagiarism Detection using the Levenshtein Distance and Smith-Waterman Algorithm. In *3rd International Conference on Innovative Computing Information and Control*, pages 569–572, 2008.

D.A.R. Torrejón and J.M.M. Ramos. Crosslingual CoReMo System. In *Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*. Notebook Papers of CLEF 11 Labs and Workshops, 2011.

O. Uzuner, B. Katz, and T. Nahnsen. Using Syntactic Information to Identify Plagiarism. In *Proceedings of the 2nd Workshop on Building Educational Applications using Natural Language Processing*, pages 37–44. ACL, 2005.

C. Vania and M. Adriani. Automatic External Plagiarism Detection Using Passage Similarities. In *Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*. Lab Report for PAN at CLEF 10, 2010.

T. Wang, X.Z. Fan, and J. Liu. Plagiarism Detection in Chinese Based on Chunk and Paragraph Weight. In *International Conference on Machine Learning and Cybernetics*, pages 2574–2579. Kunming, 2008.

W. Wang, A. Stolcke, and J. Zheng. Reranking Machine Translation Hypotheses with Structured and Web-based Language Models. In *IEEE Workshop on Automatic Speech Recognition & Understanding.*, pages 159–164. IEEE, 2007.

D.R. White and M.S. Joy. Sentence-based Natural Language Plagiarism Detection. *Journal on Educational Resources in Computing*, 4(4):1–20, 2004.

F. Wilcoxon, S.K. Katti, and R.A. Wilcox. Critical Values and Probability Levels for the Wilcoxon Rank Sum Test and the Wilcoxon Signed Tank Test. *Selected Tables in Mathematical Statistics*, 1:171–259, 1973.

M. Wise. Running Karp-Rabin Matching and Greedy String Tiling. Technical Report 463, Technical Report, Basser Department of Computer Science, University of Sydney, 1993.

M.J. Wise. Neweyes: A System for Comparing Biological Sequences using the Running Karp-Rabin Greedy String-Tiling Algorithm. In *3rd International Conference on Intelligent Systems for Molecular Biology*, pages 393–401, 1995.

M.J. Wise. YAP3: Improved Detection of Similarities in Computer Program and Other Texts. In *Proceedings of the 27th SIGCSE Technical Symposium on Compute Science Education, Philadelphia, USA*, pages 130–134, 1996.

H. Zhang and T.W.S. Chow. A Coarse-to-Fine Framework to Efficiently Thwart Plagiarism. *Pattern Recognition*, 44(2):471–487, 2011.

M. Zini, M. Fabbri, M. Moneglia, and A. Panunzi. Plagiarism Detection Through Multilevel Text Comparison. In *2nd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution*, pages 181–185, 2006.