

國立交通大學

資訊管理研究所

碩士論文

以 ROUGE 和 WordNet 為基礎的 N-gram 共現於

剽竊偵測

Plagiarism Detection using N-gram Co-occurrence

Statistics Based on ROUGE and WordNet

研究生：陳建穎

指導教授：柯皓仁 博士

中華民國九十七年七月

Plagiarism Detection using N-gram Co-occurrence Statistics Based on ROUGE and WordNet

Student: Chien-ying, Chen

Advisor: Professor Hao-ren, Ke

Institute of Information Management

National Chiao Tung University

Abstract

With the arrival of Digital Era and the Internet, control of information flow is nearly impossible; the lack of control provides an incentive for Internet users and computer owners to freely copy and paste any content available to them. Plagiarism often occurs when users fail to credit the original owner for the content borrowed, and such behavior leads to violation of intellectual property.

Two main approaches to plagiarism detection are fingerprinting and term occurrence. Although these two approaches have yielded considerable results, they are not without faults. One common weakness suffered by both approaches, especially fingerprinting, is the incapability to detect modified text plagiarism. This research proposed adoption of ROUGE and WordNet. The former includes n-gram co-occurrence statistics, skip-bigram, and longest common subsequence (LCS), while the latter acts as a thesaurus dictionary, which also provides semantic information. N-gram co-occurrence statistics can detect verbatim copy and certain sentence structural changes, skip-bigram and LCS is immune from text modification such as simple addition or deletion of words, and WordNet may handle the problem of word

substitution.

The proposed methods have been tested on two manually created corpora, *abstract* set and *paraphrased* set. Empirically derived threshold and preprocessing setting for each method are recommended based on the evaluation of the performance. Different types of plagiarism examples are shown to support the statements made about the strengths and weaknesses of the proposed methods.

Keywords: plagiarism detection; ROUGE; WordNet; n-gram co-occurrence statistics



以 ROUGE 和 WordNet 為基礎的 N-gram 共現於剽竊 偵測

研究生：陳建穎

指導教授：柯皓仁

國立交通大學資訊管理研究所 碩士班

摘要

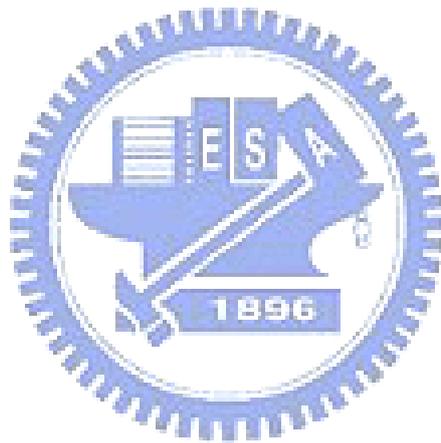
隨著數位時代的到來和網際網路的蓬勃發展，對於資訊流的控制幾乎是不可能的。而在資訊缺乏管制的情况下，網路和電腦使用者可以隨意地複製並使用任何他們能取得的資訊內容。但是如果在使用時，沒有列出資料的出處和其智慧財產的擁有者，那麼此舉就會形成剽竊而侵犯了智慧財產權。

目前大多數的剽竊偵測方法分成 fingerprinting 和 term occurrence。雖然兩種方法在剽竊偵測的領域裡已有一定的成果，它們還是有不足之處。刻意針對原文做修改就會影響上述方法對於剽竊偵測的表現，尤其是 fingerprinting 受其影響甚鉅。因此，本論文提出了套用了 ROUGE 和 WordNet 來偵測剽竊的演算法，因為前者包括了 n-gram co-occurrence statistics、skip-bigram 和 longest common subsequence (LCS)，而後者有著同義詞典的功能也提供詞意上的資訊。N-gram co-occurrence statistics 可以有效地偵測照抄和更動句子結構的剽竊，skip-bigram 和 LCS 則不會受到純粹地新增詞彙於原文中或部分原文被刪除的影響，而運用 WordNet 則得以偵測用同義詞替換原文的情形。

本論文用兩組以人力做成的資料集(稱之為 *abstract* 和 *paraphrased*)，來評估方法的效果。每個方法都依實驗結果的觀察來推薦適合的標準值和前置處理的

設定。最後，由幾個不同類型的剽竊例子來支持先前對於每個方法的強項和弱點的假設。

關鍵詞：剽竊偵測； ROUGE; WordNet; n-gram 共現



Acknowledgements

First of all, I would like to thank my instructor, Professor Hao-Ren Ke, for his guidance for the past two years. His unique sense of humor had made the entire learning process easier and more enjoyable. Despite his friendliness, he managed to provide valuable advices from time to time. It has been a pleasure to be his student. Second, I would like to thank my senior, Jen-Yuan Yeh, who had been guiding me for the last eight months of my Master's program. He showed me the right way of doing research and educated me along the way. Third, I would like to thank my fellow classmates who had helped me during the past two years, especially Jian-Quan, Huang Qiang, Zhen-Dong, and my lab mates Yi-Xiang and You-Ying. They offered their help when I encountered problems. I learned and gained knowledge each time they solved the problems and explained to me the causes patiently. I am really happy and grateful to be able to know the members of IIM. I also want to thank them for the valuable memories I share with them. Besides the students of IIM, I also would like to thank Shu-Hui and Xin-Xin who were of great help when I first started my Master's program and for the following two years as well. Finally, I would like to thank my family for their supports, especially my parents. If not for them, I will never have the chance of experiencing all these.

Contents

Abstract.....	I
摘要.....	III
Acknowledgements	V
Contents	VI
List of Figures.....	VIII
List of Tables.....	X
List of Tables.....	X
1. Introduction.....	1
1.1 Background	1
1.2 Motivation and Objective.....	3
2. Related Work.....	6
2.1 Existing Methods.....	6
2.1.1 Fingerprinting	6
2.1.2 Term Occurrence.....	10
2.1.3 Style Analysis.....	12
2.2 ROUGE.....	14
2.3 WordNet.....	15
3. Methodology	19
3.1 System Architecture for Plagiarism Detection	19
3.2 Preprocessing.....	20
3.2.1 Tokenization and Sentences	20
3.2.2 Part-of-Speech (POS) Tagger.....	22
3.2.3 Punctuation Removal and Lowercasing	23
3.2.4 Stopwords Removal	23
3.2.5 Stemming	24
3.3 Plagiarism Detection Methods.....	25
3.3.1 ROUGE-N.....	26
3.3.1.1 Unigram	26
3.3.1.2 N-grams.....	28
3.3.2 Longest Common Subsequence (LCS).....	28
3.3.3 Skip-Bigram.....	30
3.3.4 WordNet.....	31
3.3.4.1 Synonyms-based.....	32
3.3.4.2 Relationship-based.....	35
3.3.5 Google Mutual Information (MI).....	37
3.3.6 Caching	38

4. Experiments and Evaluation	40
4.1 Data Sets	40
4.2 Experiments	45
4.3 Evaluation	48
4.3.1 Recommended Settings – ROUGE-based Methods	48
4.3.2 Recommended Settings - WordNet	52
4.3.3 Evaluation of WordNet-based Methods for <i>Abstract Set</i>	58
4.3.4 Evaluation of WordNet-based Methods for <i>Paraphrased Set</i>	61
4.3.5 Evaluation of Google MI Method for <i>Abstract and Paraphrased</i> Sets	62
4.3.6 Strengths and Weaknesses of Each Method	65
5. Conclusion	71
Bibliography:	74
Appendix 1 Line Graphs of Bigram to LCS:	78
Appendix 2 Partial Line Graphs of Bigram to LCS:	81
Appendix 3 32 Plagiarism Examples in <i>Abstract Set</i>:	84



List of Figures

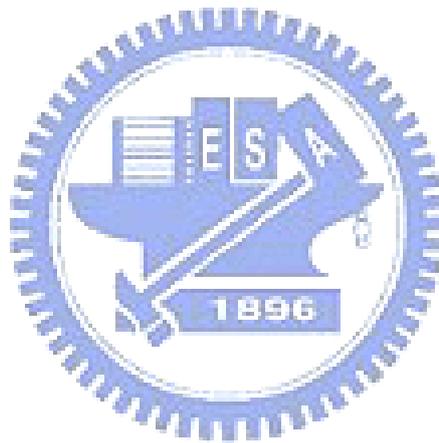
Figure 1 Classification of Detection Methods	6
Figure 2 Fingerprint Formation [32].....	7
Figure 3 Shingles of a String [2].....	9
Figure 4 Document Tree [28].....	11
Figure 5 Search Result of “fly” in WordNet.....	15
Figure 6 Lexical and Semantic Functions Available for “fly”.....	17
Figure 7 Hypernym Hierarchy of “fly” in WordNet.....	18
Figure 8 System Architecture	20
Figure 9 Unprocessed Text	22
Figure 10 Text Divided into Sentences.....	22
Figure 11 Words with Corresponding POS Tag.....	23
Figure 12 Text with Stopwords Removed.....	24
Figure 13 Stemmed Text	25
Figure 14 Example of Clipped Precision	27
Figure 15 Examples of LCS.....	29
Figure 16 Example of Skip-bigrams Formation	30
Figure 17 Example of Jaccard’s Coefficient between Two Synsets	33
Figure 18 Example of Hypernym/Hyponym Relationship between Two Words	36
Figure 19 Statistics of Each Plagiarism Type in the <i>Abstract</i> Set	43
Figure 20 Ling Graph of Unigram under Different Preprocessing Settings	49
Figure 21 Partial Graph of Figure 19.....	50
Figure 22 Hypernym Relationship in WordNet for Play and Ownership	54
Figure 23 Hypernym Relationship in WordNet for Support and Provide....	54
Figure 24 F-Measures of Three Schemes with SW Removed.....	56
Figure 25 F-Measures of Three Schemes with No Preprocessing.....	56
Figure 26 2 nd Weighting Scheme under Different Settings.....	57
Figure 27 Synonyms-based Method under Different Settings.....	58
Figure 28 F-Measures Comparison with Stopwords Removed for WordNet-based Methods.....	59
Figure 29 F-Measures Comparison with No Preprocessing for WordNet-based Methods.....	59
Figure 30 Comparison Graph for <i>Abstract</i> Set with Stopwords Removed for WordNet-based Methods.....	60
Figure 31 Comparison Graph for <i>Abstract</i> Set with No Preprocessing for WordNet-based Methods.....	60

Figure 32 TPs of Unigram and WordNet-based Methods with Stopwords
Removed.....61

Figure 33 TPs of Unigram and WordNet-based Methods with No
Preprocessing62

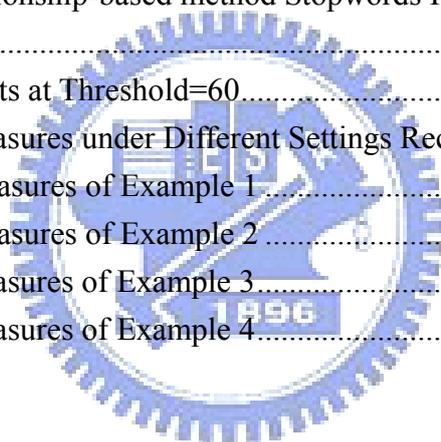
Figure 34 Comparison Graph for *Abstract* Set with Stopwords Removed
Including Google Method63

Figure 35 Comparison Graph for *Paraphrased* Set with Stopwords
Removed Including Google Method.....64



List of Tables

Table 1 Kappa Statistics [16]	41
Table 2 Example of Verbatim Copy	43
Table 3 Example of Substantial Verbatim.....	43
Table 4 Example of Lifted Sentences	44
Table 5 Example of Paraphrased but Same Key Words.....	44
Table 6 Examples of Sensitivity, Specificity, and F-measure	47
Table 7 F-measures of Unigram under Different Settings	49
Table 8 Results of F(SW+SM).....	51
Table 9 Results of F(SW).....	51
Table 10 Summarization Table of Recommended Settings	52
Table 11 Relationship-based method Stopwords Removed (Initial Weighting Scheme).....	53
Table 12 Relationship-based method Stopwords Removed (2 nd Weighting Scheme).....	55
Table 13 Results at Threshold=60.....	63
Table 14 F-measures under Different Settings Recommended Threshold...65	
Table 15 F-Measures of Example 1.....	66
Table 16 F-Measures of Example 2	67
Table 17 F-measures of Example 3.....	68
Table 18 F-measures of Example 4.....	69



1. Introduction

1.1 Background

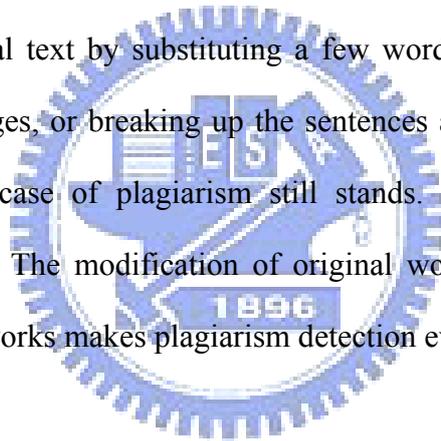
Looking up plagiarism in some of the dictionaries, one will find different definitions. Though slightly different from one another, these definitions convey an identical idea – plagiarism is the use of other people’s work/idea as one’s own without crediting the original owner. This kind of behavior is equivalent to stealing. Nevertheless, cases of plagiarism are still being reported in classes and even in academic research. Maurer et al. [22] described the policies against plagiarism in some of the most prestigious universities and how each of the schools handles such misconduct; some of these universities were seeing increasing number of reported plagiarism cases, including Web plagiarism, in recent years that ranged from 2003 to 2006.

Plagiarism can occur unintentionally or intentionally. Unintentional plagiarism is caused by the lack of understanding about plagiarism, as the offender does not know that what he/she has done constitutes plagiarism. As there is a saying – prevention is better than cure, introducing students to the concept of plagiarism and educating them to avoid plagiarism is crucial before they start their first research-oriented writing assignment. The importance of preaching the right idea about plagiarism is illustrated in [8], as one of the authors shared her own experience on students’ understanding about plagiarism. In her story, she introduced the concept of plagiarism to the students and taught them how to avoid it in the beginning of a course. Yet, three students plagiarized in an assignment; and when asked to see her anonymously, other than the three offenders another 11 students showed up as they were not sure if they were one of the students. Despite the initial effort, some students still chose to plagiarize while some were still not clear on the issue of plagiarism. Perhaps, even more time is

needed to be spent on defining plagiarism and teaching the students about citations and reference of borrowed work, so that students can develop the right research writing habit from the beginning. At the same time, complementary work can be done. For example, Maurer et al. mentioned how universities provide online resources such as tutorials, or brochures for educational purposes. There is nothing wrong to borrow and refer to existing work of other people because this is an essential step of learning; however, one just has to credit the rightful owners for their contribution in a right way.

When all precautions have been carried out to prevent plagiarism, there is still no guarantee that plagiarism will stop because there are people who plagiarize intentionally. Students who choose to plagiarize probably do not take his/her work seriously, and they do not know much about the subject they are plagiarizing, that is why they use existing information directly. Violation of copyright does not really bother them, what is more important is to complete the assignment and hand it in on time. On the contrary, some students plagiarize because they are serious about their assignments in terms of the grades they receive. As a result, they look for existing works, which are probably better than what they can come up with by themselves and hand in as their works. A survey done on Year 11 high school students suggests that those who plagiarized the most cared more about grades than the learning process [23]. There are plagiarism cases in which students downloaded papers from the Internet and turned in the exact papers as their own work. Unfortunately, cases of plagiarism are being reported by research journals too. These researchers whether graduate students or professors, should know more about the seriousness of plagiarism than the undergraduate students, and should be more self disciplined. Nonetheless, some of them decided to include other researchers' results in their works and publish the papers.

One tempting factor may be the reason for plagiarism – convenience. Especially with the arrival of digital era, electronic storage hardware has replaced papers to become the media for data keeping and channeling, and plagiarism has never been easier. The situation worsens with the invention of the Internet, which becomes a huge resource center for searching information and retrieving the information at one’s fingertips. Another cause of plagiarism besides convenience may be the lack of control for online distribution of copyrighted contents. The number of Web pages available and the amount of data flow on the Internet make scrutinized data transfer infeasible. Almost nonexistent control encourages Internet users to freely copy the information they want and use it as if it is their property. Even if a person makes some alterations to the original text by substituting a few words with synonyms, making some grammatical changes, or breaking up the sentences and inserting into different parts of his/her work, case of plagiarism still stands. These are some common examples of plagiarism. The modification of original work together with naturally similar non-plagiarism works makes plagiarism detection even more difficult.



1.2 Motivation and Objective

Copyright protection has always been an issue, especially when digital technology shortens the time for both duplication and distribution. Copyright laws cover a wide range of categories such as music, video, software, books and many other fields. The topic of this research will focus on the field of text plagiarism.

Not only does plagiarism violate copyright regulation, but also influence the quality of education and research. Knowledge is accumulated through learning and thinking, and school assignments force students to learn and think during the process of completing the assignments. Whether the students choose to look up resources in

the Internet or printed articles, or come up with innovative ideas through brainstorming, either step benefits students who put in efforts and time as they gain something new with each assignment. However, plagiarism deprives students of undergoing such process as they spend less time to think. Even if students do read the content before they plagiarize, they are most likely to forget about the content faster than those who genuinely do their work. This reasoning is not without support as the observation from a case study is in accordance with the above opinion. The observer stated that "...when asked about his learning, Brett was unable to recall anything about his topic." [23].

In the academic research domain, no new discoveries will be made if the researchers only reuse existing information. Collberg et al. [7] focused on self-plagiarism and argued that self-plagiarism causes new but similar papers to be published without contributing to the overall advancement of academic research.

With the problem of plagiarism being taken seriously, considerable amount of research has been done in detecting plagiarism. The approaches mainly focus on the calculation of document similarity through analyzing the content of the texts. Content may be referred to words, sentences or paragraphs in the texts, or even the intrinsic structure of the texts. Besides academic research on plagiarism, there are online detection services and tools available. Most of the better and more established services are for staff of educational institutions to examine suspicious works from the students. Individual service is also available. However, majority of the services are not free of charge while the remaining options may not be as effective. [22] acts as a gateway to learn more about the available tools as the authors provided a rather detail introduction to the three following tools, Turnitin, SafeAssignment, and Docol©c; the authors also briefly summarized several other tools, a couple of which are tools that detect software and program source code plagiarism.

Although there are different plagiarism detection approaches, each method has its pros and cons. One common weakness is the vulnerability to text modification that can be achieved through addition, deletion and substitution of words, and also change of sentence structure or word order. In this research, we propose a prototype of a system that adopts ROUGE and utilizes WordNet (a thesaurus-like dictionary), in hope of combining the strengths of individual method and overcoming the disadvantages of each separate method. Generally, the proposed methods should be able to conquer most of the text alteration strategies mentioned earlier. We will discuss in more detail about related work in Chapter 2, methodology in Chapter 3, experiments and evaluation in Chapter 4, and conclusion in Chapter 5.



2. Related Work

Until now, quite a considerable amount of research has focused on plagiarism detection. Figure 1 provides an overview about the development of plagiarism detection. The classification is derived from the taxonomy in [29]. As Figure 1 indicates, the plagiarism detection methods can be categorized into three main categories: fingerprinting, term occurrence, and style analysis.

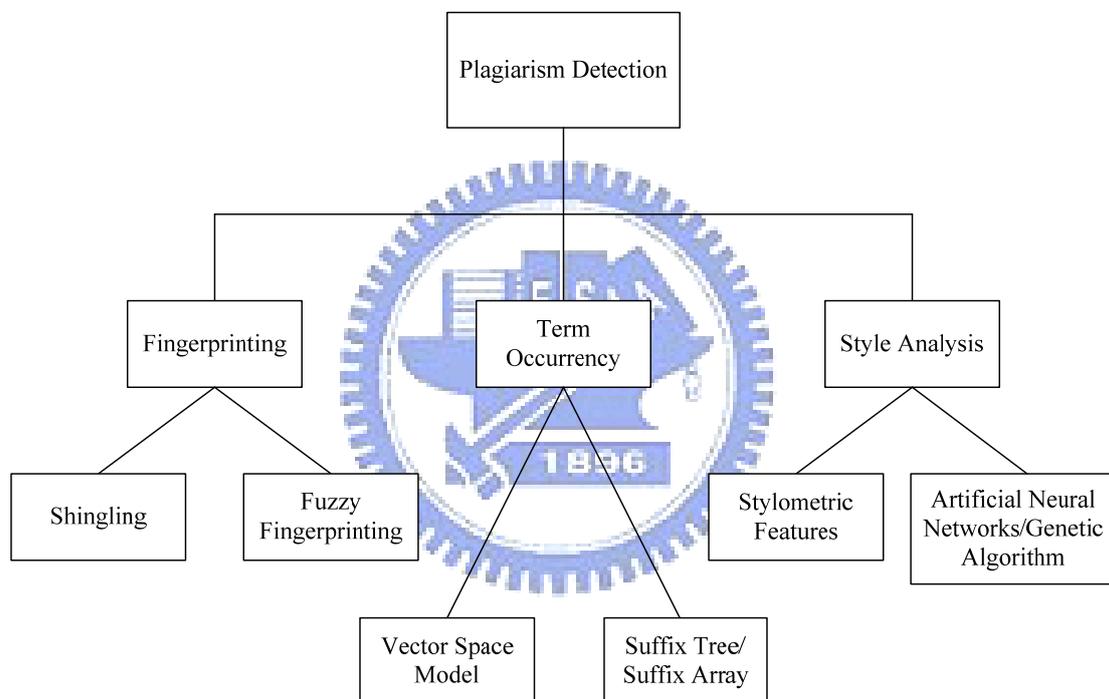


Figure 1 Classification of Detection Methods

2.1 Existing Methods

2.1.1 Fingerprinting

Fingerprinting can be considered as the most widely adopted approach in plagiarism detection. The origin of this method, as suggested by previous studies, is attributed to the work done by Udi Manber [21]. In that research, Manber aimed to find out similar documents in a database. The research was based on Rabin

Fingerprint scheme, which was applied to generate a unique identity (fingerprint) for each document. Rabin fingerprint scheme or hash function as often used interchangeably, can transform a sequence of substring into an integer. And a good scheme/function should generate the same integer for the same substring; on the other hand, it should generate different integer for each unique substring to ensure consistency and avoid undesirable collisions of fingerprints. Furthermore, there are other factors that need to be considered when generating fingerprints. They are fingerprint granularity, fingerprint selection, and fingerprint resolution. These issues are discussed in greater detail in [11]. Figure 2 below illustrates how a fingerprint is formed, followed by a summarization of the three factors.

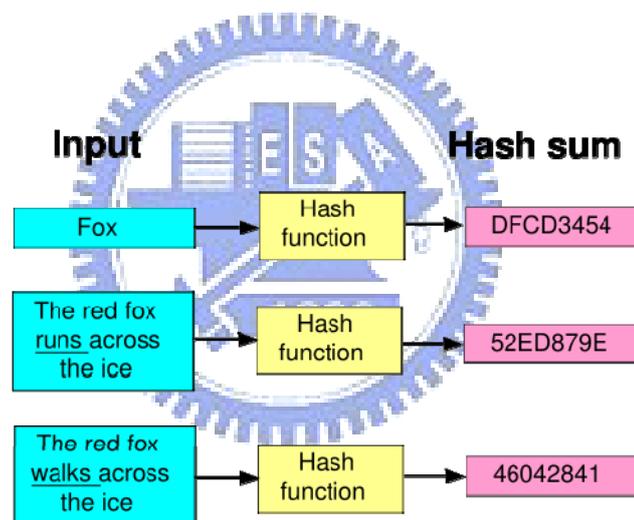


Figure 2 Fingerprint Formation [32]

First, fingerprint granularity means the amount of data a fingerprint represents. Observing from Figure 2, a fingerprint/hash can be either generated from a word or a string of text, and a word-level fingerprint has higher granularity than a sentence-level fingerprint. From a different perspective, granularity also means how much information must be exactly the same between two documents for the respective fingerprints to match. In other words, granularity defines the “fineness” of detection

because high granularity fingerprints can match in smaller portions of overlapping text. However, as high granularity fingerprints have a greater chance of finding a match, it will lead to higher similarity or even false positive between two documents.

Second, fingerprint selection means how the substrings of a text are chosen before being transformed into fingerprints to represent the documents. The more accurately fingerprint(s) can represent a document the better; therefore, selection of the most representative substrings in the text is important as the effectiveness and reliability of detection will be affected. Some selection schemes are available and they include but not limited to full fingerprinting [1], random fingerprinting adopted by [21] and [5], and selective fingerprinting as in [10].

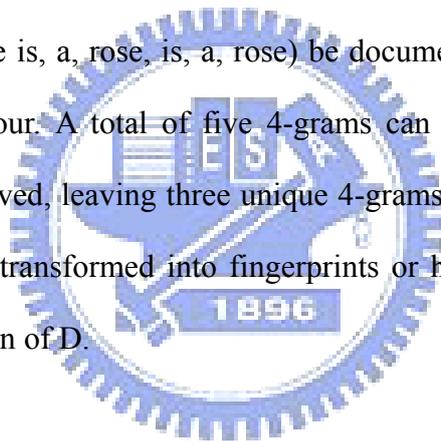
Third, fingerprint resolution means how many fingerprints are used to represent a document. When more fingerprints are included, especially those unique and important ones, matching of the fingerprints becomes more meaningful and the similarity between two documents is truly reflected.

The above three factors influence the efficiency and accuracy of the matching process directly. Each of the choices, higher granularity, full fingerprinting scheme, and higher resolution, leads to more memory consumption. Although better results may be obtained, processing time will be longer. Hence, each factor should be adjusted to best suit the need for different purposes.

Fingerprinting was first applied to the field of plagiarism detection in COPS [1] for copy detection in digital documents. In COPS, the smallest detection unit was a sentence, but multiple sentences could form a larger unit of detection called chunk. The research included a document database to save newly processed documents and compare a suspect document with registered documents. Later, SCAM (Stanford Copy Analysis Mechanism) [27] was developed based on the foundation of [1]. Different from [1], SCAM focused on word-based overlap as fingerprints were

generated in unit of word instead of sentence. The change led to better performance in detecting partial copy but more false positives as a tradeoff. Majority of later approaches focus on various aspects of fingerprinting; usually, different strategies are deployed or other techniques are integrated with fingerprinting. Variations include shingling and fuzzy fingerprinting. The former is a combination of fingerprinting and unique n-gram substrings [2] while the latter adopts inverse document frequency (idfs) into substring selection [6] to pick out feature terms. In fact, the concept behind these two methods is the same.

The concept behind shingling can be understood through brief descriptions of both methods. First, Figure 3 is a simplified example of how shingles of a document are obtained. Let (a, rose is, a, rose, is, a, rose) be document D, and the size of each shingle/n-gram, w , be four. A total of five 4-grams can be obtained from D. Any repeated 4-gram is removed, leaving three unique 4-grams, which are the shingles of D. The shingles can be transformed into fingerprints or hash values, and altogether they form a representation of D.



Document D: (a, rose, is, a, rose, is, a, rose)
 All valid 4-grams: a rose is a, rose is a rose, is a rose is, a rose is a, rose is a rose
 Unique 4-gram (Shingles): a rose is a, rose is a rose, is a rose is

Figure 3 Shingles of a String [2]

Fuzzy fingerprinting in [6] implements the same concept by taking a different approach. Instead of removing repeated terms, the method adopts (idfs) to determine which words are meaningful enough to represent the document. A single hash value will be generated using all the feature terms. Matching between two hash values

means that two documents are duplicates of each other.

2.1.2 Term Occurrence

Term occurrence is probably the most intuitive approach because lexical words contain explicit information of the text and they can be analyzed to determine the similarity between two documents. One assumption is that the more terms both documents have in common, the more similar they are. Term occurrence has been applied to a range of studies such as automatic evaluation of summaries, automatic evaluation of machine translation, and common information retrieval problems like clustering and categorization. Due to a common purpose between the aforementioned studies and plagiarism detection, i.e. determining similarities between documents, application of term occurrence in plagiarism detection seems promising. Strictly speaking, fingerprinting in word granularity may be categorized under term occurrence; however, since fingerprinting has been discussed earlier under different category, we will only discuss other term occurrence methods.

CHECK [28], which incorporates a well-known IR model - vector space model (VSM), is a plagiarism detection method that first parses a document into a tree structure before comparing two documents. The root node contains the overall information of a document while the internal and leaf nodes contain information of subsections and paragraphs respectively. The authors called the tree structure the document tree (Figure 4) and the information within as *structural characteristics* (SC) of the document. CHECK operates on one assumption that if a pair of documents does not share similar topics, they are not suspects of plagiarism. Hence, before any comparison of specific information between two documents is carried out, CHECK will compare the root nodes of the two documents first. Each root node contains the keyword set for the particular document. The keyword set is in the form of a weighted

vector. Thus, the overall similarity between two documents is equivalent to the value of the cosine of the angle between the two vectors.

If the cosine measure exceeds a certain threshold, two documents are thought to be similar in content and child nodes will be compared. Like root nodes, child nodes are expressed in weighted vectors as well and they specifically represent subsections of the documents. The process stops when cosine measure falls below the threshold or when leaf nodes are reached.

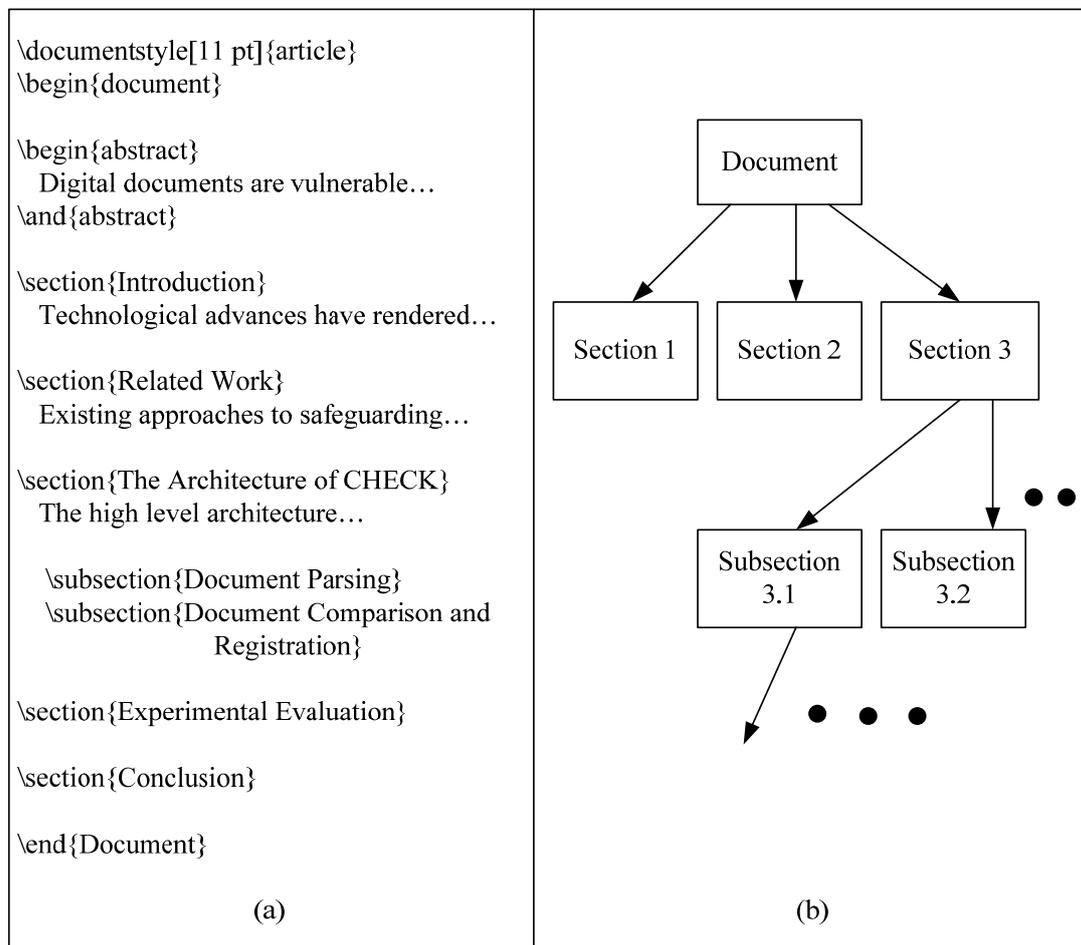


Figure 4 Document Tree [28]

Next, *Smart Version 13* is the information retrieval system in [4] and it adopts VSM to produce representatives of documents. Although [4] is not particularly

designed for plagiarism detection, one of its goals is near-duplicate detection, which can be applied to find instances of plagiarism.

Zaslavsky et al. [37] utilized suffix trees, each of which contains all the suffixes of a string and therefore all the substrings as well. By including all the substrings, a suffix tree is “a data structure ... that allows for a particularly fast implementation of many important string operations” [34]. When applied at document level, the suffix tree method, together with the matching statistics algorithm, is able to find overlapping chunks between two documents [37]. One disadvantage is that such a data structure is more memory consuming than just saving the document. In [17], instead of using suffix trees, the idea of suffix arrays is adopted to reduce the memory problem found in suffix trees.

2.1.3 Style Analysis

Style analysis is the most special approach to plagiarism, because unlike other methods, it requires no reference corpus and it focuses more on implicit information than explicit information of the texts. The basic principle behind style analysis is that every author has his/her own writing style, may it be the difference in text length or choices of words. Other measures also include richness in vocabulary and the number of closed class words and open class words used. Analysis of those measures enables the researchers to turn the abstract idea of writing style into realistic numbers [24]. The principle is in accordance with stylometry. If the style in a document is not consistent throughout the entire document, plagiarism may have occurred. The hypothesis is based on two assumptions that each person’s writing style should remain consistent throughout the text, and that the characteristics of each style is hard to manipulate or imitate, making the plagiarized portion of work to stand out in the text implicitly [9]. Although style analysis may not need a reference corpus, it needs to be

trained to learn about rules of writing. Hence, various artificial neural networks (ANNs) and genetic algorithms (GAs) have been applied to analyze style and authorship [9]. The trained ANNs or GAs will be able to recognize the style of a particular author and therefore articles written by the author.

2.1.4 Comparison and Contrast

The popularity of fingerprinting is probably due to its efficiency in speed and data storage, making it feasible to work on a large corpus. Although fingerprinting has been proved to perform well for verbatim copy of large scale and subset overlapping, it is also known for its vulnerability to modified text. With some minor changes, entirely different fingerprints are generated even if two sets of data remain highly similar. For example, the last two strings in Figure 2 are represented by two different fingerprints even though the only difference between them is just a verb. As a result, true positive may be judged as false negative. Even fuzzy fingerprinting and shingling, which do not require full matching between two sentence, are affected by substitution, addition, and deletion of words because idfs and shingles will change accordingly.

As VSM includes term frequency and inverse document frequency, it shares the same vulnerability mentioned above. Another weakness of VSM is inevitable due to the nature of this model, which works in a bag-of-words manner; as a result, the vector represents only the global information of a document. And it is only capable of global plagiarism detection and is not able to point out the exact location of an instance of plagiarism. Although the document tree in CHECK can focus the location of probable plagiarism, the smallest unit of detection is still in paragraphs.

With only the information of all substrings, the suffix tree approach is vulnerable to rewording, especially substitution of words when doing substring matching. Although a suffix tree is capable of fast string operations, [37] indicates that the tree

stores only the substrings and does not record the positions of substrings in the document. Such a data structure cannot locate exact substring match in the document.

Style analysis does not require a reference corpus when detecting plagiarism and does not seem to be affected by text alteration based on its theory. However, writing style of a person can change with time and age; thus making the analysis of style too inconsistent and unreliable. Moreover, even if no reference corpus is needed, a training corpus is still required.

2.2 ROUGE

One straightforward way to determine if a sentence in a candidate document is plagiarized from a sentence in a reference document is to compare the candidate sentence with all the sentences in the reference document. Based on the intuition that a pair of plagiarized sentence and plagiarizing sentence is identical in content, we can find out sentence pairs that may be subject to plagiarism by calculating the similarity for each pair.

There are methods for calculating the similarity between two sentences; one method is n-gram co-occurrence statistics in BLEU [25] for machine translation evaluation. Followed by the success of BLEU, the same method was included as part of ROUGE [18], which is implemented in the proposed system with some minor modifications and extended applications. Implementation consists of major ROUGE components: ROUGE-N, longest common subsequence, and skip-bigram. Applications of each individual method in this research will be discussed more deeply in Chapter 3.

Lin [18] tested the performance of each method, including variations of LCS and skip-bigram, by comparing one or more reference summary(ies) with a candidate

summary, then came up with a score representing the quality of the candidate summary. Moreover, experiments were carried out under different settings such as stopword removal and stemming. During experimental evaluation, the scores given by the methods were compared with those of human judgment, which served as the answers. Various correlation measures were used to assess the performance of the methods. Higher correlation between a method and its corresponding human scores suggests that the method can evaluate summaries in a way close to human judgment, proving the effectiveness of the method.

2.3 WordNet

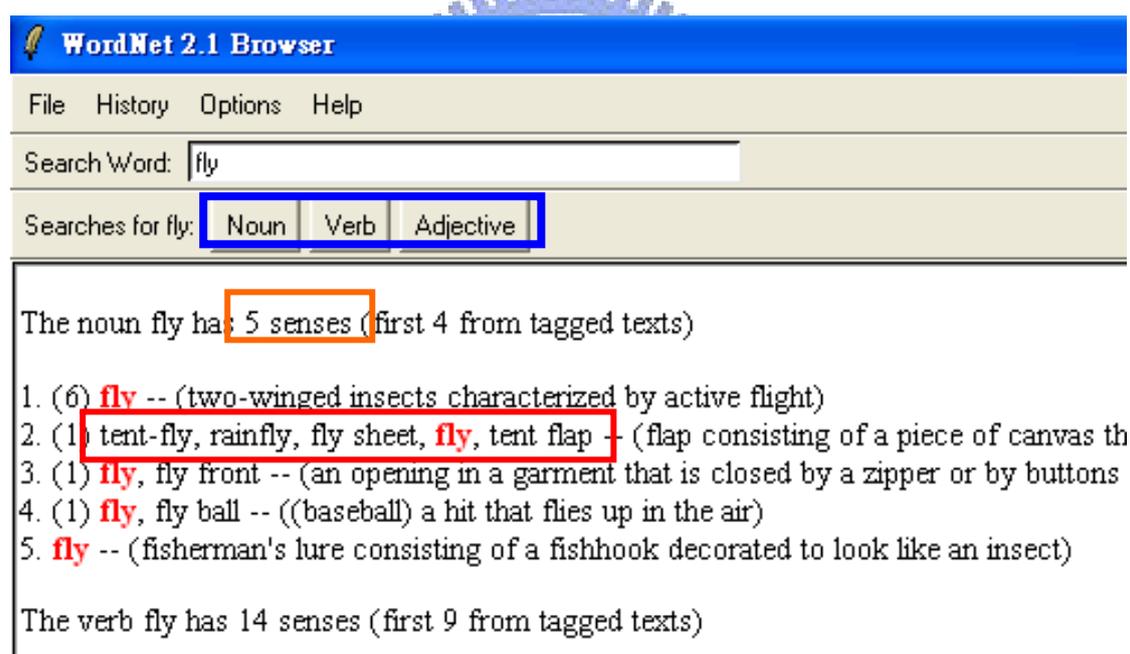
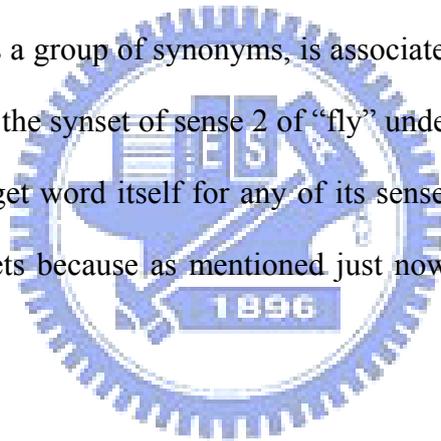


Figure 5 Search Result of “fly” in WordNet

WordNet [35] is a dictionary-like database developed by Cognitive Science Laboratory of Princeton University. Some of the previous plagiarism detection research, such as Iyer and Singh [12] and Kang et al. [14], had adopted WordNet, both used the database for finding synonyms of words to detect plagiarism through

substitution of words. However, Kang et al. did not illustrate how WordNet was used to find synonyms while Iyer and Singh compared synsets to determine if two words were synonyms of each other. Figure 5 shows the interface of local version of WordNet 2.1.

Basically, every word in WordNet can be assigned to one or more part-of-speech (POS) categories. There are four POS categories: noun, verb, adjective and adverb. For example, the word “fly” has three POS in WordNet as the blue box in Figure 5 indicates three options – Noun, Verb, and Adjective. Each word has different numbers of senses under each category. Senses can be understood as different valid meanings given to a word. The orange box in Figure 5 shows that “fly” has five senses under Noun. A synset, which is a group of synonyms, is associated with each sense. The red box in Figure 5 encloses the synset of sense 2 of “fly” under Noun. However, a synset may contain just the target word itself for any of its senses. Reasonably, a word can belong to multiple synsets because as mentioned just now, a word or polysemy can have multiple meanings.



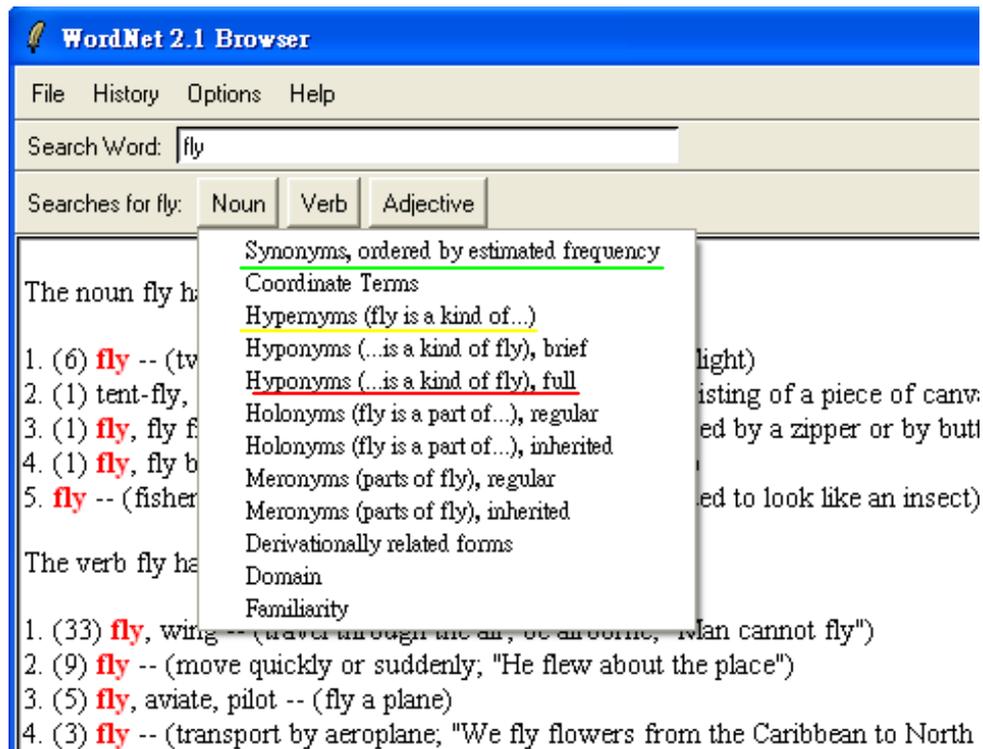


Figure 6 Lexical and Semantic Functions Available for “fly”

Words in WordNet are linked by two major relations – lexical and conceptual semantics. Besides synonyms, other lexical relations including but not limited to holonyms and meronymy are also found in WordNet. As shown in Figure 6, by clicking any of the three POS icons, one will see a drop list which contains available options. WordNet can hierarchically show the hypernym relationship between words. Figure 7 shows the hierarchy of hypernyms of “fly” under Noun.

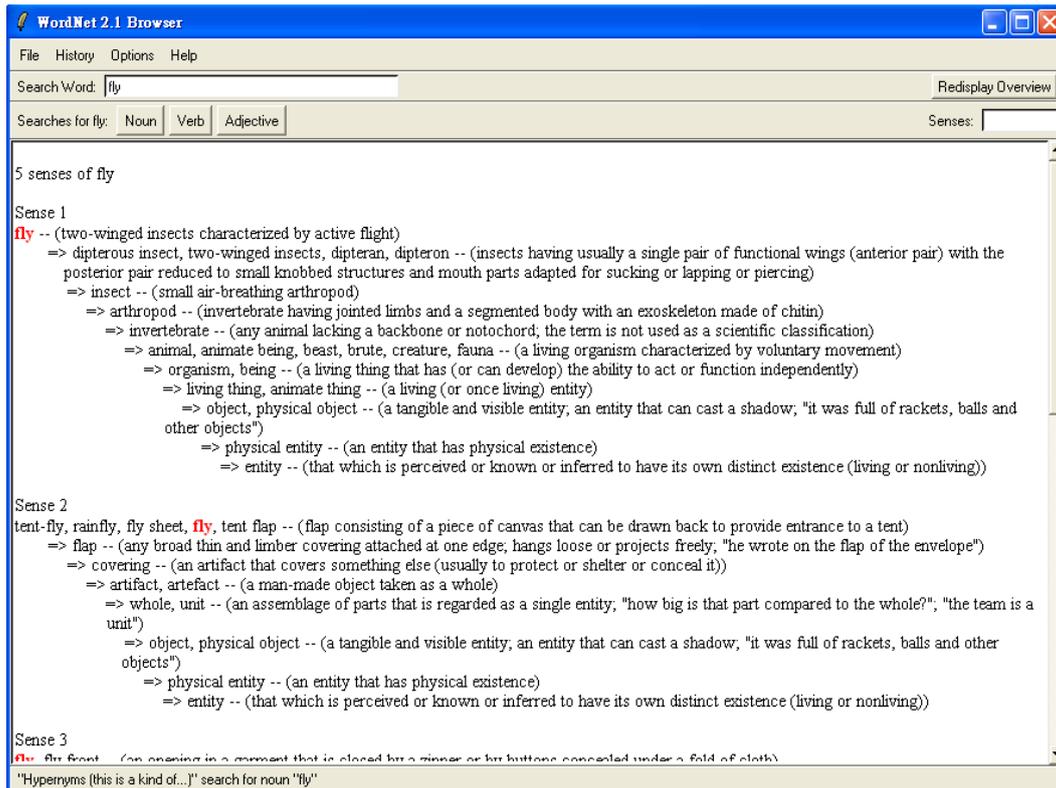


Figure 7 Hypernym Hierarchy of “fly” in WordNet

As people can substitute words in the original text with synonyms when they plagiarize, n-grams that only consider exact match is unable to detect the substitutions, resulting false negative between the two sentences. Therefore, WordNet may be helpful when analyzing a sentence pair for this type of plagiarism, because it can be applied to find implicit relationship between two words. Weighting or score can be given according to the closeness of two words either lexically or conceptually.

There are many WordNet-related projects available. A Java API – Java WordNet Library (JWNL) [36], which has a dictionary database that contains all the words and their relationships in WordNet, was integrated into the program. Through JWNL, users can retrieve the same information of a word as in the local version of WordNet via the right Java application.

3. Methodology

Having discussed about existing plagiarism detection methods, the methods proposed in this research will be discussed next. The purpose of this research is to provide a framework of a plagiarism detection tool.

The methods are based on the foundation of n-gram co-occurrence statistics at sentence level. N-gram co-occurrence statistics can detect verbatim copy as good as fingerprinting. However, by including longest common subsequence and skip-bigram, we hope to overcome problems caused by addition and deletion of original text. In situations where original words are substituted with synonyms, WordNet has been implemented to overcome this problem. Sentence level matching means that we can locate the position of plagiarism instance in the document by recording the sentence numbers for all the comparing pairs and their similarity scores. As mentioned earlier, fingerprinting can handle large amount of information, and several studies have applied their methods on relatively large corpora. However, this research focuses on the accuracy of plagiarism detection within a document, instead of trying to detect plagiarism in a corpus.

The documents are processed and saved in string tokens, which are compatible with WordNet. But by using string tokens, it means that the efficiency is most likely poorer than integer-based fingerprinting. Another issue is the comparing scheme, whose complexity is $O(n^2)$. If both documents' contents are lengthy, memory consumption problem may rise.

3.1 System Architecture for Plagiarism Detection

Two documents will be uploaded, and they will be preprocessed and then analyzed according to the options chosen by the user. At the end, the output will be

pair-wise scores that indicate the probability of plagiarism. Figure 8 shows the architecture and components of our system, and the following sections will explicate the components in detail.

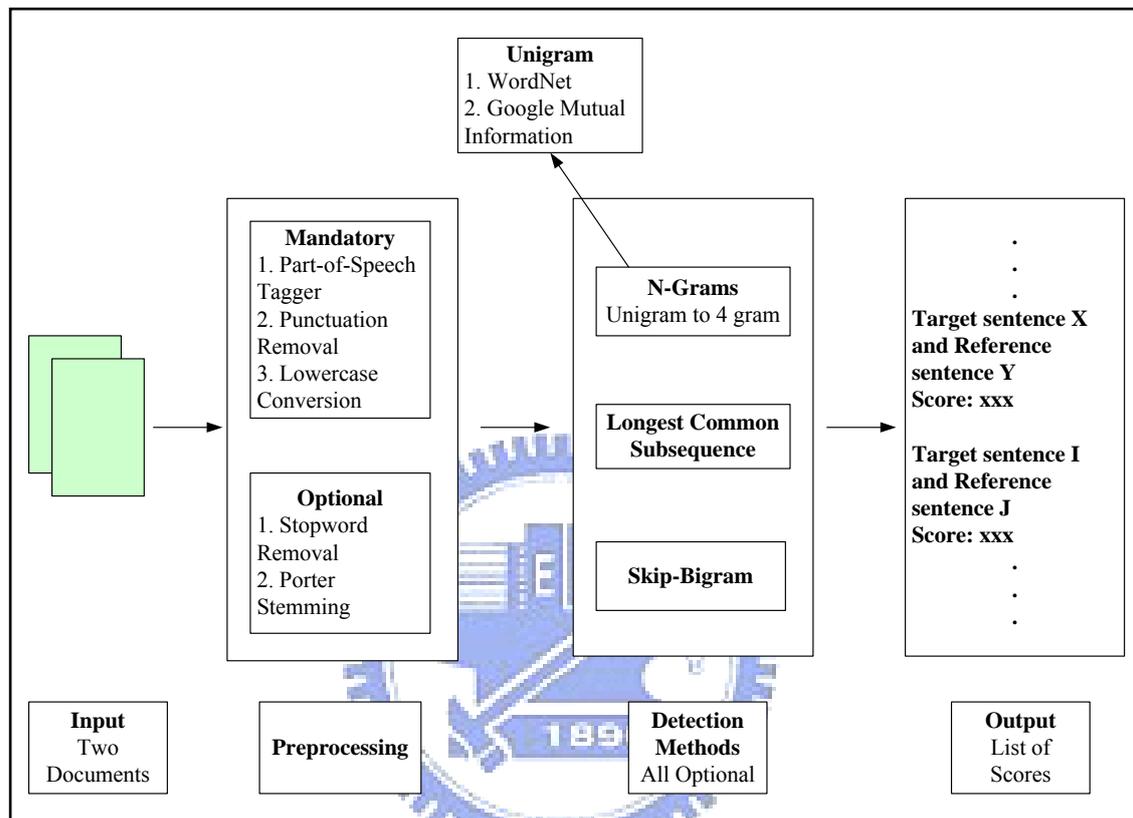


Figure 8 System Architecture

3.2 Preprocessing

3.2.1 Tokenization and Sentences

Since our basic unit of detection is in sentence, documents will be processed into tokens and sentences by using LingPipe's [19] *sentence detection* API, which is just one of the many language processing Java APIs offered by LingPipe. The API will store the text with two arrays – *tokens* and *whites*. The *tokens* array stores the tokens and punctuations while the *whites* array stores white spaces. Using the MEDLINE sentence model, the API can recognize sentence boundary indicators and present the

original text in sentences by knowing which tokens are at the end of the sentences. With this information, the API can reassemble each sentence correctly. MEDLINE, according to LingPipe, “is a collection of 13 million plus citations into the bio-medical literature maintained by the United States National Library of Medicine (NLM), and is distributed in XML format.” [20]. In this research, only the *tokens* array was used. Figure 9 shows the layout of a text in the text editor while Figure 10 shows the display of the same text after it has been processed by the API. The same problem which had been brought up in other research was encountered when the API looks for sentence boundary indicators to determine the end of a sentence: periods that are not used as an end-of-sentence indicator will be mistaken. Acronyms cannot be processed properly because individual letters that are supposed to be joined by period(s) will be separated as independent characters. It is hard to distinguish acronyms because we only use the *tokens* array; there is no additional information that can be used to disambiguate a period. The layout of research papers also causes problems because the section title usually will be included in the first sentence of the paragraph that follows immediately after the title, which has no indicator to separate it from the sentence.

Developing complex skills in the classroom involves the key ingredients identified in teaching pigeons to play ping-pong and to bowl. The key ingredients are: (1) inducing a response, (2) reinforcing subtle improvements or refinements in the behavior, (3) providing for the transfer of stimulus control by gradually withdrawing the prompts or cues, and (4) scheduling reinforcements so that the ratio of reinforcements in responses gradually increases and natural reinforcers can maintain their behavior.

Figure 9 Unprocessed Text

Sentence 1: developing complex skills in the classroom involves the key ingredients identified in teaching pigeons to play ping-pong and to bowl

Sentence 2: the key ingredients are 1 inducing a response 2 reinforcing subtle improvements or refinements in the behavior 3 providing for the transfer of stimulus control by gradually withdrawing the prompts or cues and 4 scheduling reinforcements so that the ratio of reinforcements in responses gradually increases and natural reinforcers can maintain their behavior

Figure 10 Text Divided into Sentences

3.2.2 Part-of-Speech (POS) Tagger

After the *tokens* array is obtained, the tokens are processed by LingPipe's part-of-speech tagging API. Using the Hidden Markov Model trained with the Brown Corpus, each token in the *tokens* array will be tagged with a POS, which is saved in another array *tags*. Brown Corpus is a statistical analysis of American English consisting of 1,014,312 words. The corpus used text materials printed in 1961 and was done by W. N. Francis and H. Kucera at Brown University [3][31]. POS tagging is a crucial step which enables us to look up the appropriate synsets and meanings of a word in the WordNet more accurately. Figure 11 shows the tokens and their respective

POS tag separated by an underscore.

Sentence 1: developing_vbg complex_jj skills_nns in_in the_at classroom_nn involves_vbz the_at key_jjs ingredients_nns identified_vbn in_in teaching_vbg pigeons_nns to_to play_vb ping-pong_nn and_cc to_in bowl_nn

Sentence 2: the_at key_jjs ingredients_nns are_ber 1_cd inducing_vbg a_at response_nn 2_cd reinforcing_vbg subtle_jj improvements_nns or_cc refinements_nns i_nil n_nil the_at behavior_nn 3_cd providing_vbg for_in the_at transfer_nn of_in stimulus_nn control_nn by_in gradually_rb withdrawing_vbg the_at prompts_nns or_cc cues_nns and_cc 4_cd scheduling_nn reinforcements_nns so_rb that_cs the_at ratio_nn of_in reinforcements_nns in_in responses_nns gradually_rb increases_vbz and_cc natural_jj reinforcers_nns can_md maintain_vb their_pp\$ behavior_nn

Figure 11 Words with Corresponding POS Tag

3.2.3 Punctuation Removal and Lowercasing

When a text is stored as tokens and each of the tokens has been POS tagged, common punctuations and their respective POS tag are removed from the *tokens* array. At the same time, each token is converted into lower case. Hereafter, the three terms token, word, and unigram are used interchangeably.

3.2.4 Stopwords Removal

Stopwords are words that are often not informative. They may cause false positives during matching because two unrelated sentences may get a higher score than what they really deserve due to matched stopwords such as *this*, *if*, *in* and many others. We use a Java stopwords removal API from Terrier [30], plus a stopwords list stored in plain text. Any word that finds a match in the list will be removed from the *tokens* array. At the same time, the stopwords list can be adjusted anytime. According

to the official website, Terrier stands for Terabyte Retrieval and it is developed by the Computing Science Department of the University of Glasgow. Terrier is an open source information retrieval platform. Figure 12 shows the same text as in Figure 9 with stopwords removed.

Sentence 1: developing complex skills classroom involves key ingredients identified teaching pigeons play ping-pong bowl

Sentence 2: key ingredients 1 inducing response 2 reinforcing subtle improvements refinements behavior 3 providing transfer stimulus control gradually withdrawing prompts cues 4 scheduling reinforcements ratio reinforcements responses gradually increases natural reinforcers maintain behavior

Figure 12 Text with Stopwords Removed

3.2.5 Stemming

“Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form – generally a written word form.” [33]. For example, “*computing*”, “*computer*”, and “*computation*” have a common root – “*comput*”. And words that share the same root/stem are usually semantically close. By stemming the words, higher rate of meaningful matching can be achieved. Such a concept is the opposite of stopword removal, which aims to reduce useless matching. A Java version of Porter stemming is used and it is available in Terrier as well. Figure 13 shows a sample of stemmed text.

Sentence 1: develop complex skill in the classroom involv the kei ingredi identifi in
teach pigeon to plai ping-pong and to bowl

Sentence 2: the kei ingredi ar 1 induc a respons 2 reinforce subtl improv or refin i n
the behavior 3 provid for the transfer of stimulu control by gradual withdraw the
prompt or cue and 4 schedul reinforce so that the ratio of reinforce in respons gradual
increas and natur reinforce can maintain their behavior

Figure 13 Stemmed Text

3.3 Plagiarism Detection Methods

Figure 8 clearly shows that the proposed system includes n-gram analysis up to 4-gram, LCS, skip-bigram, WordNet, and Google mutual information (MI). Initially, all the above methods were based on recall, i.e. the number of matched tokens between two sentences was divided by the length of reference sentence. The perspective is that plagiarism occurs when original work is being copied or modified without proper citation; the amount of material being plagiarized is not really an issue, whether it is a paragraph or just a sentence being copied, instance of plagiarism still holds. Therefore, the higher the recall of tokens/n-grams of a reference sentence, the higher the probability of plagiarism. However, later observation discovered that by just considering the length of reference sentence, problems occur when sentence lengths differ substantially. In case of a long candidate sentence and a short reference sentence, false positive may occur and vice versa. To minimize the problems, in the final version of all the measures, the score is represented by F-measure, a balanced average between the recall of a reference sentence and the precision of a candidate sentence.

3.3.1 ROUGE-N

3.3.1.1 Unigram

Each token in a sentence is a unigram. Before comparing the sentences, every unique unigram and its number of occurrence(s) in the sentence will be recorded for every sentence. Beginning with the first sentence of the reference document, every unique unigram is compared with all unique unigrams in every sentence of the candidate document, followed by the second sentence of the reference document and so on and so forth. Overall, all reference sentences will be compared with all candidate sentences for a total of $M \times N$ times, where M and N are the number of sentences in the reference and candidate documents respectively.

The number of overlapping unigram(s) between two sentences, one from the reference document and the other from the candidate document, will be counted. The overlapping total, numerator of Equations (1) and (2), is divided by the length of the reference sentence and length of the candidate sentence separately in order to calculate recall and precision. We take the smaller number of occurrence of the overlapping unigrams in the two sentences as the numerator. This is to avoid false positive in certain cases, in which a particular unigram are found in both the candidate and reference sentences but appears multiple times in the candidate sentence. Such modification is called *clipping* [25]. Figure 14 is an example from BLEU [25], in which the word, *the*, appears in both the reference and candidate sentences two times and seven times respectively, if according to Equation (1) without the *clipping* mechanism, the precision score contributed by this word will be $7/7$, which is clearly exaggerated. However, if the score is clipped it becomes $2/2$, which is more reasonable.

Candidate Sentence: <i>the the the the the the the</i>
Reference Sentence: <i>the cat is on the mat</i>
Standard Precision: 7/7
Clipped Precision: 2/7

Figure 14 Example of Clipped Precision

During sentence matching, we do not consider any reference sentence that has less than four tokens, because a short sentence often leads to high score and false positive.

$$R - N(S_u^R, S_v^C) = \frac{\sum_{\substack{\text{n-gram} \in S_u^R \text{ and } S_v^C \\ 1 \leq u \leq y \\ 1 \leq v \leq z}} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{\text{n-gram} \in S_u^R} \text{Count}(\text{n-gram})} \quad (1)$$

S_u^R and S_v^C represent the sentence pair, n-gram is the overlapping gram of length n

$$P - N(S_u^R, S_v^C) = \frac{\sum_{\substack{\text{n-gram} \in S_u^R \text{ and } S_v^C \\ 1 \leq u \leq y \\ 1 \leq v \leq z}} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{\text{n-gram} \in S_v^C} \text{Count}(\text{n-gram})} \quad (2)$$

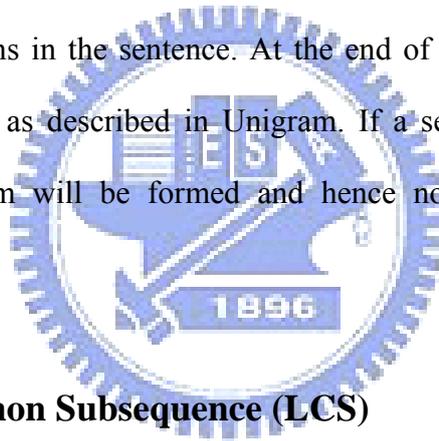
N-gram (including unigram) score is expressed as Equation (3) below:

$$F - N(S_u^R, S_v^C) = \frac{2 * R - N * P - N}{R - N + P - N} \quad (3)$$

3.3.1.2 N-grams

The comparison procedure for n-grams (from two to four) is the same as unigram. The only two differences are the definition of a unique n-gram and an extra step when processing the documents.

As mentioned in unigram, each token is a unique unigram; however, a unique n-gram is made up of more than one token, i.e. a unique bigram consists of two consecutive tokens in the sentence, three tokens for a trigram, and so on and so forth. Therefore, the number of n-grams per sentence has to be determined first. This is done by recording all the n-grams by scanning the sentence with a window size n , and advancing the window by one token along the sentence for a total of $(s - n + 1)$ times. s is the number of tokens in the sentence. At the end of this step, we can go on to compare the documents as described in Unigram. If a sentence is shorter than the window size, no n-gram will be formed and hence no score for that particular sentence pair.



3.3.2 Longest Common Subsequence (LCS)

LCS is the longest in-sequence string of matched tokens between two sentences. In unigram matching, the position of matched token is not a constraint. As long as a unigram co-occurs in both sentences, it will contribute to the similarity between two sentences. Although LCS is also based on matching unigrams, it only considers matched tokens that form the longest in-sequence subsequence of the reference sentence. In other words, even if a unique unigram is in both the reference and candidate sentences, but if it is out of order with other matched tokens, it is not included in the LCS and will not contribute to the LCS score. And if there is more than one common subsequence, LCS will only reflect the longest subsequence among them. Another characteristic of LCS is that unlike n-grams (excluding unigram), LCS

allows skip of matched tokens, which need not be strictly consecutive. Figure 15, which is taken from ROUGE [18], illustrates how LCS is derived.

Candidate sentence 1: Police kill the gunman

Candidate sentence 2: The gunman kill police

Reference sentence: Police killed the gunman

The LCS between Reference sentence and Candidate sentence 1 is *police the gunman* while the LCS between Reference sentence and Candidate sentence 2 is *the gunman, excluding police*. The first pair of sentences shows the skipping nature of LCS and the second pair of sentences shows the in-sequence rule that bounds LCS.

Figure 15 Examples of LCS

If LCS is less than four unigrams long, we drop the sentence due to the same reason mentioned in Unigram.

LCS can be expressed as follows:

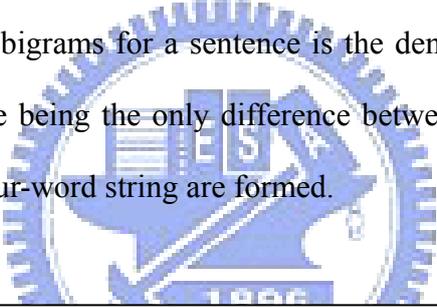
$$R-LCS(S_v^C, S_u^R) = \frac{LCS(S_v^C, S_u^R)}{\sum_{\text{unigram} \in S_u^R} \text{Count}(\text{unigram})} \quad (4)$$

$$P-LCS(S_v^C, S_u^R) = \frac{LCS(S_v^C, S_u^R)}{\sum_{\text{unigram} \in S_v^C} \text{Count}(\text{unigram})} \quad (5)$$

$$F-LCS(S_v^C, S_u^R) = \frac{2 * (R-LCS) * (P-LCS)}{(R-LCS) + (P-LCS)} \quad (6)$$

3.3.3 Skip-Bigram

Skip-bigram is an evolved version of bigram. The difference is the formation of bigrams. For skip-bigram, bigrams are formed not only by consecutive tokens, but also by other in-sequence tokens within the window. Skip distance, d , has to be set before counting skip-bigrams in the sentence. Skip distance is the maximum number of tokens in between any two combining tokens. When skip distance is determined, we can start finding all the skip-bigrams within a sentence. Let w_1, w_2, \dots, w_n be a sentence. Starting with w_1 , it will form skip-bigrams with of the following $d+1$ words, followed by w_2 , which forms skip-bigrams with another in-sequence $d+1$ words. The process stops when w_{n-1} forms the last skip-bigram with w_n . Therefore, the total number of skip-bigrams for a sentence is the denominator of Equations (7) and (8) with the sentence being the only difference between them. Figure 16 shows how skip-bigrams of a four-word string are formed.



For a given sequence: *Andy eats an apple*

When $d=2$, skip-bigrams generated will be as follows:
Start with the first word *andy*, it can form a bigram with the furthest token, *apple*, and followed by *eats* and *an* respectively. When *andy* has formed bigrams with all possible tokens, *eats* will form bigrams with *an* and *apple*. Finally, the last skip-bigram is *an apple*.

There are a total of six skip-bigrams by the sequence above when $d=2$. They are as shown:

Figure 16 Example of Skip-bigrams Formation

After finding the skip-bigrams for the sentence pair, we can compare the skip-bigrams using the same steps for n-grams.

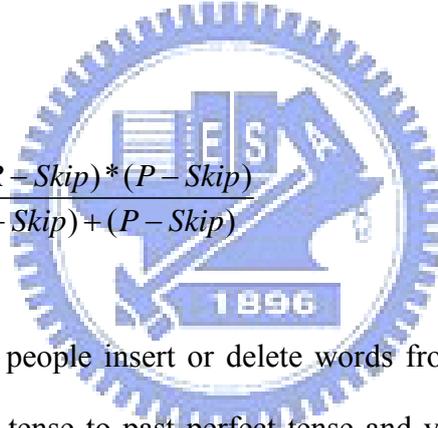
$$R - Skip(S_u^R, S_v^C) = \frac{SKIP2_{clip}(S_u^R, S_v^C)}{(p-d-1)(d+1) + \sum_{r=0}^{d-1} d-r} \quad (7)$$

Sentence S_R of length p

$$P - Skip(S_u^R, S_v^C) = \frac{SKIP2_{clip}(S_u^R, S_v^C)}{(q-d-1)(d+1) + \sum_{r=0}^{d-1} d-r} \quad (8)$$

Sentence S_C of length q

$$F - Skip(S_u^R, S_v^C) = \frac{2 * (R - Skip) * (P - Skip)}{(R - Skip) + (P - Skip)} \quad (9)$$



In situations where people insert or delete words from an original sentence, or change tenses from past tense to past perfect tense and vice versa, pure bigram has minimal use; this is because the bigrams will not be the same between reference and candidate sentences. As skip-bigram allows gaps, it has higher chance of producing the correct bigrams to match with the ones in the reference sentence.

3.3.4 WordNet

WordNet is integrated into unigram to go beyond matching of exact tokens. In unigram, no score is given if two comparing unigrams are not exactly the same. However, extra steps are taken after integrating WordNet. Following the same procedures in Section 3.3.1.1, a unique unigram from the reference sentence will be

matched against all the unique unigrams in the candidate sentence. However, if there is not an exact match, the relationship between the two words in WordNet will be looked up.

First, connection to JWNL's dictionary will be established in order to access the WordNet database. Second, the WordNet POS of each comparing word has to be determined. As mentioned in Section 2.3, WordNet uses four general POS tags but the Brown POS tagger has a much longer list of tags; therefore, POS tags have to be classified. For example, if a word has been tagged by the Brown POS tagger with any of the following tags: "nn", "nns", "np", "nps", or "nr", it will be assigned with a WordNet POS tag - *NOUN*. Similar classification is applied to the remaining three WordNet POS tags. The classification tries to include tags for meaningful terms (open word class) while exclude tags that are for stopwords (closed word class) and punctuations. Third, if both words with their specific POS can be found in the WordNet, their lexical and semantic relationship can be determined. If not, no further action is taken and this pair of words is considered irrelevant.

Two different measures are taken to determine the relationship between two words. They will be discussed as follows.

3.3.4.1 Synonyms-based

Jaccard's coefficient, Equation (10), is used to measure the similarity between two synsets. The numerator is the intersection of synonyms between synsets k and l while the denominator is the distinct union of synonyms of synsets k and l . After synsets for each word are obtained, synonyms in each synset have to be separated as an individual word. Next, the number of overlapping synonyms between two synsets of reference word and candidate word is counted, and divided by the sum of distinct synonyms in the two synsets. Similar to the sentence-based comparison between two

documents, each of the synsets of a word will be compared with all the synsets of the other word. The highest score – Equation (11) will be recorded and this is the similarity score between two unigrams. The same steps are repeated until a reference unigram has compared with all the unique unigrams in the candidate sentence. The highest similarity score among the calculated scores will be chosen - Equation (12). The maximum value adds to the total plagiarism score between the two sentences. Figure 17 is an example of how synsets of two words are matched:

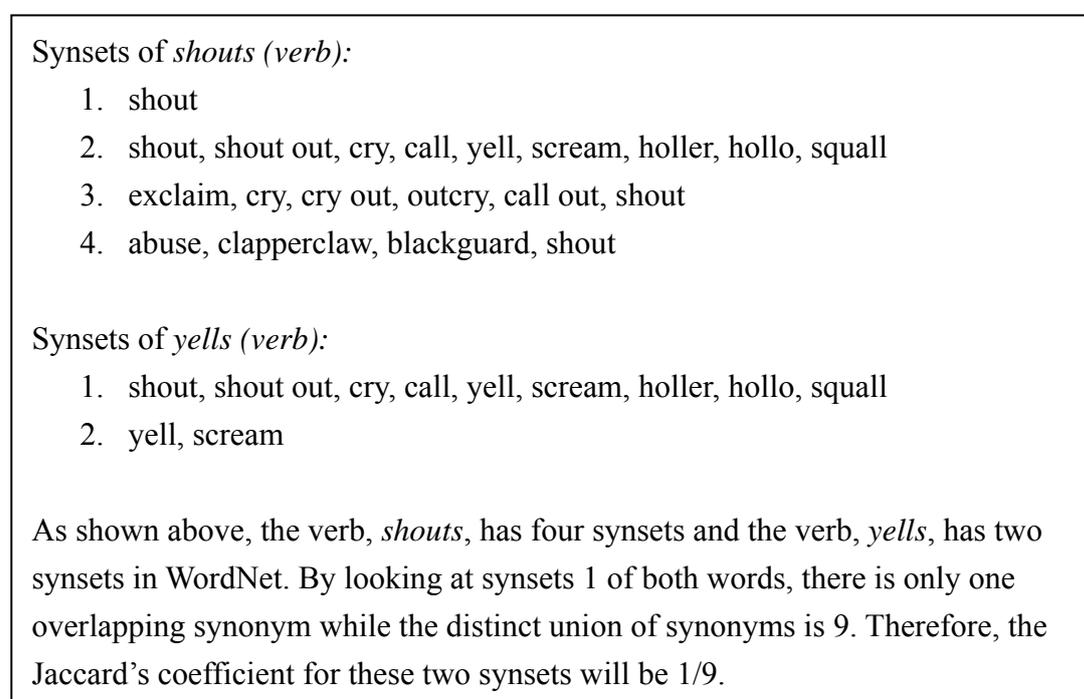


Figure 17 Example of Jaccard's Coefficient between Two Synsets

Jaccard's coefficient:

$$\frac{|k \cap l|}{|k \cup l|} = sim(s_k^i, s_l^j) \quad (10)$$

s_k^i is synset k of word i while s_l^j is synset l of word j

Similarity between two words:

$$sim(w_i^R, w_j^C) = Arg \max_{\substack{1 \leq k \leq m \\ 1 \leq l \leq n}} sim(s_k^i, s_l^j) \quad (11)$$

w_i is word i with m synsets and w_j is word j with n synsets

Similarity score between word i and a given sentence:

$$sim(w_i^R, S_C) = Arg \max_{\substack{1 \leq i \leq p \\ 1 \leq j \leq q}} sim(w_i^R, w_j^C) \quad (12)$$

Word $i \in$ sentence S_R of length p and word $j \in$ sentence S_C of length q

Therefore, the plagiarism score for any pair of reference sentence and candidate sentence is as follows:

$$R - Synset(S_u^R, S_v^C) = \frac{\left(\sum_{i=1}^p sim_{clip}(w_i^R, S_v^C) \right)}{\sum_{\text{unigram} \in S_u^R} \text{Count}(\text{unigram})} \quad (13)$$

$$P - Synset(S_u^R, S_v^C) = \frac{\left(\sum_{i=1}^p sim_{clip}(w_i^R, S_v^C) \right)}{\sum_{\text{unigram} \in S_v^C} \text{Count}(\text{unigram})} \quad (14)$$

$$F - \text{Synset}(S_u^R, S_v^C) = \frac{2 * (R - \text{Synset}) * (P - \text{Synset})}{(R - \text{Synset}) + (P - \text{Synset})} \quad (15)$$

$1 \leq u \leq y$
 $1 \leq v \leq z$

$w_i \in S_R \in \text{Document R of length } u \text{ and } S_C \in \text{Document C of length } v$

3.3.4.2 Relationship-based

Relationship refers to hypernym and hyponym relationships in WordNet. The first few steps for finding hypernym and hyponym relationships are exactly the same as the first few steps for comparing synonyms up to the step where we obtain the senses for both words. After that, hypernym and hyponym relationships between two senses can be found. The term *senses* is used here instead of *synsets* because synonyms are not the focus but how each sense/meaning of a word is semantically related to other senses of the other word. Again, all senses of the reference unigram have to be compared with all the other senses of the candidate unigram. The relationship is expressed hierarchically in terms of *depth*. If two words are actually in the same synset, the depth is zero. This research only considers relationship depth within three levels in the hierarchy. Figure 18 is an example of how hypernym/hyponym relationships between two words are determined.

Both hypernym and hyponym relationships are obtained in identical manner. As there is not any reference about how to set the weight, w_t in Equation (18), initial assignment of weight for each depth will be as follows: 1.0 if the returned depth is zero, 0.9 if the depth is one and so forth with the maximum depth being three. Then we choose the bigger value between the hypernym score and hyponym score and use it to represent the relationship between the word pair like Equation (16).

Synsets of *shouts* (*verb*):

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

Synsets of *yells* (*verb*):

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

Although there are a total of eight possible combinations of synsets between the two verbs, *shouts* and *yells*, these two words are linked by two pairs of synsets. Synset 1 of *shouts* and synset 2 of *yells*; synset 2 of *shouts* and synset 1 of *yells*. The first pair only has hypernym relationship of depth 1, while the second pair has both hypernym and hyponym relationships of depth 0. Since depth 0 is the closest relationship possible, the relationship between *shouts* and *yells* is represented by the second pair.

Figure 18 Example of Hypernym/Hyponym Relationship between Two Words

$$RS(w_i^R, w_j^C) = \underset{\substack{1 \leq k \leq m \\ 1 \leq l \leq n}}{\text{Arg max}} (\text{Arg max}_{\text{hypernym}}(s_k^i, s_l^j), \text{Arg max}_{\text{hyponym}}(s_k^i, s_l^j)) \quad (16)$$

w_i is word i with m synsets and w_j is word j with n synsets

$$RS(w_i^R, S_C) = \underset{\substack{1 \leq i \leq p \\ 1 \leq j \leq q}}{\text{Arg max}} RS(w_i^R, w_j^C) \quad (17)$$

Word $i \in$ sentence S_R of length p and word $j \in$ sentence S_C of length q

$$wt = \begin{cases} \text{if depth}=0, wt_1 \\ \text{if depth}=1, wt_2 \\ \text{if depth}=2, wt_3 \\ \text{if depth}=3, wt_4 \end{cases}$$

$$R - RS(S_u^R, S_v^C) = \frac{\left(\sum_{i=1}^p sim_{clip}(w_i^R, S_v^C) \square wt \right)}{\sum_{\text{unigram} \in S_u^R} \text{Count}(\text{unigram})} \quad (18)$$

$$P - RS(S_u^R, S_v^C) = \frac{\left(\sum_{i=1}^p sim_{clip}(w_i^R, S_v^C) \square wt \right)}{\sum_{\text{unigram} \in S_v^C} \text{Count}(\text{unigram})} \quad (19)$$

$$F - RS(S_u^R, S_v^C) = \frac{2 * (R - RS) * (P - RS)}{(R - RS) + (P - RS)} \quad (20)$$

3.3.5 Google Mutual Information (MI)

The tremendous number of Web pages on the Internet constitutes a giant database of tokens. A method for calculating the mutual information (relatedness) between two words [13][26] is applied to plagiarism detection between two sentences. Google's SOAP Search API is responsible for sending the queries to Google and retrieving information about the queries. For each pair of words, three queries are sent: one query per word plus an additional query with both words. We record the number of retrieved pages and use the numbers to calculate MI. The expressions are as follows:

$$MI(w_i, w_j) = \frac{\log(w_i) + \log(w_j) - 2 * \log(w_i, w_j)}{Max MI} \quad (21)$$

$$sim(w_i^R, S_v^C) = Arg \max_{\substack{1 \leq i \leq p \\ 1 \leq j \leq q}} MI(w_i^R, w_j^C) \quad (22)$$

$$R - MI(S_u^R, S_v^C) = \frac{\sum_{i=1}^p sim_{clip}(w_i^R, S_v^C)}{\sum_{unigram \in S_u^R} Count(unigram)} \quad (23)$$

$$P - MI(S_u^R, S_v^C) = \frac{\sum_{i=1}^p sim_{clip}(w_i^R, S_v^C)}{\sum_{unigram \in S_v^C} Count(unigram)} \quad (24)$$

$$F - MI(S_u^R, S_v^C) = \frac{2 * (R - MI) * (P - MI)}{(R - MI) + (P - MI)} \quad (25)$$

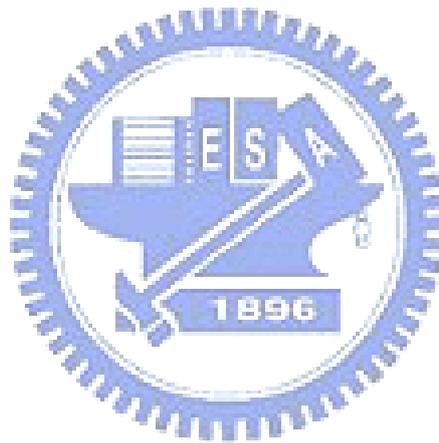
Note: Google SOAP Search API only allows 1000 queries per key per day.

3.3.6 Caching

We realize that the processing speed is impractically slow even for short paragraphs after implementing WordNet. Therefore, we add a caching mechanism into WordNet and Google MI in hope of improving the speed. The entire process is described as follows:

1. If two unigrams do not match, go to the established MySQL database and search for the score of the specific word pair;

2. If no match of the word pair can be found in the database, calculate the score for the word pair and update the score into the database for future use;
3. Repeat the step 1 and 2 if the condition applies.



4. Experiments and Evaluation

4.1 Data Sets

In the field of plagiarism detection, there is not a standard and valid plagiarism corpus that is publicly available yet. A number of works used news corpus such as the Reuters News corpus for their evaluation, while a small number of works used research articles corpora that are managed by the university and therefore only accessible by the university members. The remaining choice is to make one's own plagiarism corpus, which usually is relatively small due to limited resources. This research adopted the last approach instead of using a news corpus because even though news content is often reused, modification of this nature may not be able to represent plagiarism.

There are two different manually generated data sets for evaluating the proposed methods. One of the sets contains 978 pairs of sentences while the other set contains 100 pairs of sentences. These two sets will be referred to as the *abstract* set and the *paraphrased* set respectively hereafter. The *abstract* set was used primarily to determine the ideal settings for the methods. It was based on the observation that abstracts of some papers are actually formed by sentences taken from the main text. Such characteristic may be utilized to simulate the plagiarism scenario by treating the abstract as the candidate of plagiarism and the main text as the source being plagiarized. First, a collection of research papers were retrieved from research databases like Elsevier and EBSCOhost using the query “plagiarism”. Second, the abstract and the main text of each paper were separated and saved in two different plain text files. Some manual efforts were required to remove undesirable texts that appeared in certain parts of the papers as they interfered with the main body of the text and might affect the outcome of the experiment. Third, each abstract sentence

was compared with the main text using six different methods namely n-grams (unigram to 4-gram), LCS and skip-bigram. Top five matching in each method were recorded regardless of the scores. In other words, each abstract sentence produced 30 pairs of sentences with the main text; however, if there were repeated pairs within the 30 pairs, only one of them would be included in the *abstract* corpus. There were 1000 unique pairs of sentences out of 19 research papers in the end. Fourth, four people who had been educated about plagiarism and understood the concept of plagiarism were asked to annotate the *abstract* corpus. The sentence pairs were randomly divided into four groups and each person was given 500 pairs so that each pair would be annotated by two different persons. After the annotation was completed, kappa statistics [15] was applied to ensure the reliability and validity of the annotation by measuring the agreement between the annotators. In the end, the groups had a minimum kappa score of 0.696 and a maximum score of 0.863, with two groups falling into the range of *substantial* while the other two groups falling into the range of *almost perfect*. Table 1 below serves as a general reference for kappa scores:

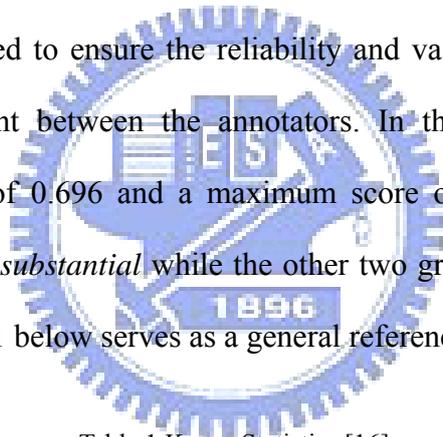
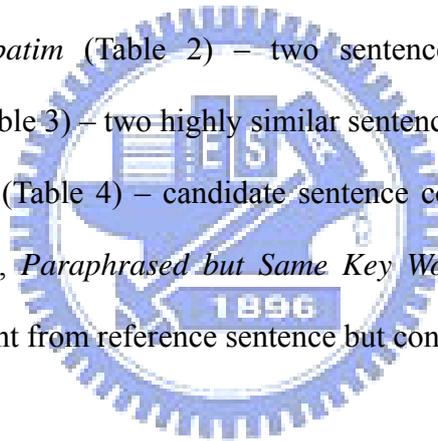


Table 1 Kappa Statistics [16]

Kappa	Strength of agreement
0.00	Poor
0.01-0.20	Slight
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Substantial
0.81-1.00	Almost perfect

Sentence pairs which were annotated differently by the annotators were removed, and the end product was 978 pairs of sentences in which only 32 pairs were annotated as candidates of plagiarism (see Appendix 3 for the 32 pairs of sentences). The number of positive plagiarism examples in the *abstract* corpus was rather disappointing, and further analysis of the 32 pairs showed that majority of the pairs came from seven papers. This observation indicates that some authors do use similar or exact sentences from the main text for their abstract. The small number of valid plagiarizing pairs was due to the fact that selection of the research papers in the beginning was random. Figure 19 is a pie chart that shows the statistics of each plagiarism type in the 32 pairs of sentences. Definitions of the plagiarism types are as follows: *Complete Verbatim* (Table 2) – two sentences are exactly the same, *Substantial Verbatim* (Table 3) – two highly similar sentences that differ by only a few words, *Lifted Sentences* (Table 4) – candidate sentence copied one or more phrases from reference sentence, *Paraphrased but Same Key Words* (Table 5) – candidate sentence is rather different from reference sentence but contains the same key words.



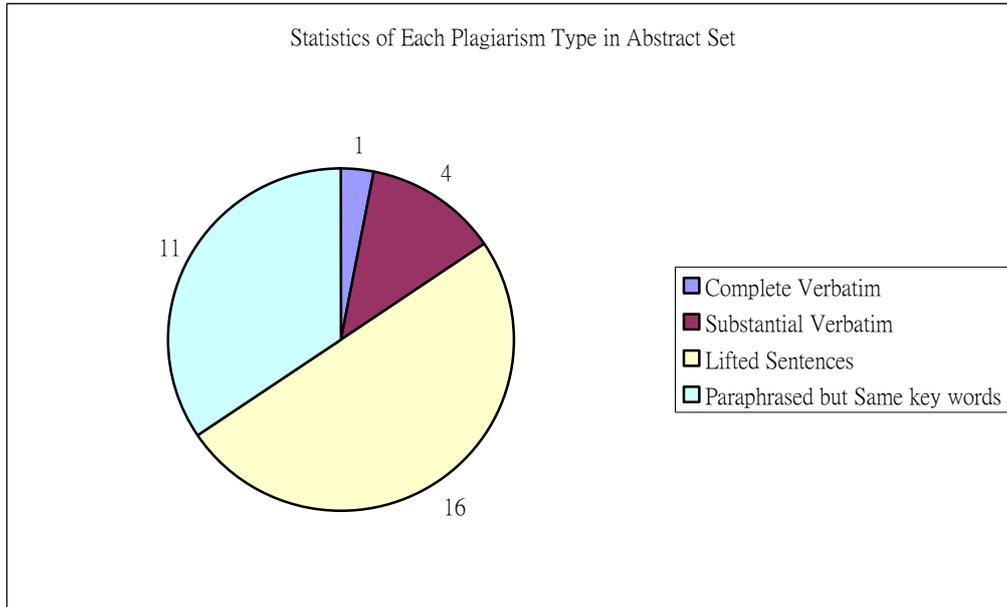


Figure 19 Statistics of Each Plagiarism Type in the *Abstract Set*

Table 2 Example of Verbatim Copy

Candidate sentence	Reference sentence
this article draws on the poststructuralist theory of consumption developed by michel de cerateau to consider plagiarism as a tactic deployed by consumers in their attempts to negotiate the demands of an increasingly commodired tertiary education sector	this article draws on the poststructuralist theory of consumption developed by michel de cerateau to consider plagiarism as a tactic deployed by consumers in their attempts to negotiate the demands of an increasingly commodired tertiary education sector

Table 3 Example of Substantial Verbatim

Candidate sentence	Reference sentence
it is also concluded that there is a growing need for uk institutions to develop cohesive frameworks for dealing with student plagiarism that are based on prevention supported by robust detection and penalty systems that are transparent and applied consistently	there is a growing need for uk institutions to develop cohesive frameworks for dealing with student plagiarism that are based on prevention supported by robust detection and penalty systems that are transparent and applied consistently

Table 4 Example of Lifted Sentences

Candidate sentence	Reference sentence
this paper reviews the literature on plagiarism by students much of it based on north american experience to discover what lessons it holds for institutional policy and practice within institutions of higher education in the uk	conclusion there is an extensive literature on plagiarism by students particularly in the context of north america experience but it clearly holds important lessons for institutional policy and practice within institutions of higher education in the uk

Table 5 Example of Paraphrased but Same Key Words

Candidate sentence	Reference sentence
those who plagiarized least incorporated direct quotations more effectively used fewer quotations and synthesized information and ideas better than did the others	the two students who plagiarized least used minimal quotations see table 1 and used them effectively capably synthesizing their information and ideas a challenge in a task that required primarily reporting of information

The initial motivation for generating the *paraphrased* set was to add more plagiarizing examples. One possible way is to retrieve plagiarism examples from the Internet, where Websites focusing on the topic of plagiarism can be found. In those Websites, usually one can find plagiarism examples in short passages of about a few sentences long. Hence, the query “plagiarism examples” was sent to Google, and only paraphrased plagiarism examples were retrieved manually. *Paraphrased* set mainly consists of plagiarism types like addition, deletion or substitution of words in the original content, change of sentence structure, and partial verbatim copy. A total of 30 plus plagiarism examples were retrieved. All the examples were retrieved from the first 10 pages of the returned search results because repeated examples appeared after the first few pages of search results and the relevancy of search results began to decrease. The plagiarizing sentences were paired up with the corresponding original

sentences manually; therefore each pair was a valid example of plagiarism.

4.2 Experiments

Since the effectiveness of the proposed methods in detecting plagiarism were unknown, all methods were tested using the *abstract* set with different thresholds, in multiples of 10 starting from 0 with the maximum threshold being 100. The threshold values can be taken as from 0 to 1.0 and multiply by 100. This is because F-measures were multiplied by 100 and shown percentage-wise. In order to work with the data set, the comparison procedure was modified so that each reference sentence only compared with its corresponding candidate sentence, not with all the candidate sentences as described in Section 3.3.1.1. Every pair of sentences received a score from each method and the score was compared with the threshold. The comparison would yield four different outcomes namely true positive (TP), false positive (FP), true negative (TN), and false negative (FN). If the score was larger than the threshold and the pair was annotated as *plagiarism*, then it would be considered as the method had correctly identified a plagiarism instance and TP would be recorded; on the other hand, a FP would be recorded instead if the pair was annotated otherwise. The same logic applied when judging TN and FN but with opposite criteria, i.e. the score was smaller than the threshold and the pair was annotated as *not plagiarism*.

With the four values, sensitivity and specificity could be calculated. Sensitivity and specificity are measures used to evaluate the performance of binary classification, *plagiarism* and *not plagiarism* in this case.

$$Sensitivity = \frac{TP}{TP + FN} \quad (26)$$

$$\textit{Specificity} = \frac{TN}{TN + FP} \quad (27)$$

By looking at the problem from an information retrieval perspective, the number of plagiarism pairs was the number of relevant documents. Therefore the performance of each method could be evaluated in terms of recall, precision, and F-measure. While sensitivity was actually the equivalent of recall, precision could be calculated as Equation (28):

$$\textit{Precision} = \frac{TP}{TP + FP} \quad (28)$$

F-measure was further derived from recall and precision as Equation (29):

$$\textit{F-measure} = \frac{2 \times \textit{Recall} \times \textit{Precision}}{\textit{Recall} + \textit{Precision}} \quad (29)$$

Note: recall, precision, and F-measure here refer to TPs, FPs, TNs, and FNs of the *abstract set*. They should not be confused with the measures mentioned in Chapter 3.3.

Finally, all the values of a particular method with different thresholds could be summarized and expressed as Table 6 below:

Table 6 Examples of Sensitivity, Specificity, and F-measure

Threshold	TP	FP	TN	FN	Sensitivity	Specificity	Precision	F-measure
0	32	946	0	0	1	0	0.03272	0.063366
10	31	884	62	1	0.96875	0.065539	0.03388	0.06547
20	31	701	245	1	0.96875	0.258985	0.04235	0.081152
30	30	381	565	2	0.9375	0.597252	0.072993	0.13544
.
.
.
100	6	0	946	26	0.1875	1	1	0.315789

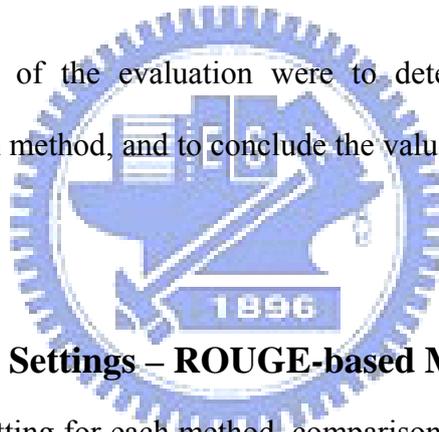
Each method under a specific preprocessing setting would generate a table like Table 6. For ROUGE, there are four possible settings: both stopwords and stemming are applied (SW+SM), stopwords are removed (SW), stemming is applied (SM), and neither stopwords nor stemming are applied (No Pre). Meanwhile, there are only two possible settings for WordNet-based methods: no preprocessing and SW. Stemming is not compatible with WordNet because some words have stems that WordNet cannot recognize. For example, “happy” will be stemmed to “happi”; in such case, WordNet is unable to find a match in its database.

Although the *paraphrased* data set was originally meant for increasing the number of true positives, the nature of the *abstract* set and *paraphrased* set is not similar at all. While the *abstract* set is made up of research papers, the *paraphrased* set is comprised of various types of text ranging from discussion of Shakespeare’s

literature to scientific statements. As a result, these two sets were not merged together. Furthermore, the *paraphrased* set does not have true negatives so it cannot be used for evaluation in the same way as the *abstract* set. Instead, this research tried to utilize the nature of the *paraphrased* set. The idea was based on the fact that the results would be either true positives or false negatives if the methods were tested with the *paraphrased* set. Observation of the results would be made in hope of learning the strengths and weaknesses of each method by looking specifically at the difference in performance of each method in different types of plagiarism.

4.3 Evaluation

The primary goals of the evaluation were to determine and recommend a desirable setting for each method, and to conclude the value of WordNet in plagiarism detection.



4.3.1 Recommended Settings – ROUGE-based Methods

To recommend a setting for each method, comparison between the performances of the same method under different settings was necessary. Comparison was made easier by creating Table 7 and generating Figure 20 as follows:

Table 7 F-measures of Unigram under Different Settings

Threshold	F(SW+SM)	F(SW)	F(SM)	F(No Pre)
0	0.063366	0.063366	0.063366	0.063366
10	0.064854	0.07721	0.066184	0.069264
20	0.094512	0.133333	0.084584	0.098257
30	0.221374	0.31016	0.172308	0.219608
40	0.396694	0.45098	0.419355	0.490566
50	0.638889	0.67692	0.6875	0.70968
60	0.65385	0.666667	0.625	0.625
70	0.653061	0.638298	0.439024	0.439024
80	0.4	0.4	0.358974	0.358974
90	0.222222	0.222222	0.27027	0.27027
100	0.171429	0.117647	0.060606	0.060606

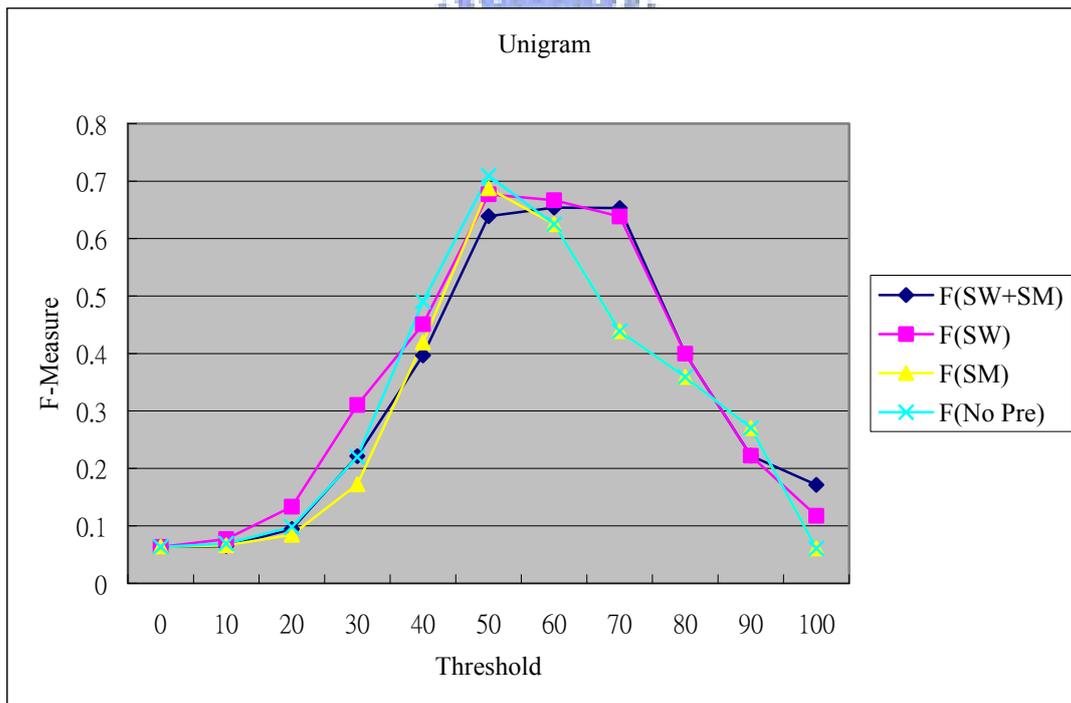


Figure 20 Line Graph of Unigram under Different Preprocessing Settings

By visualizing the results (Figure 20), the characteristics and performance of each setting were relatively clearer than just by looking at the numbers. Nevertheless, the lines were close sometimes and more detailed information was needed to make the right judgment. In this case, F-measure with No Pre - F(No Pre) and F-measure with

SM - F(SM) had the top two highest F-measures at threshold=50 while F(SW+SM) and F(SW) formed smoother curves on the graph. By only looking at threshold \geq 50 as in Figure 21, F(SW+SM) and F(SW) obviously performed better than F(SM) and F(No Pre); hence the choices were cut down to two: F(SW+SM) and F(SW). For partial graphs of other methods please refer to Appendix 2.

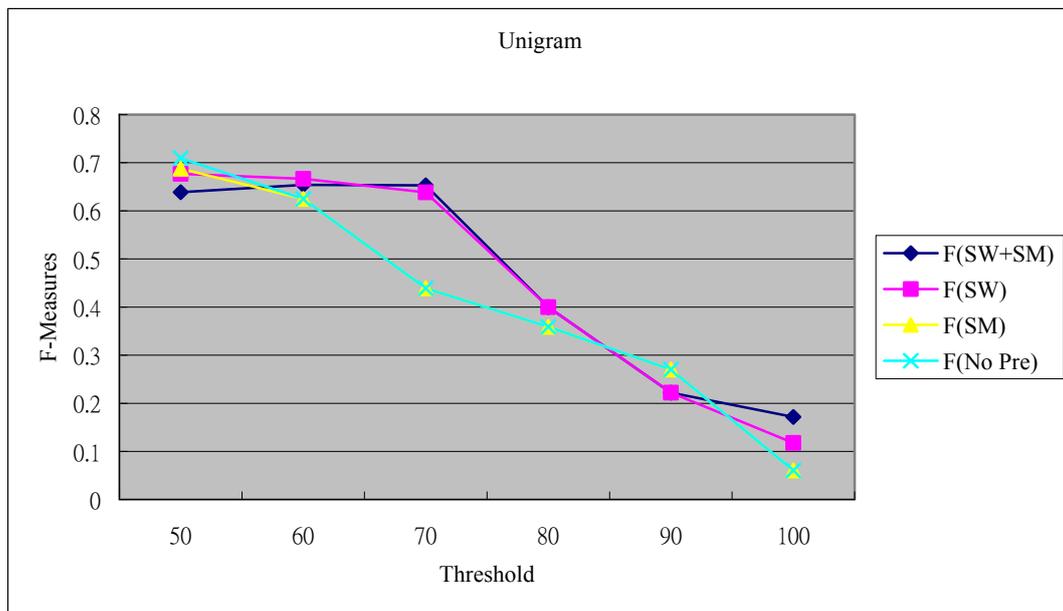


Figure 21 Partial Graph of Figure 20

Here, the information of TPs, FPs, TNs, and FNs could provide some insights from another perspective. Table 8 and Table 9 below were variations of Table 6.

Table 8 Results of F(SW+SM)

Stopword+Stemming				
Threshold	TP	FP	TN	FN
50	23	17	929	9
60	17	3	943	15
70	16	1	945	16
80	8	0	946	24
90	4	0	946	28
100	3	0	946	29

Table 9 Results of F(SW)

Stopword				
Threshold	TP	FP	TN	FN
50	22	11	935	10
60	17	2	944	15
70	15	0	946	17
80	8	0	946	24
90	4	0	946	28
100	2	0	946	30

At threshold=50, the number of FPs for both settings were unacceptable, but FPs dropped significantly at threshold=60. And by observing threshold \geq 60, F(SW) had slightly lower number of FPs and TPs than F(SW+SM). Up to this point, a recommended setting for unigram was determined – stopwords removal with threshold=60. One reason for sacrificing TPs in exchange of lower FPs is that the definition of plagiarism does not depend on the number of plagiarism instances, but depend on whether or not a plagiarism instance really exists. With this criterion in mind, as long as the number of TPs is substantial, lower FPs will be the top priority. The other reason is that no system so far can guarantee fully automatic detection,

human judgment is mandatory at the very end of the detection process; therefore, less time is required to filter out FPs by the evaluator.

Table 10 shows the recommended settings for the remaining ROUGE methods, which were determined under the same principle. Please refer to Appendix 1 for the line graph of other ROUGE methods.

Table 10 Summarization Table of Recommended Settings

Methods	Recommended Setting
Unigram	Stopwords, Threshold=60
Bigram	No Preprocessing, Threshold=40
Trigram	Stemming, Threshold=30
4-gram	Stopwords, Threshold=30
Skip-bigram	Stopwords & Stemming, Threshold=30
LCS	Stopwords & Stemming, Threshold=50

4.3.2 Recommended Settings - WordNet

As mentioned in 3.3.4.2, the weightings for different depths in the hypernym/hyponym relationship were intuitively set. Therefore, evaluation on the performance of relationship-based WordNet was necessary before going further to compare the performances between the two WordNet methods and unigram. To get a general idea about the performance of current weightings, the *abstract* set was run on both synonyms-based and relationship-based methods under two different settings, stopwords and no preprocessing. At first glance on the F-measures of the two methods, relationship-based method was significantly lower than synonyms-based. The cause of this poor performance could be explained by observing Table 11 below:

Table 11 Relationship-based method Stopwords Removed (Initial Weighting Scheme)

Threshold	TP	FP	TN	FN
0	32	946	0	0
10	32	913	33	0
20	31	809	137	1
30	29	607	339	3
40	29	374	572	3
50	26	152	794	6
60	20	40	906	12
70	16	8	938	16
80	11	1	945	21
90	6	0	946	26
100	4	0	946	28

Although relationship-based method was more capable than synonyms-based method in identifying TPs correctly, this strength was overshadowed by its inability to correctly identify TNs, and consequently the number of FPs was significantly higher. The numbers in Table 11 suggested that the weightings were too high. Besides looking at Table 11 observations were made on word pairs whose relationship depth was three or lower. It was not hard to realize that quite a number of word pairs that had a relationship depth of three were in fact not closely related. The following is an example of how two words that appeared in the *abstract* set, *play* and *ownership*, were linked through hypernym relationship in WordNet.

Words: turn, play -- (game) the activity of doing something in an agreed succession Words: activity -- any specific behavior Words: control -- the activity of managing or exerting control over something Words: possession, ownership -- the act of having and controlling property
--

Figure 22 Hypernym Relationship in WordNet for Play and Ownership

Figure 22 above is the output of Java applications of JWNL, with tags and other content removed for clarity. Even though *play* and *ownership* are within a depth of three in one of their senses, *play* has to go up the hierarchy to *activity*, a general definition of all behavior, before going down two levels in the hierarchy to arrive at *ownership*. Obviously, a weighting of 0.7 did not accurately reflect this relationship. At depth of one, the relationship between words is more reasonable as shown in Figure 23.

Words: support -- (support materially or financially) Words: provide , bring_home_the_bacon -- (supply means of subsistence; earn a living)
--

Figure 23 Hypernym Relationship in WordNet for Support and Provide

Based on the above observations, a new weighting scheme should assign smaller weights in a decreasing rate according to the depth, i.e. depth three should see more decrease between old and new weight while depth two should see less decrease in new weight. By this rule, the second weighting scheme was as follows: 1.0 for depth 0, 0.85 for depth 1, 0.5 for depth 2, and 0.2 for depth 3. When efforts were spent in determining a new weighting scheme for hypernym relationship, the weightings for hyponym were not as much a concern because hyponym relationship was not as common as hypernym relationship through observation. This means that most of the word pairs were scored based on their hypernym relationship. And whenever there was a hyponym relationship between the words, the default weight was accurate

enough to reflect the relationship between the words. Once new weightings had been set, the *abstract* set was run on relationship-based method. This time, instead of comparing with the synonyms-based method, the performance of second weighting scheme was compared with the first scheme.

Table 12 Relationship-based method Stopwords Removed (2nd Weighting Scheme)

Threshold	TP	FP	TN	FN
0	32	946	0	0
10	32	895	51	0
20	31	738	208	1
30	29	454	492	3
40	28	178	768	4
50	23	55	891	9
60	18	14	932	14
70	14	1	945	18
80	9	0	946	23
90	6	0	946	26
100	3	0	946	29

The numbers between Table 11 and 12 clearly showed that 2nd weighting scheme could better represent the hypernym/hyponym relationship. Despite slight decrease in TPs, FPs dropped by a greater margin. Since the overall performance of relationship-based method improved by lowering the weights, 3rd weighting scheme was tested and compared with the other two schemes. The latest weightings were as follows: 1.0, 0.7, 0.4, and 0.1. The weights corresponded from depth 0 to 3 respectively.

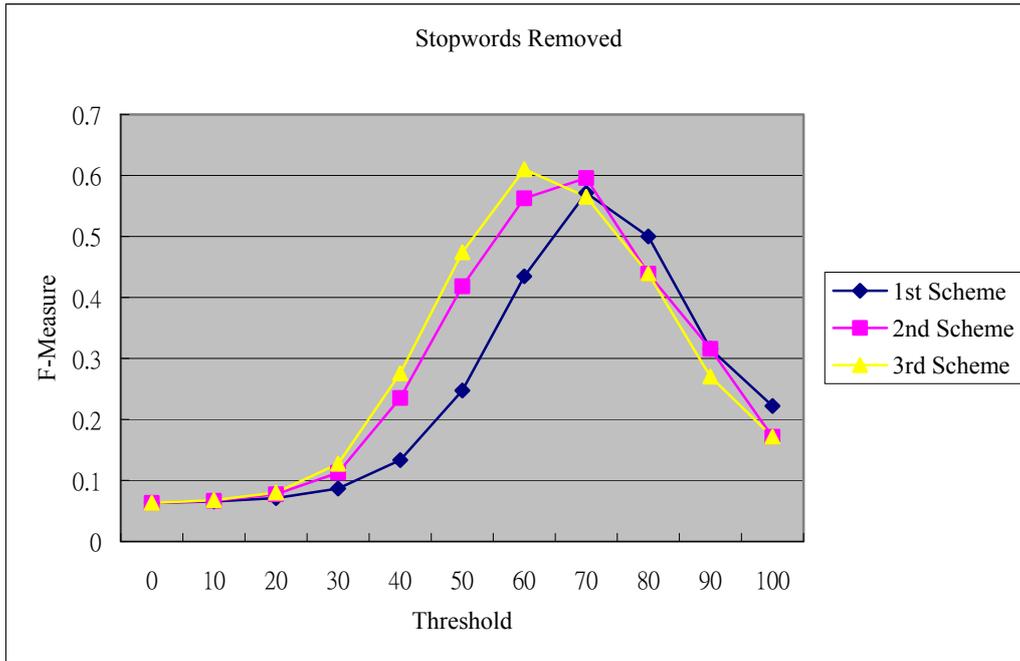


Figure 24 F-Measures of Three Schemes with SW Removed

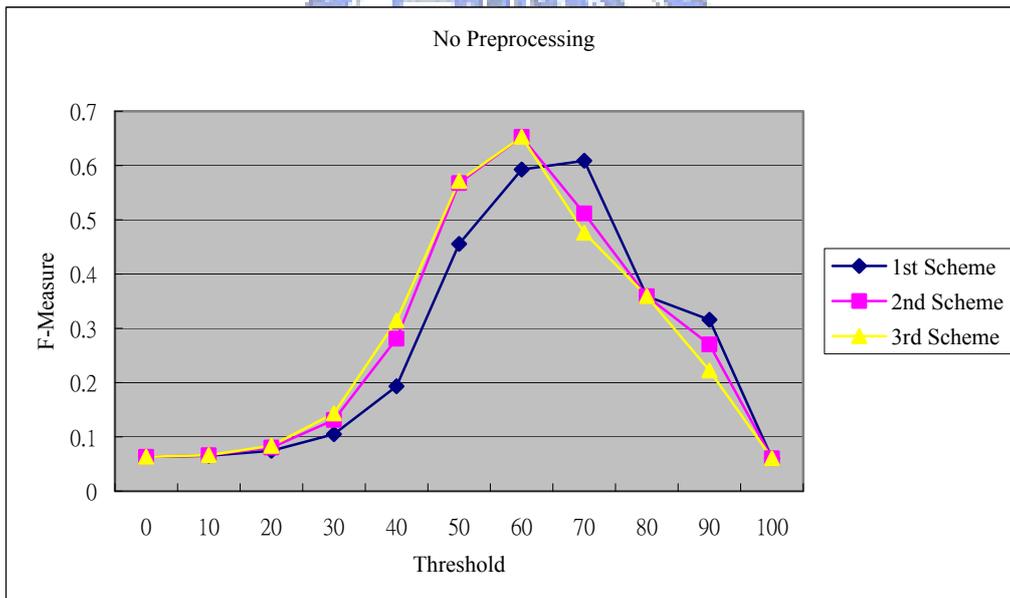


Figure 25 F-Measures of Three Schemes with No Preprocessing

As Figure 24 and 25 indicates, there was not much difference between 2nd weighting scheme and 3rd weighting scheme. By excluding 1st weighting scheme due to its large number of FPs, the best weighting scheme between the remaining two candidates was obvious by looking at Figure 24 and 25, which showed that the

highest F-measures ranged from 60 to 70, and at threshold=70, 2nd scheme was the best of the two when stopwords were removed. Even under no preprocessing, at threshold=60, 2nd scheme had the highest F-measure too. Overall, 2nd scheme performed better than 3rd scheme when threshold was high, and at the same time, 2nd scheme had a smoother curve than 1st scheme. In conclusion, 2nd scheme should be adopted for relationship-based method. Having decided the scheme, different preprocessing settings could be compared as Figure 26, which showed that SW performed better than No Pre at threshold=70 and beyond. At last, the recommended setting for relationship-based method was the adoption of 2nd weighting scheme, with stopwords removed and threshold=70.

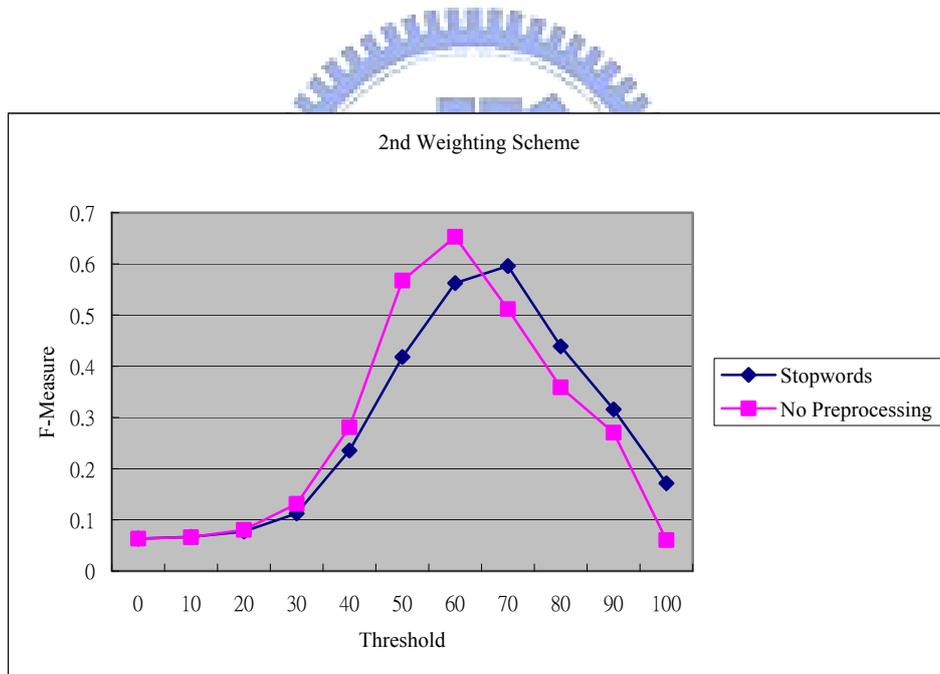


Figure 26 2nd Weighting Scheme under Different Settings

Recommendation of the setting for synonyms-based method was easier as one could immediately tell which type of preprocessing was better in Figure 27. Synonyms-based method performed better with stopwords removed and the number of FPs at threshold=60 were less than the number of FPs at threshold=50. Therefore,

synonyms-based method should be applied at threshold=60 with stopwords removed.

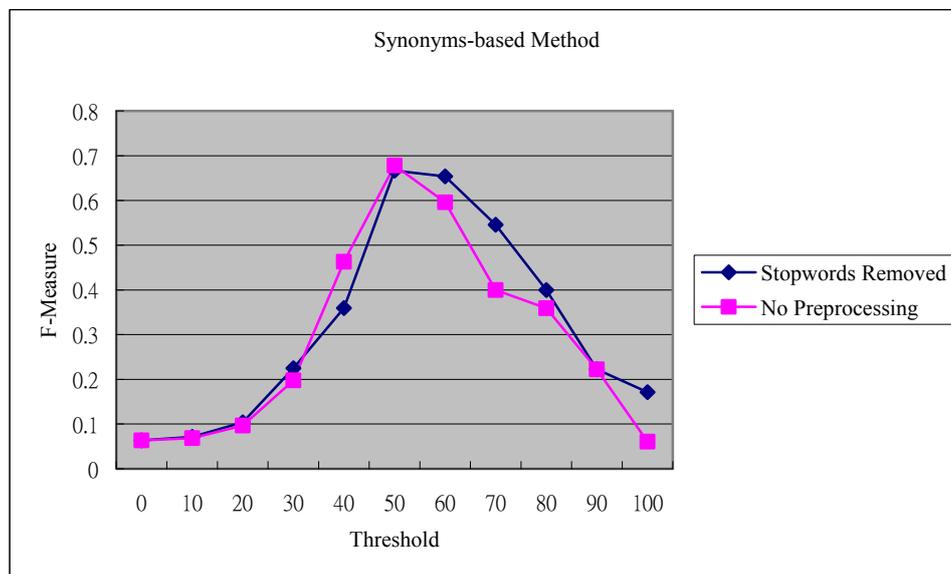


Figure 27 Synonyms-based Method under Different Settings

4.3.3 Evaluation of WordNet-based Methods for *Abstract Set*

Because both synonyms-based and relationship-based methods were derived from unigram, therefore technically unigram was the best candidate among ROUGE to be compared with WordNet-based methods. However, to ensure valid comparison between the methods, like WordNet-based methods, POS tags had to be included in unigram during the matching of tokens. Besides this modification, unigram (SW+SM) was compared with the two WordNet-based methods (SW) – Figure 28, and unigram (SM) was compared with the two other methods (No Pre) – Figure 29. The main reason for such match-ups was due to the fact that WordNet transformed the words into their original form in the database. For example, from “paid” to “pay”. Such process was similar to the concept of stemming and hopefully the match-ups would make the comparisons more meaningful.

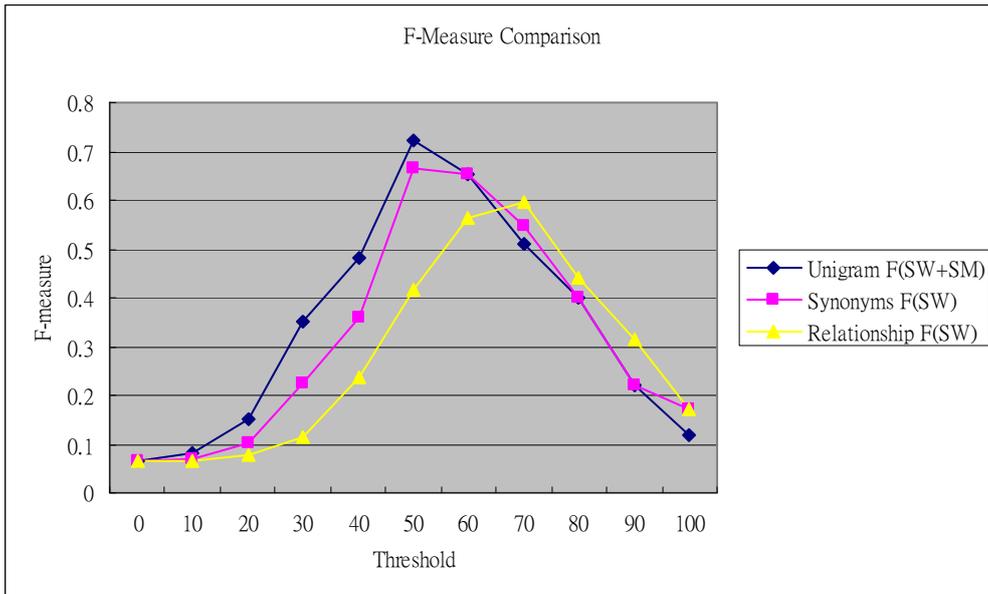


Figure 28 F-Measures Comparison with Stopwords Removed for WordNet-based Methods

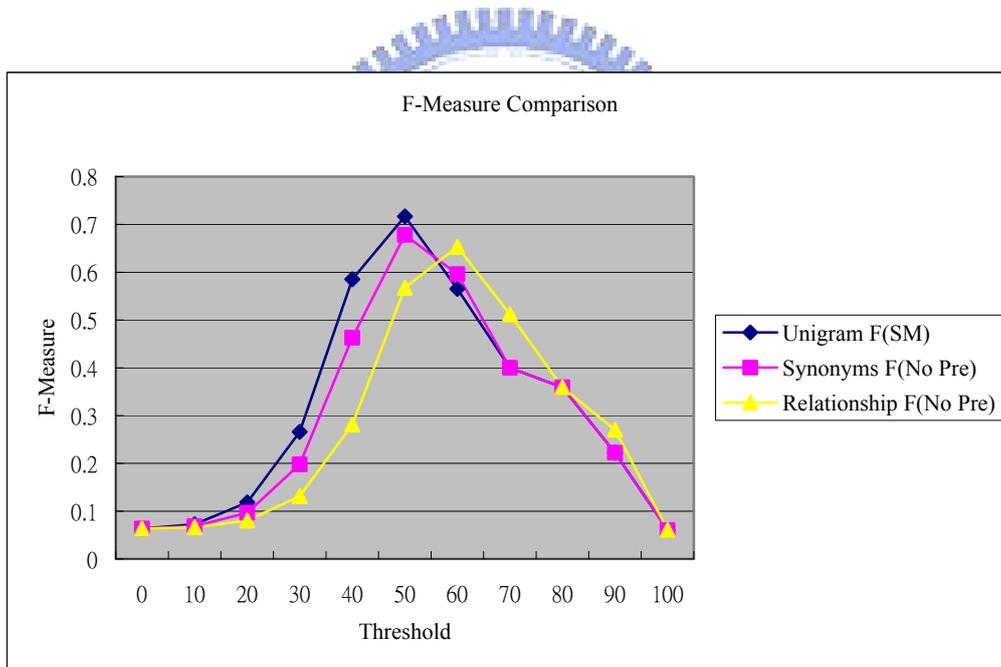


Figure 29 F-Measures Comparison with No Preprocessing for WordNet-based Methods

Figures 28 and 29 above showed that on overall it was hard to tell if WordNet-based methods were better than unigram in the *abstract* set. However, by looking at thresholds ≥ 60 , i.e. beyond the recommended thresholds of the three methods, WordNet-based methods did perform better than unigram as supported by Figure 30 and 31. The outcomes were probably due to the nature of the *abstract* set –

most valid plagiarism pairs in the corpus were made up of verbatim type of plagiarism, and there were not as many substitutions of words. As a result, WordNet was not being fully utilized and could not be of much help.

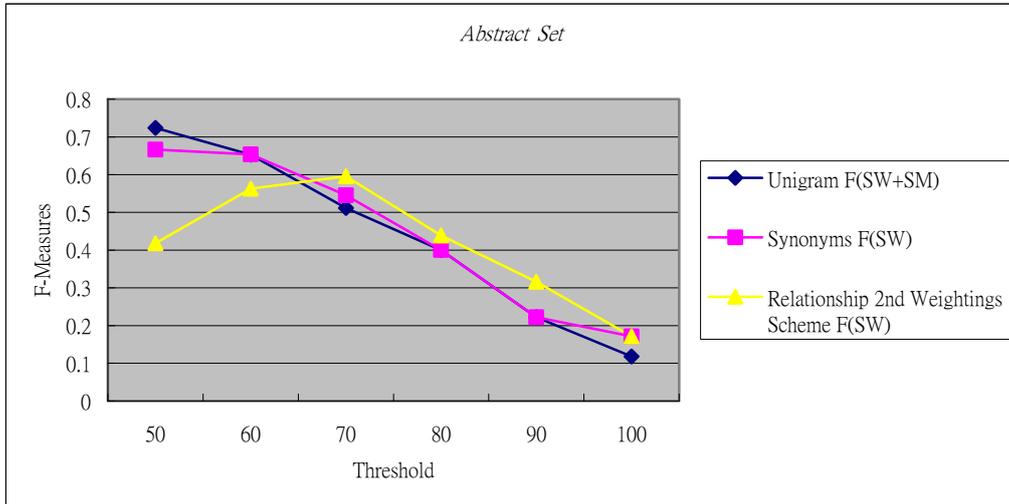


Figure 30 Comparison Graph for *Abstract Set* with Stopwords Removed for WordNet-based Methods

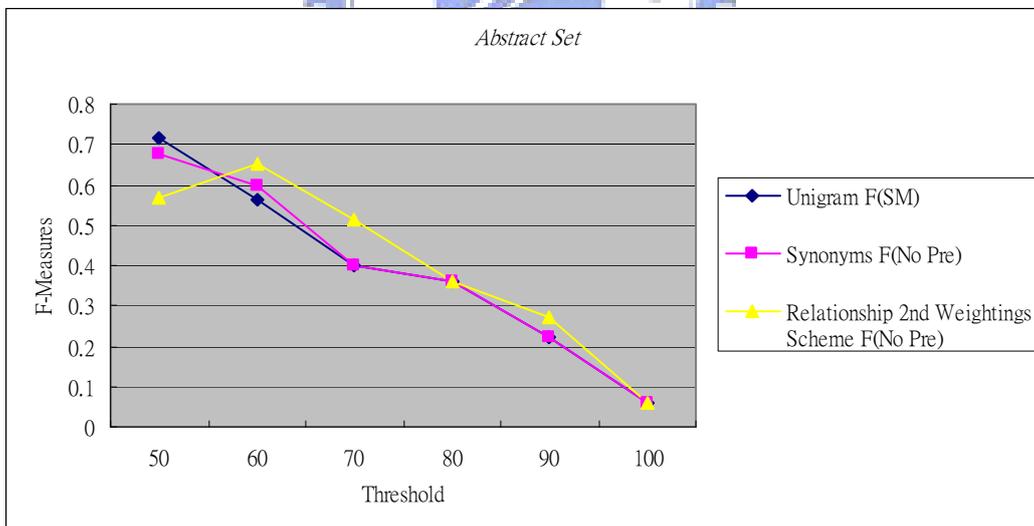


Figure 31 Comparison Graph for *Abstract Set* with No Preprocessing for WordNet-based Methods

4.3.4 Evaluation of WordNet-based Methods for *Paraphrased Set*

In the previous Section, WordNet-based methods performed moderately in the *abstract* set. Further evaluation on WordNet-based methods was done with the *paraphrased* set. Evaluation was based on the number of TPs and FNs for each method and comparison of the results was made. Again, POS tags were included in unigram and the same match-ups for preprocessing were deployed. As the nature of plagiarism examples changed, the results were very different from Section 4.3.3. Figure 32 and 33 show the same pattern of results, with relationship-based method on top, followed by synonyms, and unigram was at the bottom of the graph. Although this evaluation could not determine the effectiveness of WordNet-based methods in identifying TNs, one affirmation was that WordNet-based methods were able to identify substitution of words better than unigram since this was the only difference between these two approaches.

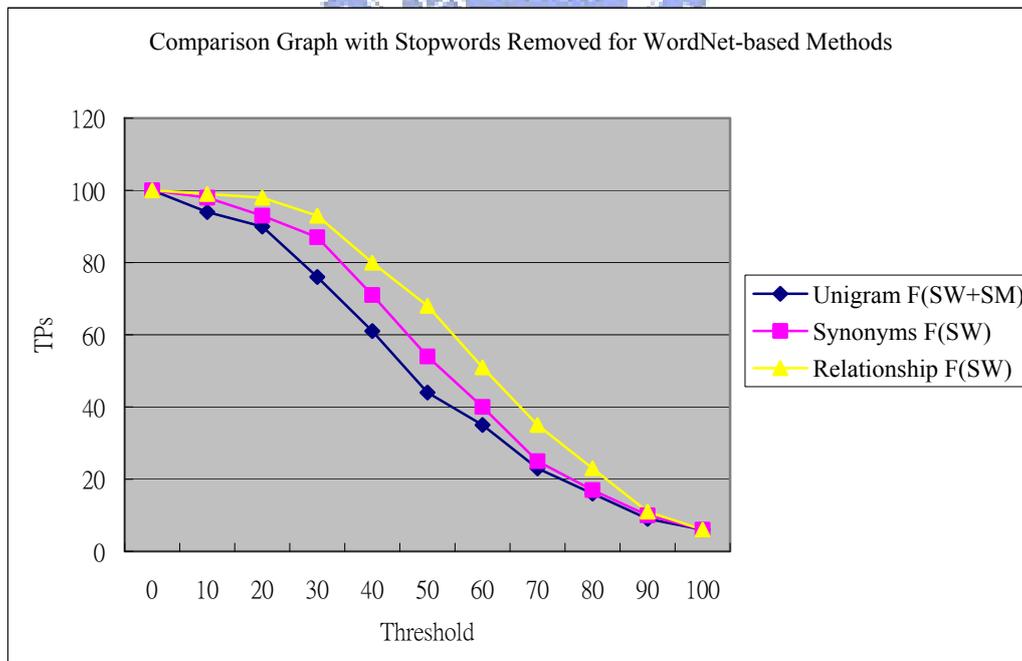


Figure 32 TPs of Unigram and WordNet-based Methods with Stopwords Removed

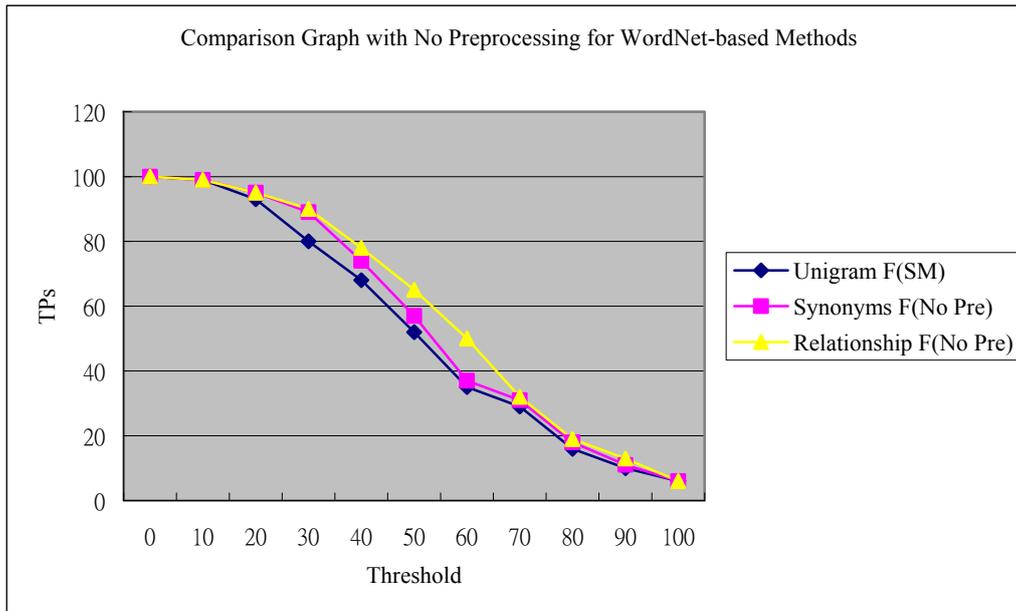


Figure 33 TPs of Unigram and WordNet-based Methods with No Preprocessing

4.3.5 Evaluation of Google MI Method for *Abstract* and *Paraphrased* Sets

The Google MI method was tested with the two data sets under one preprocessing setting only. Stopwords were removed because it was meaningless to calculate the mutual information between a stopwords and a given word. Stemming was not applied for the same reason for WordNet-based methods. Mutual information of all word pairs were calculated if two words in a pair did not match. Since there was only one set of F-measures for the Google MI method, only the best threshold remained to be determined. Through observation of the experimental results, threshold was set at 80. Recommendation of high threshold was due to the fact that Google method had too many false positives, which were probably because all pairs of words would have a mutual information score under Google method thus increasing the similarity between two sentences. Table 13 shows the results of Google MI and three other methods when threshold=60. One can see that the number of FPs of Google MI were significantly larger.

Table 13 Results at Threshold=60

	TP	FP	TN	FN
Google MI	26	187	759	6
Unigram	16	1	945	16
Synonyms-based	17	3	943	15
Relationship-based	18	14	932	14

Figure 34 shows that the Google MI method performed better at threshold \geq 80 and Figure 35 shows that the Google MI method could detect more plagiarism examples in *paraphrased* set; however, the observations could not prove the value of Google mutual information because of the number of false positives.

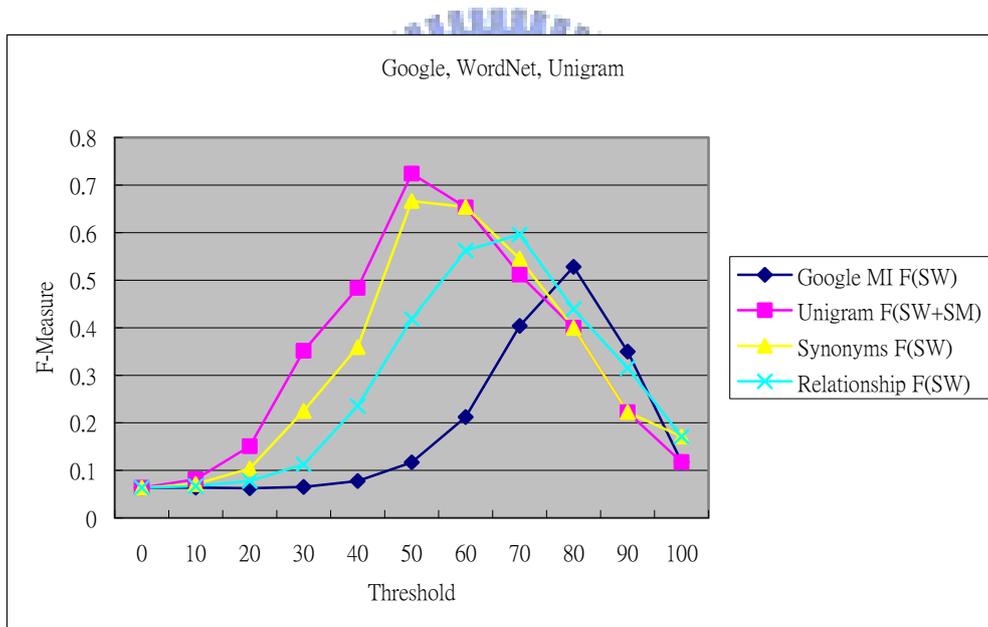


Figure 34 Comparison Graph for *Abstract* Set with Stopwords Removed Including Google Method

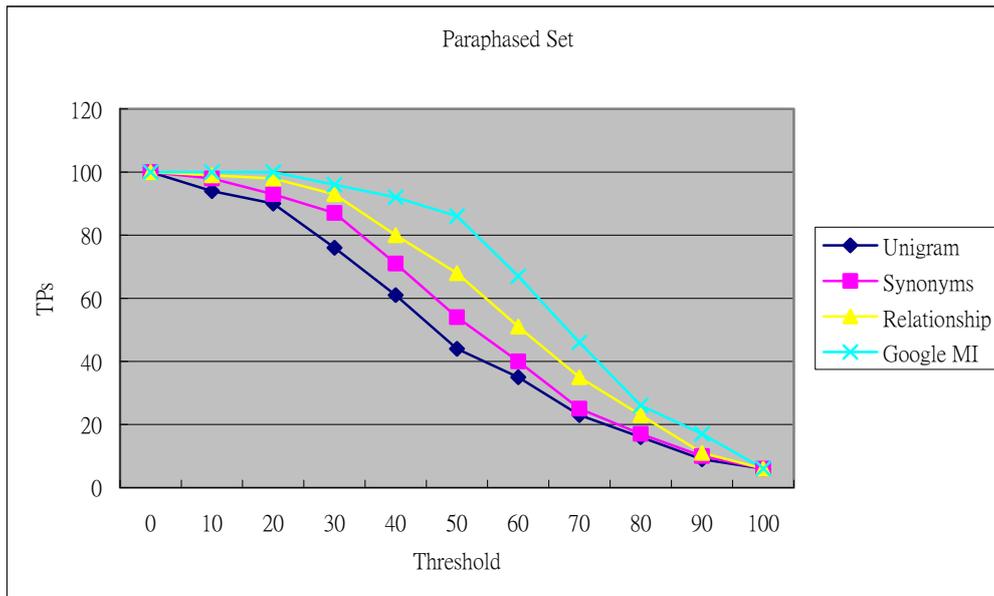


Figure 35 Comparison Graph for *Paraphrased Set* with Stopwords Removed Including Google Method

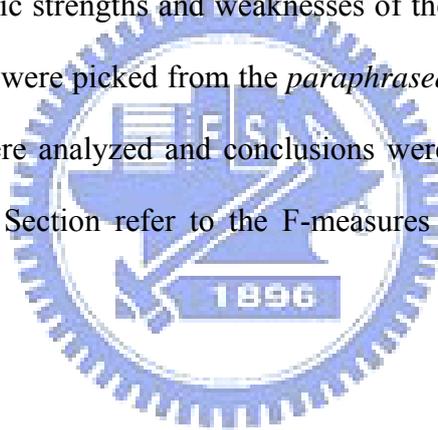
At this point, ideal threshold and preprocessing setting for each method were empirically determined. Table 14 shows the performance of all the methods under different settings at their recommended threshold. By observing Table 14, one interesting discovery was made. Under every different setting, the highest F-measure was either from Skip-bigram or LCS. One possible explanation may be that both allow gaps between matching tokens but at the same time require tokens to be in-sequence. These two rules balance the values of recall and precision, and lead to higher F-measure.

Table 14 F-measures under Different Settings Recommended Threshold

	F(SW+SM)	F(SW)	F(SM)	F(No Pre)
Unigram, T=60	0.653846154	0.666667	0.625	0.625
Bigram, T=40	0.654545455	0.666667	0.6666667	0.66667
Trigram, T=30	0.571428571	0.583333	0.627451	0.627451
4-gram, T=30	0.565217391	0.565217	0.5652174	0.565217
Skip-bigram, T=30	0.62962963	0.627451	0.690909	0.66667
LCS, T=50	0.7037037	0.7037	0.6538462	0.627451
Synonyms-based, T=60	0.653846154	NA	0.5957447	NA
Relationship-based, T=70	0.595744681	NA	0.5116279	NA
Google MI, T=80	NA	0.528	NA	NA

4.3.6 Strengths and Weaknesses of Each Method

To assess the specific strengths and weaknesses of the methods, a few examples for each plagiarism type were picked from the *paraphrased* set and tested with all the methods. The results were analyzed and conclusions were drawn from the analysis. The F-measures in this Section refer to the F-measures between two sentences in Chapter 3.



Example 1:

Candidate Sentence: brown dwarfs **are difficult to locate and** rank among the most elusive objects in the universe

Reference Sentence: brown dwarfs rank among the most elusive objects in the universe

Table 15 F-Measures of Example 1

Methods	SW+SM	SW	SM	No Pre
Unigram	85.714	85.714	81.481	81.481
Bigram	66.667	66.667	72	72
Trigram	40	40	60.87	60.87
4-gram	0	0	57.143	57.143
Skip-bigram	40	40	60.392	60.392
LCS	85.714	85.714	80.952	80.952
Synonyms	78.571	NA	77.778	NA
Relationship	71.429	NA	74.074	NA
Google MI	NA	76.786	NA	NA
Unigram (POS)	71.429	71.429	74.074	74.074

For example 1, all the methods had relatively high F-measures. This indicates that if consecutive new words are inserted into the reference sentence and the lengths of the sentences do not differ much, all the methods still should be able to detect plagiarism. The zeros from 4-gram were due to the fact that there was not any matching four gram after stopwords were removed. NAs in Table 14 and the following tables mean that synonyms-based and relationship-based methods were not tested under those settings as mentioned in Section 4.3.3.

Example 2:

Candidate Sentence: in this view meaning is determined by **the** real world and is **therefore** external to the **learner**

Reference Sentence: meaning is **eventually** determined by this real world and is thus external to the understander

Table 16 F-Measures of Example 2

Methods	SW+SM	SW	SM	No Pre
Unigram	71.429	71.429	75	75
Bigram	50	50	46.667	46.667
Trigram	40	40	21.429	21.429
4-gram	25	25	7.692	7.692
Skip-bigram	53.333	53.333	49.697	49.697
LCS	71	71	68.204	68.204
Synonyms	71.429	NA	75	NA
Relationship	71.429	NA	75	NA
Google MI	NA	83.76	NA	NA
Unigram (POS)	71.429	71.429	75	75

In this example, bigram, trigram, and 4-gram had lower scores than they had in Example 1 because of addition, deletion, and substitution of words throughout the candidate sentence, causing less matching of consecutive tokens. On the other hand, unlike the three methods above, skip-bigram and LCS allowed in-sequence skip within the sentence; as a result, skip-bigram was slightly better than bigram, and LCS still had satisfactory F-measures.

Example 3:

Candidate Sentence: those complexes that contain paired electrons are **repelled** by a magnetic field and are said to be **diamagnetic** whereas those with no paired electrons are attracted to such a field and are called **paramagnetic**

Reference Sentence: those complexes that contain unpaired electrons are **attracted** into a magnetic field and are said to be **paramagnetic** while those with no unpaired electrons are repelled by such a field and are called **diamagnetic**

Table 17 F-measures of Example 3

Methods	SW+SM	SW	SM	No Pre
Unigram	84.615	84.615	88.235	88.235
Bigram	33.333	33.333	66.667	66.667
Trigram	0	0	50	50
4-gram	0	0	32.258	32.258
Skip-bigram	37.5	37.5	64.138	64.138
LCS	53	53	73	73
Synonyms	84.615	NA	88.235	NA
Relationship	84.615	NA	88.235	NA
Google MI	NA	90.161	NA	NA
Unigram (POS)	84.615	84.615	88.235	88.235

Example 3 is a representation of changing the sentence structure/order. As the sequence of the words in the reference sentence had been changed in the candidate sentence, LCS was obviously affected by this type of plagiarism. Because usually LCS had similar scores like unigram but in this case its score was significantly lower than unigram. Actually bigram to 4-gram were also affected, especially after the stopwords were removed. This was probably because the order of open class words had been changed forming different n-grams.

Example 4:

Candidate Sentence: the **increase** of industry the growth of cities and the **explosion** of the population were three **large** factors of nineteenth century america

Reference Sentence: the **rise** of industry the growth of cities and the **expansion** of the population were the three **great** developments of late nineteenth century american history

Table 18 F-measures of Example 4

Methods	SW+SM	SW	SM	No Pre
Unigram	50	50	72.34	72.34
Bigram	27.273	27.273	48.889	48.889
Trigram	10	10	37.209	37.209
4-gram	0	0	29.268	29.268
Skip-bigram	20	20	55.495	55.495
LCS	49.68	49.68	72.221	72.221
Synonyms	66.667	NA	80.851	NA
Relationship	81.667	NA	88.511	NA
Google MI	NA	66.202	NA	NA
Unigram (POS)	50	50	72.34	72.34

This example demonstrated how original content could be modified with synonyms. Totally different words were used leading to false negative judgment on the similarity between two sentences. The impact was more obvious after stopwords were removed. However, WordNet-based methods performed pretty well in such situation, relationship-based method in particular.

The strengths and weaknesses of the methods were determined through the analysis of the above four examples. Meaningful observations were made to learn about the characteristics of each method and assess the value of each method in plagiarism detection. One of the observations was that stemming did not have obvious influence on the detection results. This argument was derived from the statistics of the *abstract* set and was further confirmed by the *paraphrased* set. The cause of this

phenomenon is probably due to the scope of detection in this research; the probability of words “*computer*”, “*computation*”, and “*compute*” to appear in a pair of sentences can be reasonably assumed to be lower than the probability of the same words to appear in a pair of documents, limiting the effect of stemming. The observations also raised some concerns about a couple of probable scenarios, which have not been discussed in this research. The scenarios expose the weaknesses of the proposed methods and they are as follows:

Scenario 1 – if a word is substituted with a phrase bearing the same meaning. For example, “regardless” is substituted with “no matter”. In this case, “no matter” will be treated as two individual words, so even WordNet-based methods cannot detect the substitution.

Scenario 2 – two or more original sentences are combined into a long sentence, or an original sentence is split into two shorter sentences. As all proposed methods adopt sentence to sentence comparison, similarity between reference sentence and candidate sentence will be greatly affected because the number of matched words is divided by a larger denominator (long candidate sentence), or the number of matched words decreases (divide into separate sentences).

5. Conclusion

Contemporary technology has made plagiarism easy but hard to restrain. Since deliberate plagiarism will always occur, plagiarism detection is necessary to fend off potential candidates, who are afraid of being caught. This research proposed implementation of ROUGE, which was previously applied in the fields of automatic evaluation of summaries and machine translation evaluation (n-gram co-occurrence statistics). WordNet was introduced to integrate with ROUGE in order to handle as many forms of plagiarism as possible. Google was included in this research to experiment the reliability of obtaining mutual information between two words from the Internet.

Although the system is still a prototype with only a few fundamental functions, it serves as a start in developing a comprehensive tool that will help fight against plagiarism. At the same time, hopefully the system can be further developed for educational purpose by adding warnings messages and explanations regarding the detection results. By providing explanations according to the types of plagiarism, users (including students) can better understand plagiarism and know how to avoid it with real examples.

Through the analysis of the experimental results, the proposed methods are proven to have research value in the field of plagiarism detection. Each method has its strengths in dealing with certain types of plagiarism; while at the same time, each has its weaknesses in other situations. ROUGE is capable of detecting verbatim copy, and the efficiency is acceptable when comparing two complete documents. Unigram is not bounded by the “in-sequence” constraint like LCS and other n-grams, but conversely, it is may be prone to false positive. Other n-grams are stricter in matching tokens and therefore they have higher precision. While LCS and skip-bigram take a middle

ground because both allow skips when scanning through a sentence but require matching tokens to be in-sequence. WordNet extends the capability of the system by digging into the semantic aspect of words so that matching is not just by exact match, but also by the meanings of the tokens.

Being a prototype, it means that there is room for improvement. For instance, there are a few possible areas for future work of the current system. First, further confirmation on the performances of the proposed methods and recommended settings can be achieved by running tests with a larger and more diversified corpus. The corpora used in this research are relatively small compared to other research, especially the number of true positives in the *abstract* set. The results of the experiments could be affected by the nature of the corpora. However, a valid and compatible plagiarism corpus can be hard to find. Most likely, building a corpus may be an option but a lot of efforts and time will be required. Second, the proposed methods were tested and evaluated separately. Since each method has its strengths and weaknesses, particular combination of the methods may produce better results than the results of each individual method. To combine different methods together, a weighting scheme should be developed so that the score of each method contributes in the right proportion and the final score at the end accurately represent the methods involved. Third, to overcome the problem of splitting or integrating original sentence(s) into one or more sentence(s), chunk comparison may be a worthy attempt. The inclusion of neighboring sentences and comparison of these sentences as two chunks should be able to solve this loophole. However, neighboring irrelevant sentences may lower the similarity between two chunks and result in false negative judgment. Application of n-gram to chunk comparison can make the method more robust. New chunks can be formed with in-sequence consecutive sentences. Such formation of chunks was discussed in [1], together with several other approaches.

Meanwhile, the efficiency of WordNet-based methods and Google MI may be improved by constantly increasing the size of the databases where information between two words are stored.

The above are some advices for future work. Hopefully the initial attempt of applying ROUGE with WordNet and any subsequent research will be of any help in the field of plagiarism detection.



Bibliography:

[1] Brin, S., Davis, J., & Garcia-Molina, H. (1995). Copy Detection Mechanisms for Digital Documents. *ACM SIGMOD Record*, vol. 24, no. 2, 398 – 409.

[2] Broder, A. Z., Glassman, S. C., Manasse, M. S., & Zweig, G. (1997). Syntactic Clustering of the Web. *Computer Networks and ISDN Systems*, vol. 29, no. 8-13, 1157 – 1166.

[3] Brown Corpus Manual:

<<http://khnt.hit.uib.no/icame/manuals/brown/INDEX.HTM>>

[4] Buckley, .C, Walz, J., Cardie, C., Mardis, S., Mitra, M., Pierce, D. & Wagstaff, K. (1996). The Smart/Empire TIPSTER IR System. *In Proceedings of a Workshop on held at Baltimore, Maryland* (pp. 107-121). Baltimore, Maryland.

[5] Campbell, D. M., Chen, W. R., & Smith, R. D. (2000). Copy Detection Systems for Digital Documents. *In Advances in Digital Libraries, 2000. ADL 2000. Proceedings. IEEE* (pp. 78-88). Washington, DC, USA.

[6] Chowdhury, A., Frieder, O., Grossman, D. & McCabe, M. C. (2002). Collection Statistics for Fast Duplicate Document Detection. *ACM Transactions on Information Systems*, vol. 20, no. 2, 171 – 191.

[7] Collberg, C., Kobourov, S., Louie, J., & Slattery, T. (2003). SPLaT: A System for Self-Plagiarism Detection. *Proceedings of IADIS International Conference WWW/Internet 2003*, vol. 1, 508 – 514. Algarve, Portugal.

[8] DeVoss, D., & Rosati, A. C. (2002). “It wasn’t me, was it?” Plagiarism and the Web. *Computers and Composition*, 19, 191 – 203.

[9] Dierderich, J. (2006). Computational Methods to Detect Plagiarism in Assessment. *Information Technology Based Higher Education and Training* (pp. 147-154).

[10] Heintze, N. (1996). Scalable Document Fingerprinting. *In Proceedings of the Second USENIX Workshop on Electronic Commerce*. Oakland, California.

[11] Hoad, T. C., & Zobel, J. (2003). Methods for Identifying Versioned and Plagiarized Documents. *Journal of the American Society for Information Science and Technology*, 54(3), 203 – 215.

[12] Iyer, P. & Singh, A. (2005). Document Similarity Analysis for a Plagiarism Detection System. *2nd Inidan International Conference on Artificial Intelligence* (pp. 2534-2544). Pune, India.

[13] Jiang, J. J., & Conrath, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *In Proceedings of Rocling X International Conference 1997 Research on Computational Linguistics*. Taiwan.

[14] Kang, N.-O., Gelbukh, A., & Han, S.-Y. (2006). PPChecker: Plagiarism Pattern Checker in Document Copy Detection. *Lecture Notes in Computer Science*, vol. 4188, 661 – 667.

[15] Kappa Statistics: <<http://www.dmi.columbia.edu/homepages/chuangj/kappa/>>

[16] Kappa Statistics – Table:
<<http://www.dmi.columbia.edu/homepages/chuangj/kappa/>>

[17] Khmelev, D. V. & Teahan, W. J. (2003). A Repetition Based Measure for Verification of Text Collections and for Text Categorization. *In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 104-110). Toronto, Canada.

[18] Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *In Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004* (pp. 74-81). Barcelona, Spain.

[19] LingPipe: <<http://alias-i.com/lingpipe/index.html>>

[20] LingPipe – Sentence Detection:
<<http://alias-i.com/lingpipe/demos/tutorial/sentences/read-me.html>>

[21] Manber, U. (1994). Finding Similar Files in a Large File. *In Proceedings of the USENIX Winter 1994 Technical Conference* (pp. 2-2). San Francisco, California.

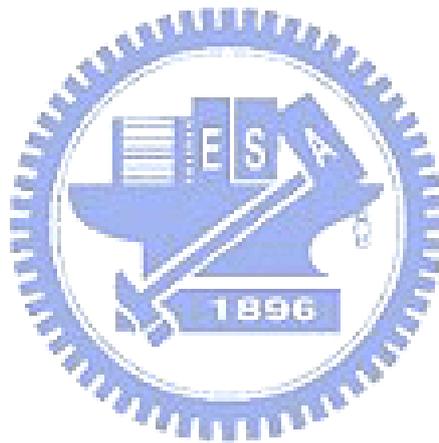
- [22] Maurer, H., Kappe, F., & Zaka, B. (2006). Plagiarism – A Survey. *Journal of Universal Computer Science*, vol. 12, no. 8, 1050 – 1084.
- [23] McGregor, J. H., & Williamson, K. (2005). Appropriate Use of Information at the Secondary School Level: Understanding and Avoiding Plagiarism. *Library & Information Science Research*, 27, 496 – 512.
- [24] Meyer Zu Eissen, S., Stein, B., & Kulig, M. (2007). Plagiarism Detection without Reference Collections. *Advances in Data Analysis*, vol. v, 359 – 366.
- [25] Papineni, K., Roukos, S., Ward, T., & Zhu, W. -J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 311-318). Philadelphia, USA.
- [26] Raveendranathan, P. (2005). Identifying Sets of Related Words from the World Wide Web. *The Faculty of the Graduate School of the University of Minnesota*.
- [27] Shivakumar, N. & Garcia-Molina, H. (1995). SCAM: A Copy Detection Mechanism for Digital Documents. *In Proceedings of the Second International Conference in Theory and Practice of Digital Libraries*. Austin, Texas.
- [28] Si, A., Lenong, H. V., & Lau, R. W. H. (1997). CHECK: A Document Plagiarism Detection System. *In Proceedings of the 1997 AMC Symposium on Applied Computing* (pp. 70-77). San Jose, California.
- [29] Stein, B., & Meyer Zu Eissen, S. (2006). Near Similarity Search and Plagiarism Analysis. *From Data and Information Analysis to Knowledge Engineering*, vol. 10, 430 – 437.
- [30] TERabyte RetrIEveR: <<http://ir.dcs.gla.ac.uk/terrier/>>
- [31] Wikipedia – Brown Corpus: <http://en.wikipedia.org/wiki/Brown_Corpus>
- [32] Wikipedia – Hash Function:
<http://en.wikipedia.org/wiki/Hash_function>
- [33] Wikipedia – Stemming: <<http://en.wikipedia.org/wiki/Stemming>>

[34] Wikipedia – Suffix Tree: <http://en.wikipedia.org/wiki/Suffix_tree>

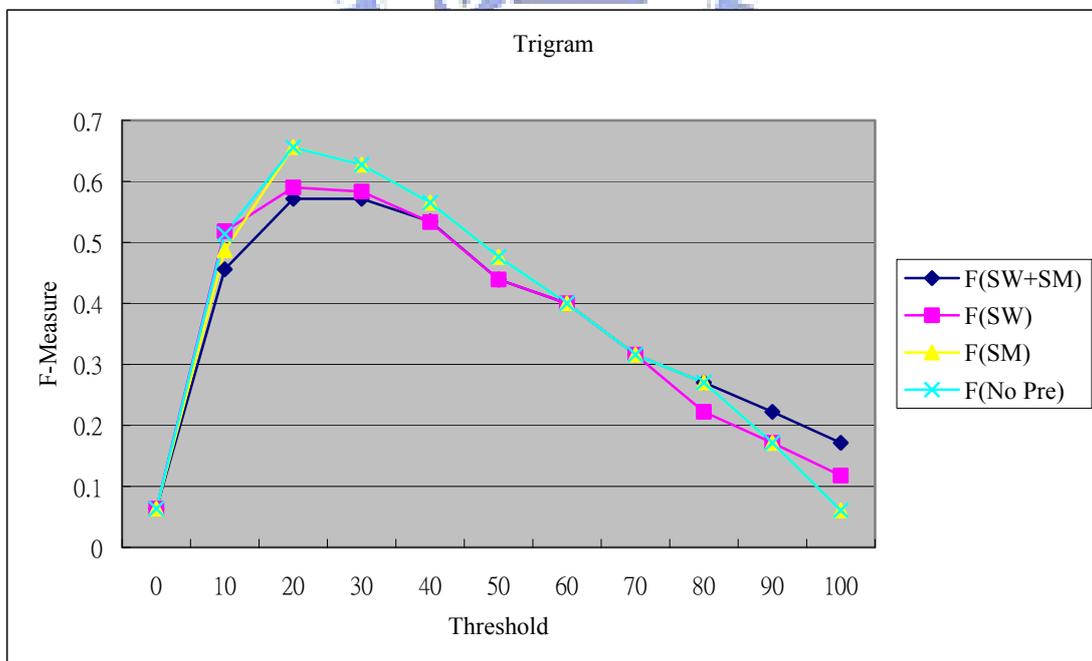
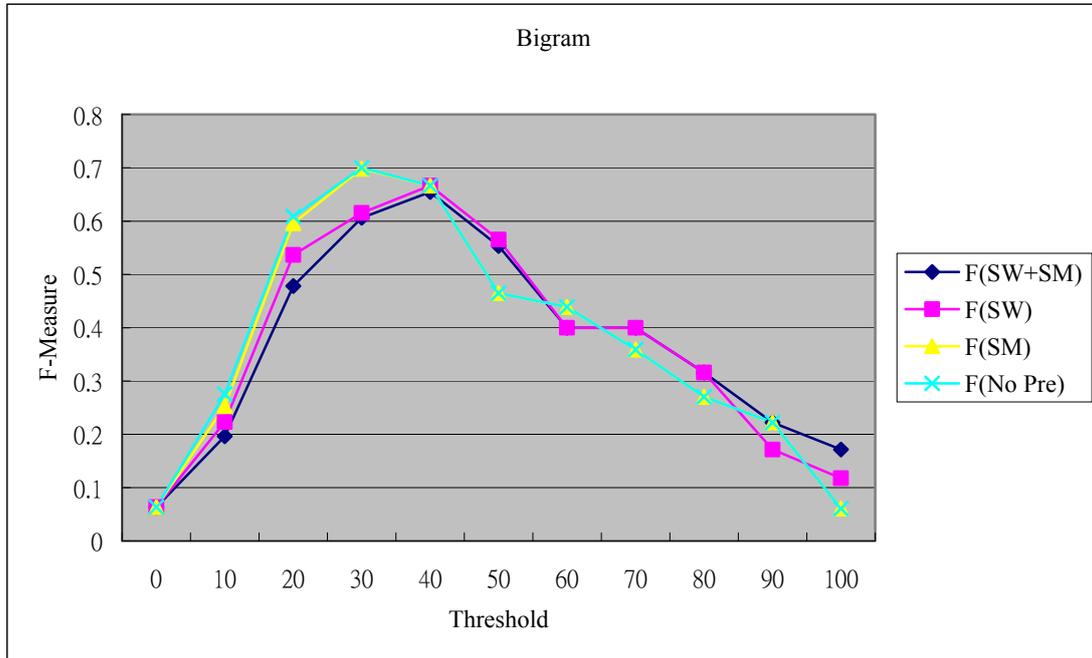
[35] WordNet: <<http://wordnet.princeton.edu/>>

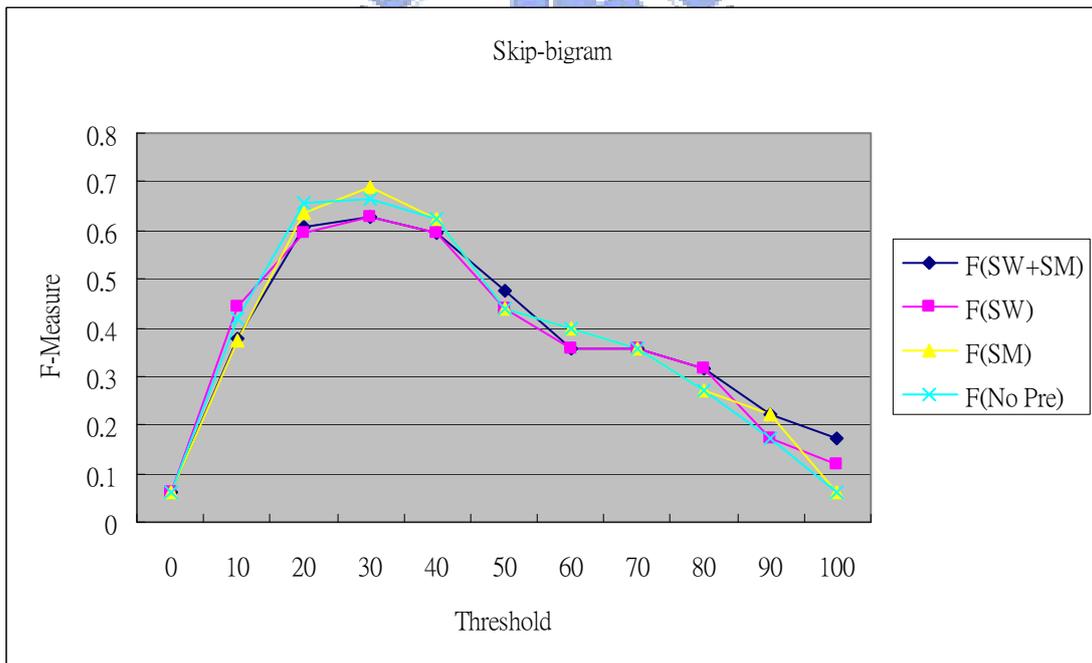
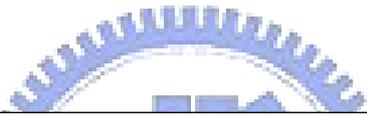
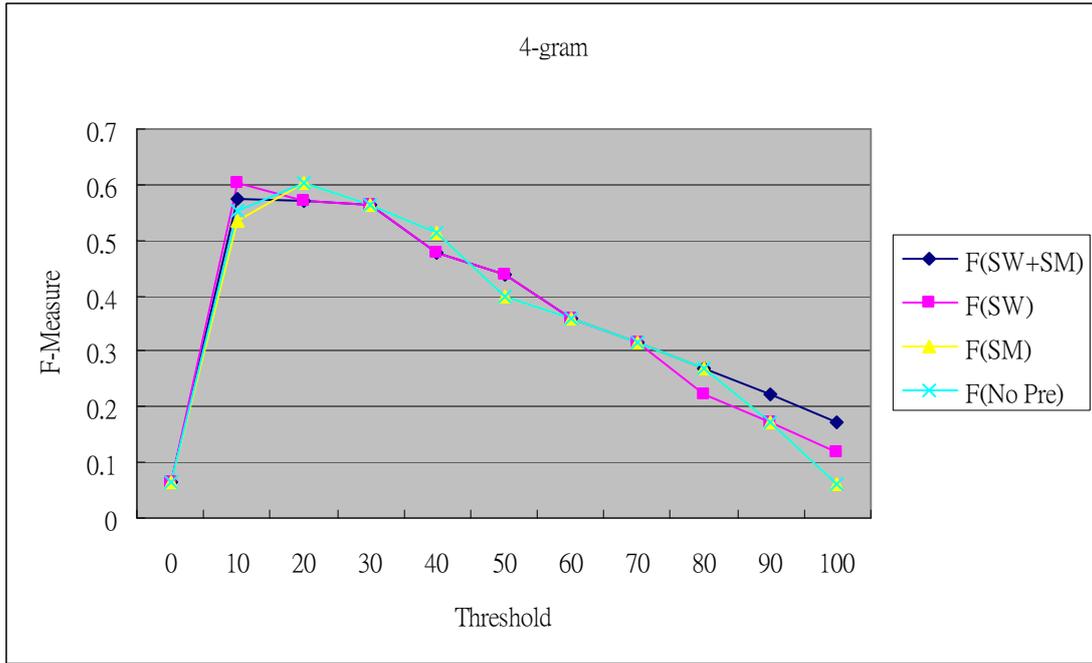
[36] WordNet – Related Projects (Java): <<http://wordnet.princeton.edu/links#Java>>

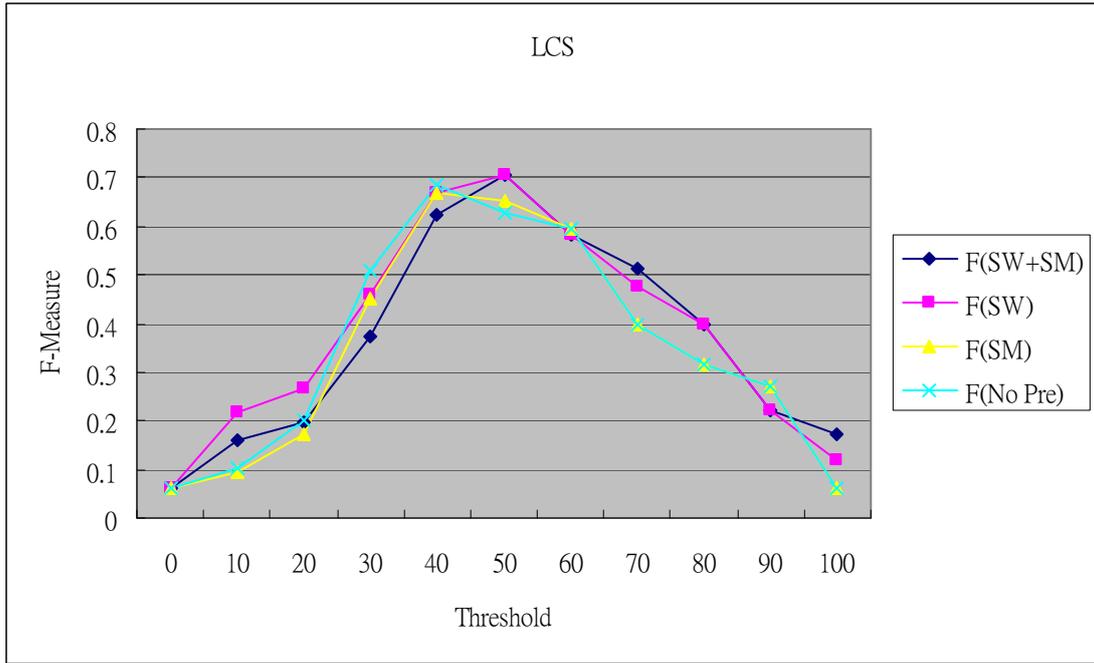
[37] Zaslavsky, A., Bia, A. & Monostori, K. (2001). Using Copy-Detection and Text Comparison Algorithms for Cross-Referencing Multiple Editions of Literary Works. *Lecture Notes in Computer Science*, vol. 2163, 103 – 114.



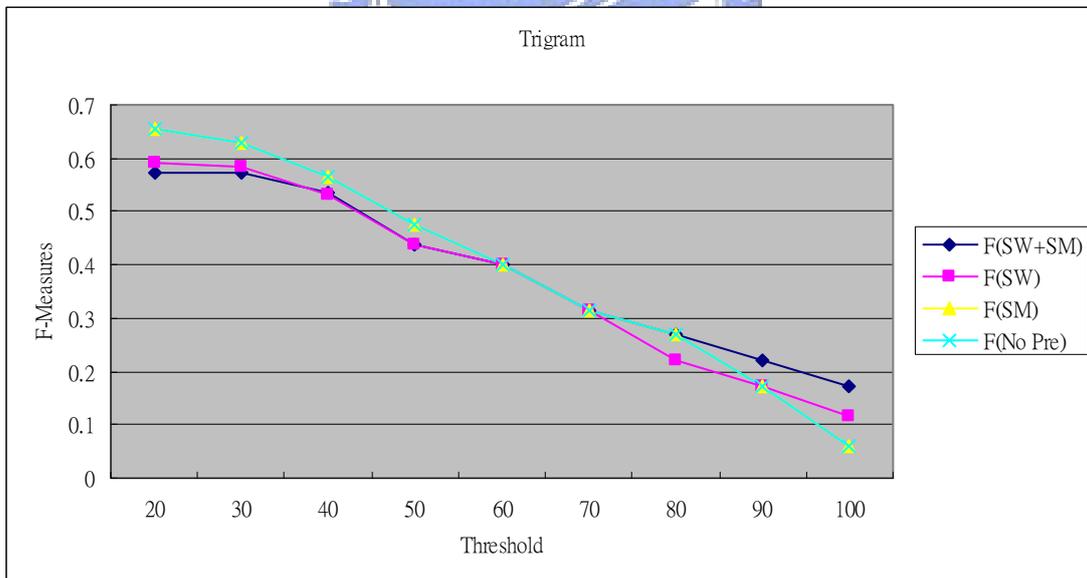
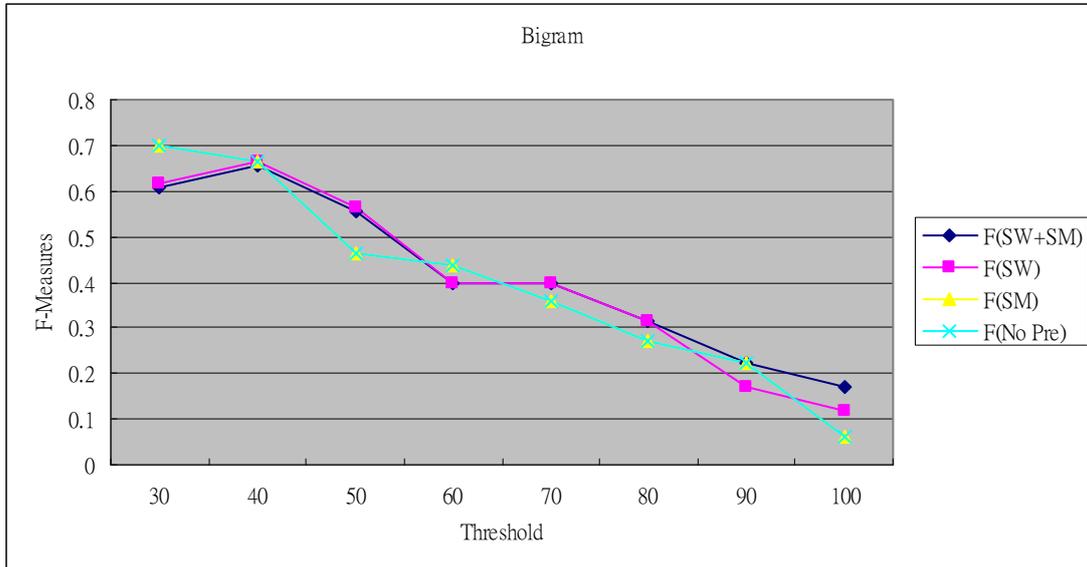
Appendix 1 Line Graphs of Bigram to LCS:

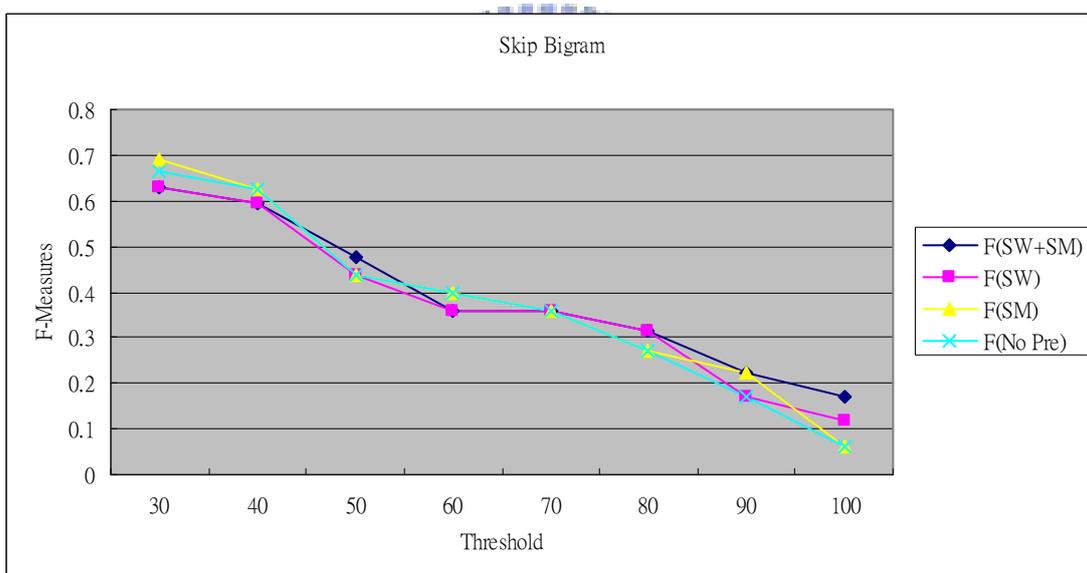
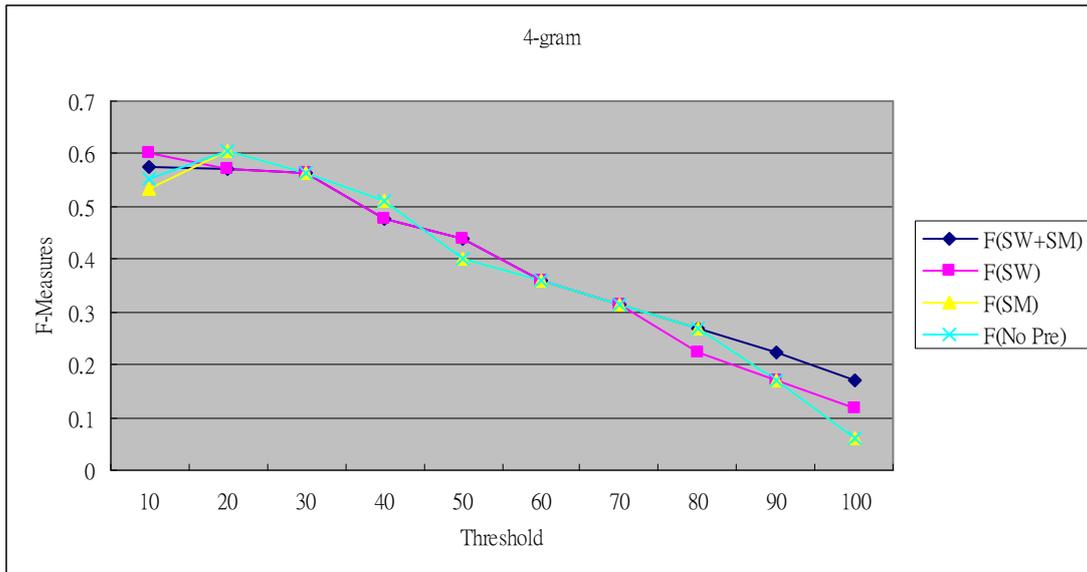


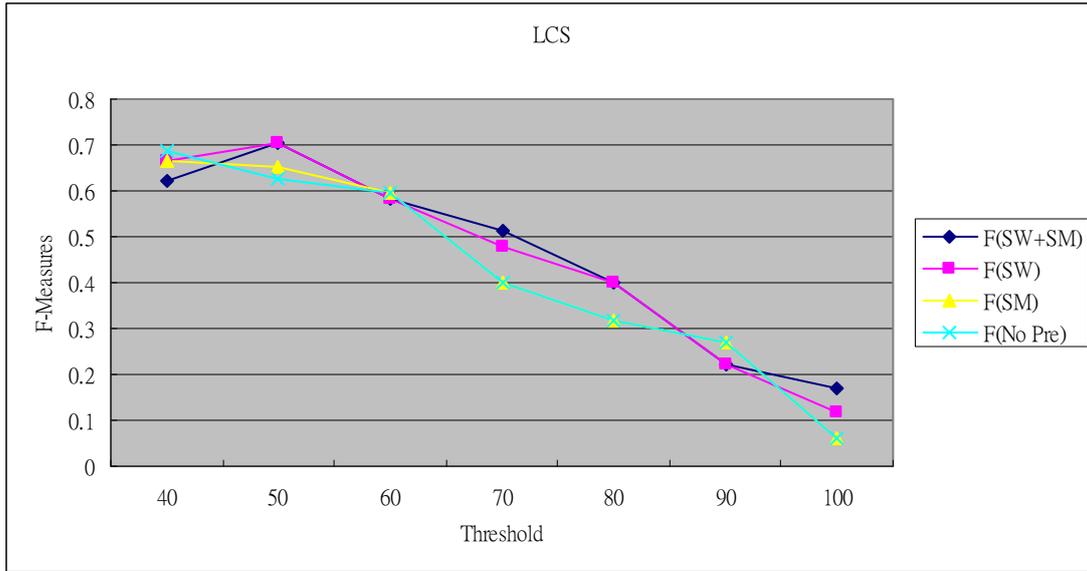




Appendix 2 Partial Line Graphs of Bigram to LCS:







Appendix 3 32 Plagiarism Examples in *Abstract Set*:

Candidate sentence	Reference sentence
this study explores students' understanding of plagiarism and their information use practices	does their understanding match their information use behavior
those who plagiarized least incorporated direct quotations more effectively used fewer quotations and synthesized information and ideas better than did the others	the two students who plagiarized least used minimal quotations see table 1 and used them effectively capably synthesizing their information and ideas a challenge in a task that required primarily reporting of information
the two students who plagiarized least are compared with the two who plagiarized most in an ancient history assignment	findings the practices of the two students who plagiarized most and the two who plagiarized least were dichotomous and therefore provided obvious contrast
the two students who plagiarized least are compared with the two who plagiarized most in an ancient history assignment	for this article four case studies - of the two students who plagiarized most and the two students who plagiarized least - have been developed
the two students who plagiarized least are compared with the two who plagiarized most in an ancient history assignment	understanding of protocols for acknowledging sources as expressed by these students tended to be more vague when compared with those who plagiarized least
in this study we investigated five esl graduate students' awareness of the identities that they constructed through the appropriation of others' words and ideas in their texts	research questions the study addressed the following three research questions a are esl students aware of the textual identities that are constructed in their writings b what are the identities that esl students construct as they appropriate others' words and ideas in their texts

recent research on academic writing has established the intersection of writing and identity	recent research on academic writing has revealed the intersection of writing and identity construction cherry 1988 hatch hill hayes 1993 ivanic 1998
a review of the literature related to unintended outcomes of the use of technology in nursing education and continuing education was conducted to determine the ethical implications for the nursing profession	the purpose of this paper is to provide a review of the literature related to unintended outcomes of the use of technology in nursing education and continuing education in order to determine the ethical implications for the nursing profession
given this correlation between unethical classroom behavior and unethical clinical behavior efforts to staunch academic dishonesty may help allay professional misconduct	implications for nursing education given the correlation between unethical behavior in the classroom and the clinical setting efforts to staunch academic dishonesty may help allay unethical clinical actions
the editorial concludes that a measured degree of vigilance and a greater willingness to pursue any well-founded suspicions of research misconduct are required by editors referees publishers and the wider academic community if the scourge of plagiarism is to be kept at bay	to achieve this a measured degree of vigilance and a greater willingness to pursue any wellfounded suspicions are required on the part of the wider research community as well as from editors referees and publishers
the editorial concludes that a measured degree of vigilance and a greater willingness to pursue any well-founded suspicions of research misconduct are required by editors referees publishers and the wider academic community if the scourge of plagiarism is to be kept at bay	only in this way can the scourge of plagiarism be kept firmly at bay
conversely the summaries of l1 writers contained significantly more moderate and substantial revisions than those of the l2 writers	conversely while most l1 writers used both moderate and substantial revisions most l2 writers did not
conversely the summaries of l1 writers contained significantly more moderate and substantial revisions than those of the l2 writers	moderate and substantial revisions on the other hand were used more frequently by l1 writers than l2 writers

to expand our understanding of university students' paraphrasing strategies the present study analyzed 11 n = 79 and 12 n = 74 writers' use of paraphrase within a summary task and developed a method for classifying these paraphrases into four major paraphrase types near copy minimal revision moderate revision and substantial revision	because no consistent methods for describing different paraphrasing strategies have been employed across studies of textual borrowing a taxonomy of paraphrase types was also developed so that attempted paraphrases could be classified into four linguistically-defined mutually-exclusive categories near copy minimal revision moderate revision and substantial revision
the study then compared the 11 and 12 writers' use of these paraphrase types within their summaries	the study then compared the 11 and 12 writers' use of attempted paraphrases in the summaries by investigating the following research questions 1
conversely the summaries of 11 writers contained significantly more moderate and substantial revisions than those of the 12 writers	as table 4 shows while most near copies were composed by 12 writers most moderate and substantial revisions were composed by 11 writers
it was found that while both groups used about five paraphrases per summary 12 writers used significantly more near copies than 11 writers	12 writers also used significantly more near copies than 11 writers $t = 7.52 p < .01$
this article argues that comparing academic citation and hip-hop sampling can help students become better users of sourcework	therefore prompting students to examine and understand hip-hop sampling can help them become better users of sources in academic papers
secondly turnitin com socializes student writers toward traditional notions of textual normality and docility	advertised as remedial pedagogy the turnitin com service socializes student writers toward traditional normality and docility notions
and third turnitin com represents a new phase in the bureaucratization of composition instruction consistent with past administrative practices and reflective of emerging corporate management alliances in higher education	moreover as a corporate solution to a nagging pedagogical problem the turnitin com phenomenon represents what i see as a continuing bureaucratization of writing and writing instruction consistent with past administrative practices and reflective of emerging corporate management alliances in higher education

