UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

RAFAEL COREZOLA PEREIRA

Cross-Language Plagiarism Detection

Dissertation presented in partial fulfillment of the requirements for the degree of Master of Computer Science

Prof. Dr[a]. Viviane Pereira Moreia
Advisor

Prof. Dr[a]. Renata Galante
Coadvisor

Porto Alegre, August 2010.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Pereira, Rafael Corezola

Cross-Language Plagiarism Detection / Rafael Corezola Pereira – Porto Alegre: Programa de Pós-Graduação em Computação, 2010.

64 f.:il.

Dissertation (mastership) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2010. Advisor: Viviane Pereira Moreira; Coadvisor: Renata Galante.

1.Plagiarism. 2.Cross-Language Plagiarism Detection. 3. Plagiarism Test Collections. I. Moreira, Viviane Pereira. II. Galante, Renata. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Rui Vicente Oppermann
Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion
Diretor do Instituto de Informática: Prof. Flávio Rech Wagner
Coordenador do PPGC: Prof. Álvaro Freitas Moreira
Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

CL-ESA	Cross-Language Explicit Semantic Analysis
CLIR	Cross-Language Information Retrieval
CLPA	Cross-Language Plagiarism Analysis
ECLaPA	Europarl Cross-Language Plagiarism Analysis
IDF	Inverse Document Frequency
IR	Information Retrieval
TF-IDF	Term Frequency – Inverse Document Frequency
TREC	Text Retrieval Conference
WEKA	Waikato Environment for Knowledge Analysis
XML	Extensible Markup Language

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

ABSTRACT

Plagiarism is one of the most serious forms of academic misconduct. It is defined as “the use of another person's written work without acknowledging the source”. As a countermeasure to this problem, there are several methods that attempt to automatically detect plagiarism between documents. In this context, this work proposes a new method for Cross-Language Plagiarism Analysis. The method aims at detecting external plagiarism cases, i.e., it tries to detect the plagiarized passages in the suspicious documents (the documents to be investigated) and their corresponding text fragments in the source documents (the original documents). To accomplish this task, we propose a plagiarism detection method composed by five main phases: language normalization, retrieval of candidate documents, classifier training, plagiarism analysis, and post-processing. Since the method is designed to detect cross-language plagiarism, we used a language guesser to identify the language of the documents and an automatic translation tool to translate all the documents in the collection into a common language (so they can be analyzed in a uniform way). After language normalization, we applied a classification algorithm in order to build a model that is able to differentiate a plagiarized text passage from a non-plagiarized one. Once the classifier is trained, the suspicious documents can be analyzed. An information retrieval system is used to retrieve, based on passages extracted from each suspicious document, the passages from the original documents that are more likely to be the source of plagiarism. Only after the candidate passages are retrieved, the plagiarism analysis is performed. Finally, a post-processing technique is applied in the reported results in order to join the contiguous plagiarized passages. We evaluated our method using three freely available test collections. Two of them were created for the PAN competitions (PAN’09 and PAN’10), which are international competitions on plagiarism detection. Since only a small percentage of these two collections contained cross-language plagiarism cases, we also created an artificial test collection especially designed to contain this kind of offense. We named the test collection ECLaPA (Europarl Cross-Language Plagiarism Analysis). The results achieved while analyzing these collections showed that the proposed method is a viable approach to the task of cross-language plagiarism analysis.

Keywords: Plagiarism, cross-language plagiarism detection, plagiarism test collections.

Detecção de Plágio Multilíngue

RESUMO

Plágio é um dos delitos mais graves no meio acadêmico. É definido como “o uso do trabalho de uma pessoa sem a devida referência ao trabalho original”. Em contrapartida a esse problema, existem diversos métodos que tentam detectar automaticamente plágio entre documentos. Nesse contexto, esse trabalho propõe um novo método para Análise de Plágio Multilíngue. O objetivo do método é detectar casos de plágio em documentos suspeitos baseado em uma coleção de documentos ditos originais. Para realizar essa tarefa, é proposto um método de detecção de plágio composto por cinco fases principais: normalização do idioma, recuperação dos documentos candidatos, treinamento do classificador, análise de plágio, pós-processamento. Uma vez que o método é projetado para detectar plágio entre documentos escritos em idiomas diferentes, nós usamos um *language guesser* para identificar o idioma de cada documento e um tradutor automático para traduzir todos os documentos para um idioma comum (para que eles possam ser analisados de uma mesma forma). Após a normalização, nós aplicamos um algoritmo de classificação com o objetivo de construir um modelo que consiga diferenciar entre um trecho plagiado e um trecho não plagiado. Após a fase de treinamento, os documentos suspeitos podem ser analisados. Um sistema de recuperação é usado para buscar, baseado em trechos extraídos de cada documento suspeito, os trechos dos documentos originais que são mais propensos de terem sido utilizados como fonte de plágio. Somente após os trechos candidatos terem sido retornados, a análise de plágio é realizada. Por fim, uma técnica de pós-processamento é aplicada nos resultados da detecção a fim de juntar os trechos plagiados que estão próximos um dos outros. Nós avaliamos o métodos utilizando três coleções de testes disponíveis. Duas delas foram criadas para as competições PAN (PAN’09 e PAN’10), que são competições internacionais de detecção de plágio. Como apenas um pequeno percentual dos casos de plágio dessas coleções era multilíngue, nós criamos uma coleção com casos de plágio multilíngue artificiais. Essa coleção foi chamada de ECLaPA (Europarl Cross-Language Plagiarism Analysis). Os resultados alcançados ao analisar as três coleções de testes mostraram que o método proposto é uma alternativa viável para a tarefa de detecção de plágio multilíngue.

Palavras-chave: Plágio, detecção de plágio multilíngue, coleções de teste de plágio.

1 INTRODUCTION

1.1 Motivation and Goals

Plagiarism is one of the most serious forms of academic misconduct. It is defined as "the use of another person's written work without acknowledging the source". According to (MAURER; KAPPE; ZAKA, 2006), there are several types of plagiarism. It can range from simply copying another's work word-for-word to paraphrasing the text in order to disguise the offense.

A study by (MCCABE, 2005) with over 80,000 students in the US and Canada found that 36% of undergraduate students and 24% of graduate students admit to have copied or paraphrased sentences from the Internet without referencing them. Amongst the several methods for plagiarism commonly in practice, (MAURER; KAPPE; ZAKA, 2006) mention cross-language content translation. The authors also surveyed plagiarism detection systems and found that none of the available tools support search for cross-language plagiarism. The increasing availability of textual content in many languages, and the evolution of automatic translation can potentially make this type of plagiarism more common. Cross-language plagiarism can be seen as an advanced form of paraphrasing since every single word might have been replaced by a synonym (in the other language). Furthermore, word order might have changed. These facts make cross-language plagiarism harder to detect.

Cross-language plagiarism, as acknowledged by (ROIG, 2010), can also involve self-plagiarism, i.e., the act of translating self published work without referencing the original. This offense usually aims at increasing the number of publications. As stated by (LATHROP; FOSS, 2000), another common scenario of cross-language plagiarism happens when a student downloads a paper, translates it using an automatic translation tool, corrects some translation errors and presents it as their own work.

The aim of this work is to propose and evaluate a new method for *Cross-Language Plagiarism Analysis* (CLPA). The main difference of our method when compared to the existing ones is that we applied a classification algorithm to build a model that can distinguish between a plagiarized and a non-plagiarized text passage. Note that there are two different areas of plagiarism analysis. One area, known as external plagiarism analysis, uses a reference collection to find the plagiarized passages. The other area, known as intrinsic plagiarism analysis, tries to detect plagiarism without a reference collection, usually by considering differences in the writing style of the suspicious document (MALYUTOV, 2006, STEIN; EISSEN, 2007). In this work, we focus on external plagiarism analysis. Thus, our task is to detect the plagiarized passages in the suspicious documents (i.e., the documents to be investigated) and their corresponding

text fragments in the source documents (i.e., in the reference collection) even if the documents are in different languages.

1.2 Overview

The proposed method is divided into five main phases: language normalization, retrieval of candidate documents, classifier training, plagiarism analysis, and result post-processing. Since our method aims at detecting plagiarism between documents written in different languages, we used an automatic translation tool to translate the suspicious and the source documents into a single common language in order to analyze them in a uniform way. After the normalization phase, we used a classification algorithm to build a model to enable the method to learn how to distinguish between a plagiarized and a non-plagiarized text passage. To accomplish this task, we selected a pre-defined set of features to be considered during the training phase of the method.

Once the classification model is built, we used an Information Retrieval (IR) system to retrieve, based on the text passages extracted from the suspicious documents, the documents that are more likely to be the source of plagiarism offenses. The idea behind the retrieval phase is that it would not be feasible to perform a detailed analysis between the suspicious document and the entire reference collection. Only after retrieving a small subset of the reference collection, plagiarism analysis is performed. Finally, the detection results are post-processed to join contiguous plagiarized passages.

Since this is a new area of research, the experiments reported in the literature were done over small test collections which most of the times were assembled by the authors. With the goal of resolving this problem, in 2009, the 1st International Competition on Plagiarism Detection (PAN-2009, 2009) took place (the PAN'09 competition). The aim was to provide a common basis for the evaluation of plagiarism detection systems. Thus, in order to validate the proposed method, we assessed its performance while detecting the plagiarism cases in the test collections of the PAN competition. However, as there are only a few studies in the area of cross-language plagiarism analysis, the corpus created for the competition contained only a small percentage of cross-language plagiarism cases. Besides, none of the participating groups tried to detect this type of plagiarism offense during the competition. Therefore, in the absence of a corpus especially designed to evaluate cross-language plagiarism methods, we created an artificial plagiarism corpus called ECLaPA (Europarl Cross-Language Plagiarism Analysis). The corpus is based on the Europarl Parallel Corpus (KOEHN, 2005), which is a collection of documents generated from the proceedings of the European Parliament. We conducted two different experiments with this corpus; the first one considers only monolingual plagiarism cases, while the second one considers only cross-language plagiarism cases. The results showed that the cross-language experiment achieved 86% of the performance of the monolingual baseline.

1.3 Contributions

In summary, the main contributions of this work are:

- Definition of a new CLPA method as well as its evaluation against freely available plagiarism corpora.

- The employment of a classification algorithm to build a model that is able to distinguish between a plagiarized and a non-plagiarized text passage (we do not know of any experiments applying classification algorithms to the external plagiarism analysis task).

- Creation of a plagiarism test collection especially designed to contain cross-language plagiarism cases, which provides a common basis of comparison for cross-language plagiarism methods.

1.4 Organization of the text

This dissertation is divided into 6 chapters. In this chapter, we presented the motivations that led us to propose the method described throughout this dissertation. We also presented the goals as well as an overview of the proposed method. Finally, we enumerated the main contributions of this work. The remainder of the text is organized as follows:

- Chapter 2 presents an overview of the three main areas of research that are related to this work: monolingual plagiarism detection, cross-language information retrieval, and cross-language plagiarism analysis.

- Chapter 3 describes in detail how the proposed method works, i.e., it explains each one of its five phases: language normalization, retrieval of candidate documents, feature selection and classifier training, plagiarism analysis, and result post-processing.

- Chapter 4 presents the artificial cross-language plagiarism test collection created in order to evaluate the proposed method.

- Chapter 5 describes the resources used during the experiments. It also presents the evaluation metrics employed as well as the results achieved during the evaluation of the method.

- Chapter 6 summarizes our main contributions, presents a list of published papers, and shows our final conclusions as well as a discussion of future work.

2 RELATED WORK

This chapter presents a literature review divided into the three areas that are closely related to this dissertation. First, we discuss monolingual plagiarism detection, then cross-language information retrieval, and finally, cross-language plagiarism analysis, which is the focus of this work.

2.1 Monolingual Plagiarism Detection

Research on document processing has recently devoted more attention to the problem of detecting plagiarism. The standard method for monolingual plagiarism analysis involves comparing chunks from suspicious and source documents. The most popular approach, according to (STEIN; EISSEN, 2006), is to use the MD5 hashing algorithm (RIVEST, 1992) to calculate a hash signature (called fingerprint) for each chunk. Identical chunks will have the same fingerprint. Note that since plagiarized texts are not likely to be identical to its source, a minor change in the plagiarized text will avoid its detection. Although very simple, this approach has several drawbacks (it is computationally expensive and requires large storage capacity). To overcome these problems, the authors proposed a new hashing technique called fuzzy fingerprints (STEIN; EISSEN, 2006). The idea behind fuzzy fingerprints is to generate the same hash signature for lexically similar chunks. This enables the usage of larger chunks, which tends to decrease both retrieval time and the required disk space.

The work by (BARRÓN-CEDEÑO; ROSSO, 2009) proposes the division of the suspicious documents into sentences, which are then split into word n-grams. The source documents are also split into word n-grams. Then, an exhaustive comparison is performed between the n-grams of each suspicious sentence and the n-grams of each source document. To decide whether the suspicious sentence is plagiarized, the authors applied the containment measure to compare the corresponding sets. The experiments showed that the best results are achieved when using bi-grams (better recall) and tri-grams (better precision). In (BARRÓN-CEDEÑO; ROSSO; BENEDÍ, 2009), the authors applied the Kullback-Leibler distance to reduce the number of documents that must be compared against the suspicious document. The main difference to our approach is that they build feature vectors for each reference document and compare these vectors against the vector of the suspicious document. The top ten reference documents with the lowest distance with respect to the vector of the suspicious document are selected to the plagiarism analysis phase.

The method proposed in (GROZEA; GEHL; POPESCU, 2009), winner of the PAN`09 competition (PAN-2009, 2009), computes a matrix of string kernel values in order to find the similarity between suspicious and source documents. An exhaustive pairwise comparison is necessary between each source and each suspicious document.

After that, for each source document, the suspicious documents are ranked in decreasing order of similarity and only the first 51 one are kept for further investigation. Finally, to identify the text passages that were plagiarized, a pairwise sequence matching technique, called encoplot, is used.

In (KASPRZAK; BRANDEJS; KŘIPAČ, 2009), the authors used overlapping sequences of five words to create an inverted index that maps the 5-word chunk hash value to the list of source documents in which the chunk appears. Once the inverted index is created, each suspicious document is split using the same strategy and the hash values of its 5-word chunks is looked up in the inverted index. Documents that shared more than 20 chunks were considered similar. After all the common chunks are identified, a merging algorithm is applied to combine the chunks that appear near each other in the suspicious and in the source document. This method achieved the second highest score in the PAN`09 competition.

It is important to notice that methods for monolingual plagiarism detection cannot be directly applied to CLPA because the terms in the suspicious and source text segments will not match. Even if the plagiarized text is an exact translation of the original, word order will change.

2.2 Cross-Language Information Retrieval

CLPA is related to Cross-Language Information Retrieval (CLIR), in which a query in one natural language is matched against documents in another language. The main problem of CLIR is knowing how to map concepts between languages (GREFENSTETTE, 1998), whereas the problem of CLPA is more difficult as it is necessary to match a segment of text in one language to a segment of text of equal content in another language. The sizes of these segments can vary from one sentence to hundreds of pages (i.e., whole books).

There are three traditional approaches for CLIR which are used to bring the query into the language of the documents: (*i*) machine translation (MT) systems, where the query is automatically translated to a specified language (FUJII; ISHIKAWA, 2004); (*ii*) multilingual thesauri or dictionaries, where the query terms are replaced by the terms found in the thesaurus or dictionary (GEY; JIANG, 1999); and (*iii*) throughout the analysis of parallel or comparable corpora, where term equivalences are automatically extracted (ORENGO; HUYCK, 2002). CLIR approaches grow in and out of preference throughout the years. While machine readable dictionaries were popular in the late 90's (HULL; GREFENSTETTE, 1996), recently MT-based systems are the most employed strategy. For the evaluation campaign CLEF 2009 (PETERS; FERRO, 2009), seven out of ten bilingual approaches used MT systems to bring the queries and the documents into the same language.

2.3 Cross-Language Plagiarism Analysis

Interest on CLPA is recent and it is growing quickly. So far, only a few studies have dealt with CLPA. The work by (BARRÓN-CEDEÑO, et al., 2008) relies on a statistical bilingual dictionary created from parallel corpora and on an algorithm for bilingual text alignment. The authors report experiments on a collection composed of 5 original fragments which were used to generate plagiarized versions. The results of the experiments showed that the similarity between the original documents and their

plagiarized versions was much higher than the similarity between non-plagiarized documents. However, experiments with larger collections must be conducted in order to check the real efficiency of the method.

MLPlag (CESKA; TOMAN; JEZEK, 2008) is a CLPA method based on the analysis of word positions. EuroWordNet is used to transform words into a language independent representation. The authors built two multilingual corpora: JRC-EU and Fairy-tale. The first corpus is composed of 400 randomly selected European Union legislative texts containing 200 reports written in English and the same number of corresponding reports written in Czech. The second corpus represents a smaller set of text documents with a simplified language. This corpus is composed of 54 documents, 27 English and 27 corresponding translations in Czech. The method showed good results. However, the authors stated that the incompleteness of the EuroWordNet may lead to difficulties during cross-language plagiarism detection, especially when handling less common languages.

Other studies propose multilingual retrieval approaches that can help detect document plagiarism across languages. The work by (POULIQUEN; STEINBERGER; IGNAT, 2003) proposes a system that identifies translations and very similar documents among a large number of candidates. The contents of the documents are represented as vectors of terms from a multilingual thesaurus. The similarity measure for documents is the same, independent from the document language. The authors report experiments that search for Spanish and French translations of English documents, using several parallel corpora ranging from 795 to 1130 text pairs and searching in a search space of up to 1640 documents. The result of the experiments showed that the system can detect translations with over 96% precision.

The work by (POTTHAST, 2007) introduces a new multilingual retrieval model called Cross-Language Explicit Semantic Analysis (CL-ESA) for the analysis of cross-language similarity. The authors report experiments on a multilingual parallel corpus (JRC-Acquis) and a multilingual comparable corpus (Wikipedia). Recently, in (POTTHAST, et al., 2010a) the authors compare CL-ESA to other methods and report that character n-grams achieves a better performance. However, character n-grams will not be suitable for languages with unrelated syntax.

3 A METHOD FOR CROSS-LANGUAGE PLAGIARISM ANALYSIS

Given a reference corpus ***D*** of original documents and a corpus ***D'*** of suspicious documents, our proposed method aims at detecting all passages ***s'∈ D'*** which have been plagiarized from a passage ***s ∈ D***. To accomplish this task, a classification model is built, based on a training collection ***D''***, in order to let a classifier decides whether a suspicious passage ***s'*** is plagiarized or not from a passage ***s***. Note that both the original and the suspicious documents can be written in any given language.

As mentioned before, monolingual plagiarism analysis methods cannot be directly applied to CLPA. Thus, the method proposed here tries to overcome this problem by using an automatic translation tool to have both suspicious and source documents in the same language. Only after the translation process is done, the plagiarism analysis is performed. It is important to notice that even if we used an excellent translation tool, we will probably have some content loss during this phase.

In order to accomplish the task defined above we propose a method divided into five main phases, which are briefly described below:

(1) *Language Normalization*: at this phase, both the suspicious (***D'***) and the source documents (***D***) are translated into a common language.

(2) *Retrieval of Candidate Documents*: at this phase, passages extracted from the suspicious documents (***D'***) are used to find out which of the source documents (***D***) are more likely to be the source of plagiarism offenses. This is a very important phase since it would not be feasible to perform a detailed analysis between the suspicious documents and the entire reference collection.

(3) *Feature Selection and Classifier Training*: at this phase, using the training collection ***D''*** (composed of suspicious and source documents), a pre-defined set of features is selected in order to build the classification model. Based on the classifier built the method is able to decide whether a suspicious passage ***s'*** is plagiarized or not.

(4) *Plagiarism Analysis*: at this phase, each passage ***s'*** extracted from the suspicious documents is compared against its respective set of candidate documents in order to evaluate whether the suspicious passage is, in fact, plagiarized.

(5) *Result Post-Processing*: at this phase, we join contiguous plagiarized passages into a single one in order to report a plagiarism offense as a whole instead of several small plagiarized passages.

These five phases as well as their inputs and outputs are depicted in Figure 3.1 and explained in more detail in the next sections (the phases are numbered from one to five).

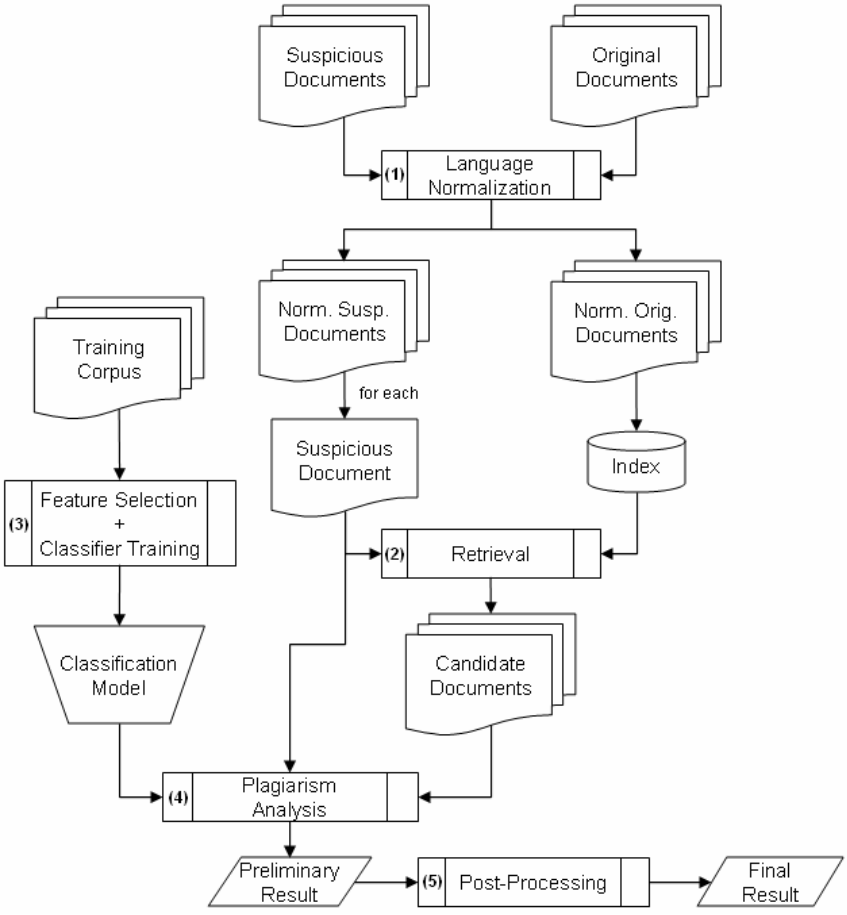

Figure 3.1: The five main phases of the proposed method.

3.1 Language Normalization

Since the method is designed to handle documents written in several different languages, at this phase, we have to translate the documents of ***D*** and ***D'*** into a single common language in order to analyze them in a uniform way. We chose English as the default language, since it is the most commonly used language on the Internet[1] and translation resources to and from English are more easily found (e.g., it is easier to find a translator from Finnish into English than from Finnish into Portuguese). Furthermore, according to (KOEHN, 2005), English is amongst the easiest languages to translate into since it has few inflectional forms.

In order to translate the non-English documents in the collection into English we use an automatic translation tool. However, before translating the documents, we first have to identify the language in which each document in the collection was written. To accomplish this task, we use a language guesser.

Since there is no perfect language guesser, we implemented the method below to define the language of each document:

[1] http://www.internetworldstats.com/stats7.htm

(1) Extract a randomly chosen chunk of approximately 1000 contiguous characters from the document;

(2) Submit the extracted chunk to the language guesser;

(3) Store the given language;

(4) Repeat the steps (1) to (3) ten times;

(5) If the guesser gives as output the same language for seven times or more, assume that the document was written in that language. Otherwise, assume that the document was written in an unknown language;

(6) If the selected language is English (the default language) or an unknown language, we do not mark the document for translation;

Note that instead of simply assuming that the language given as output by the guesser was the right one, we decided to apply the heuristic above. The reason we did that is because we encountered some problems with the language guesser. Thus, with the method described above, we mitigate two problems: (*i*) documents may have small passages written in a language other than its main language; (*ii*) documents may have passages containing only numbers (e.g., tables).

Once we detect the language of all the documents in the collection, we can proceed to the translation of the non-English documents. It is important to notice that the documents whose language could not be detected are not selected for translation. Figure 3.2 illustrates the steps necessary to obtain the normalized collection.

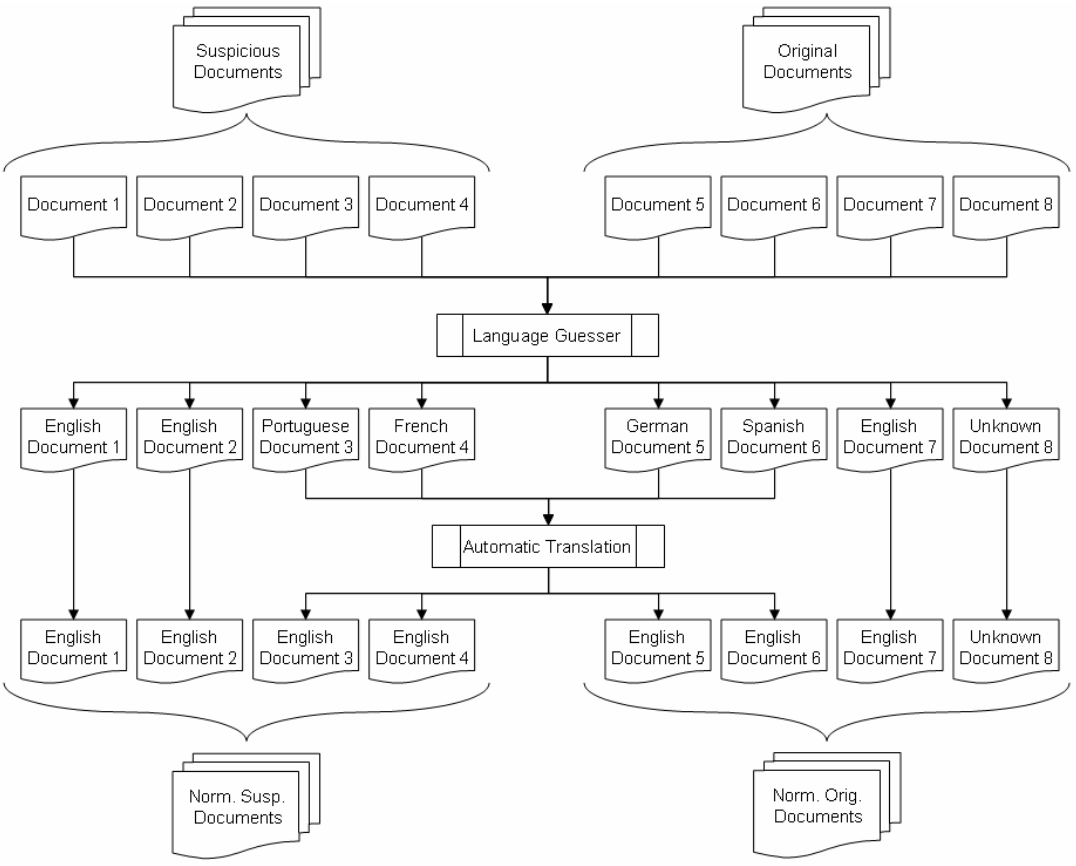

Figure 3.2: Language normalization phase.

After the documents are normalized, (i.e., translated to English), we have for each non-English document its respective parallel English document.

3.2 Retrieval of Candidate Documents

The main goal of this phase is to select, based on each suspicious document of ***D'***, the documents in the reference collection ***D*** that are more likely to be the source of plagiarism offenses (these selected documents, hereafter, are called candidate documents). Therefore, after the candidates documents are retrieved, only a very small part of the corpus needs to be analyzed. Note that this is one of the most important phases of the proposed method since it would not be feasible to perform a plagiarism analysis between the suspicious document and the entire reference collection. As a result, to find out the candidate documents among all the documents in the reference collection, we use an Information Retrieval system.

To reduce the amount of text that must be analyzed during the plagiarism analysis phase we divide the source documents into several subdocuments, each one composed of a single passage of the source document (as depicted in Figure 3.3). The rationale behind this is that after the documents are retrieved, it is not necessary to have a detailed analysis against the entire contents of the candidate source document, only against the content of the subdocuments retrieved.

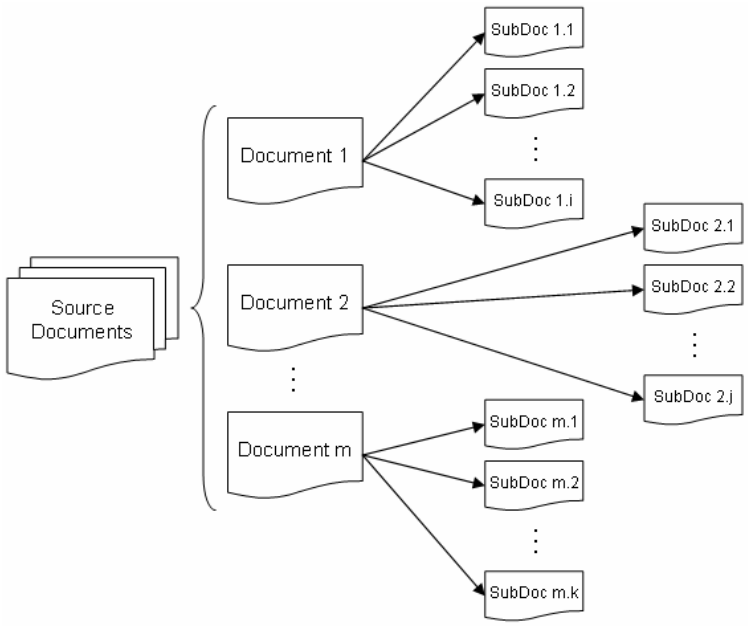

Figure 3.3: Source documents division.

The unit used to split the documents is the paragraph. Thus, the number of subdocuments each document generates is directly related to the number of paragraphs the document has. It is important to mention here that we must split the documents in their original language in order to keep the real offset and length of the passages. Thus, as shown in Figure 3.4, only after they are split we can get the English translations from the normalized source documents. Note that only the subdocuments that are not already written in the default language need to pass through this process.

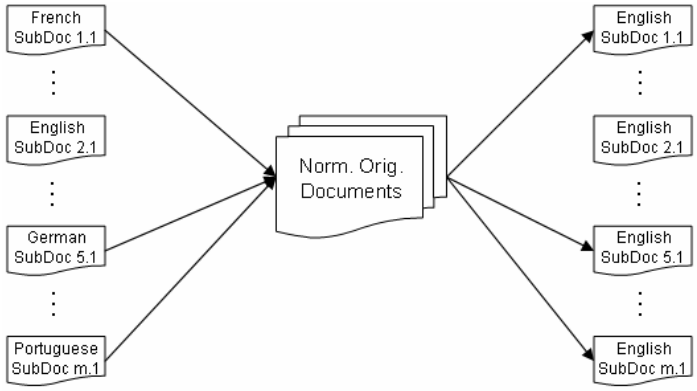

Figure 3.4: Translating the subdocuments.

After splitting all the source documents and translating the subdocuments, the reference collection can be indexed by the IR system. An example of an original document before and after being split is shown in Figure 3.5.

Original document
"So it was in old times," said Djemboulát, with a smile, "when our old men trusted more to prayer, and God oftener listened to them; but now, my friends, there is a better hope—your valour! Our omens are in the scabbards of our shoóshkas, (sabres,) and we must show that we are not ashamed of them. Harkye, Ammalát," he continued, twisting his mustache, "I will not conceal from you that the affair may be warm. I have just heard that Colonel K---- has collected his division; but where he is, or how many troops he has, nobody knows." "Glory is a good bird, when she lays a golden egg; but he that returns with his toróks (straps behind the saddle) empty, is ashamed to appear before his wife. Winter is near, and we must provide our households at the expense of the Russians, that we may feast our friends and allies. Choose your station, Ammalát Bek. Do you prefer to advance in front to carry off the flocks, or will you remain with me in the rear? I and the Abréks will march at a foot's pace to restrain the pursuers."
Document after being split
<DOC><DOCNO>source-document00003.txt:9156:537</DOCNO> "So it was in old times," said Djemboulát, with a smile, "when our old men trusted more to prayer, and God oftener listened to them; but now, my friends, there is a better hope—your valour! Our omens are in the scabbards of our shoóshkas, (sabres,) and we must show that we are not ashamed of them. Harkye, Ammalát," he continued, twisting his mustache, "I will not conceal from you that the affair may be warm. I have just heard that Colonel K---- has collected his division; but where he is, or how many troops he has, nobody knows."</DOC> <DOC><DOCNO>source-document00003.txt:9888:489</DOCNO> "Glory is a good bird, when she lays a golden egg; but he that returns with his toróks (straps behind the saddle) empty, is ashamed to appear before his wife. Winter is near, and we must provide our households at the expense of the Russians, that we may feast our friends and allies. Choose your station, Ammalát Bek. Do you prefer to advance in front to carry off the flocks, or will you remain with me in the rear? I and the Abréks will march at a foot's pace to restrain the pursuers."</DOC>

Figure 3.5: Example of a document before and after being split.

Note that we adopted the TREC file format, which is the format used in the Text Retrieval Conference (TREC, 2007), to store the documents after they have been divided. The reason for adopting this format is that most search engine tools available can handle this type of file. Therefore, each passage of the original document becomes a TREC document itself, which is referred to as a subdocument of the original document. Each TREC document created is composed by two types of information:

- *The document identifier (DOCNO)*: the identifier will be used by the IR system to identify the document (e.g., when building the rank in response to a query). It is formed by the original document number, and the offset and length (in characters) of the passage in the original document. For instance, looking at the first document of Figure 3.5 we have, respectively, "source-document00003.txt" as the original document name, "9156" as the passage offset, and "537" as the passage length;
- *The document content (DOC)*: the content will be used to build the index and to be compared against the suspicious documents;

As mentioned before, after each document of the source collection is divided and translated to the default language, the collection can be indexed. It is important to mention that during the indexing process the unit used is the word. The reason word n-grams are not used is that the word order may change after the normalization process (since the text that is being indexed may be originated from any language). During the indexing process, we use two IR techniques: stopword removal and stemming. These techniques are explained below:

- *Stopword Removal*: this technique aims at discarding words that do not carry significant meaning. Usually these words are very frequent in the documents, therefore, they have low discriminating power. Examples of words that are considered stopwords are: articles (*the*, *a*, *an*), prepositions (*at*, *by*, *as*), conjunctions (*and*, *or*, *both*), etc. The stopword removal technique can reduce the index size considerably (~40%). Figure 3.6 shows a passage before and after this technique is applied.

Original text passage
The G8 and G20 summits in Toronto are expected to top $1 billion in costs. But former White House aides say that even with the hefty price, the meetings more than pay for themselves in both tangible and intangible ways.
Text passage after applying stopword removal
g8 g20 summits toronto expected top $1 billion costs. white house aides say hefty price, meetings pay tangible intangible ways.

Figure 3.6: Text passage after applying stopword removal.

- *Stemming*: this technique aims at reducing the inflected or derived words to their stem. Thus, a user that runs a query on "fishing" would also get documents about "fish" and "fisher". Note that the stem does not need to be a valid word, however, it is necessary that related words map to the same stem. Figure 3.7 shows a passage before and after this technique is applied. Stemming also helps

to reduce the index size. Note that lemmatization could also be used, however, it is more computationally expensive and its implementation is more complex than the stemming technique.

Original text passage
The G8 and G20 summits in Toronto are expected to top $1 billion in costs. But former White House aides say that even with the hefty price, the meetings more than pay for themselves in both tangible and intangible ways.
Text passage after applying stemming
the g8 and g20 summit in toronto ar expect to top $1 billion in costs. but former white hous aid sai that even with the hefti price, the meet more than pai for themselv in both tangibl and intang ways.

Figure 3.7: Text passage after applying stemming.

As said before, we combine these two techniques while indexing the source collection. Figure 3.8 shows an example of the resulting text that is actually used by the IR system (after applying both techniques described above) to build the index. Note the difference in length between the original text passage and the text that is actually indexed.

Original text passage
The G8 and G20 summits in Toronto are expected to top $1 billion in costs. But former White House aides say that even with the hefty price, the meetings more than pay for themselves in both tangible and intangible ways.
Text passage after applying stopword removal and stemming
g8 g20 summit toronto expect top $1 billion costs. white hous aid sai hefti price, meet pai tangibl intang ways.

Figure 3.8: Text passage after combining stopword removal and stemming.

Once the reference collection is indexed, the system is ready to receive queries to retrieve the candidate subdocuments. Thus, for each suspicious document, we also divide it into passages. Note that, as we do with the source documents, the suspicious documents must also be split in their original language to keep the real offset and length of the passage. Only after they are split we can get the English translations from the normalized suspicious documents, as shown in Figure 3.9.

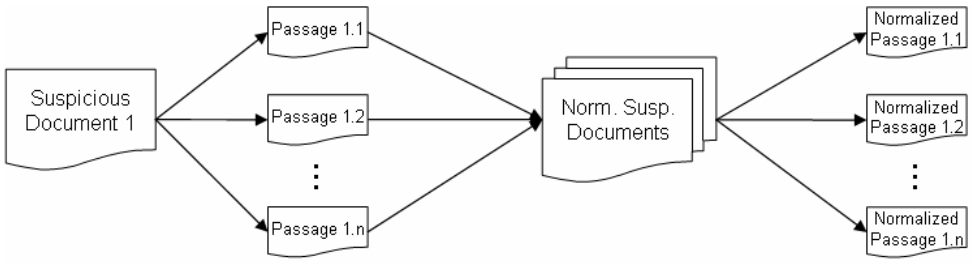

Figure 3.9: Suspicious document division.

As depicted in Figure 3.9, the normalized passages are the ones used to query the index. Therefore, for each normalized text passage (i.e., paragraph) in the suspicious document, we query the index to get the candidate subdocuments with the highest similarity scores. These are the subdocuments with the highest probability of having been used as source of the plagiarism offenses.

Note that we use the same techniques applied during the indexing process (stopword removal and stemming) before submitting the passage to the IR system. However, we also use a term extraction technique in order to keep only the most discriminating terms of the passage. Thus, we only use the terms in the selected passage which have an IDF (Inverse Document Frequency) (MANNING; RAGHAVAN; SCHUTZE, 2008) greater than a certain threshold. The IDF measure assigns a low score to terms that occur very frequently in the collection, i.e., terms that have a low discriminating power. On the other hand, it assigns a high score to terms that occur rarely in the collection, i.e., terms that have a high discriminating power. The formula to calculate the IDF value for a term t is given in Equation 3.1, where N is the number of documents in the collection and n_t is the number of documents that contain the term t. Note that to calculate the IDF value of a term t only the source documents (i.e., the documents in $\boldsymbol{D}$) are considered.

$$\mathrm{IDF_t} = \log\frac{N}{n_t}$$

Equation 3.1: Inverse Document Frequency.

As a result of applying this term extraction technique, the time spent in this phase decreases significantly, since we only pass to the IR system the terms that have the most discriminating power. Thus, since the majority of the discarded terms did not carry any significant meaning (for the sake of retrieval), the IR system does not waste time looking up the index for terms that will not help locate the most relevant candidate subdocuments. An example of a passage after we applied term extraction is shown in Figure 3.10. Note that from the 40 terms in the passage we only passed to the IR system the 8 terms that have an IDF value greater than the pre-defined threshold.

Suspicious text passage
The G8 and G20 summits in Toronto are expected to top $1 billion in costs. But former White House aides say that even with the hefty price, the meetings more than pay for themselves in both tangible and intangible ways.
Text passage after applying stopword removal and stemming
g8 g20 summit toronto expect top $1 billion costs. white hous aid sai hefti price, meet pai tangibl intang ways.
Text passage after applying term extraction
g8 g20 toronto billion hou hefti tangibl intang

Figure 3.10: Text passage after applying term extraction.

At the end of this phase, we have a list of at most ten candidate subdocuments for each passage in the suspicious document. It is important to notice that these subdocuments might belong to different source documents. Figure 3.11 depicts the

overall retrieval process described in this section. Note that the second passage of the suspicious document is being used to query the index. In response, the IR system returns a list of the ten candidate subdocuments in order of estimated relevance. The subdocuments retrieved are then passed to the plagiarism analysis phase, which is described in Section 3.4. In the next section, the feature selection and classifier training phase is described.

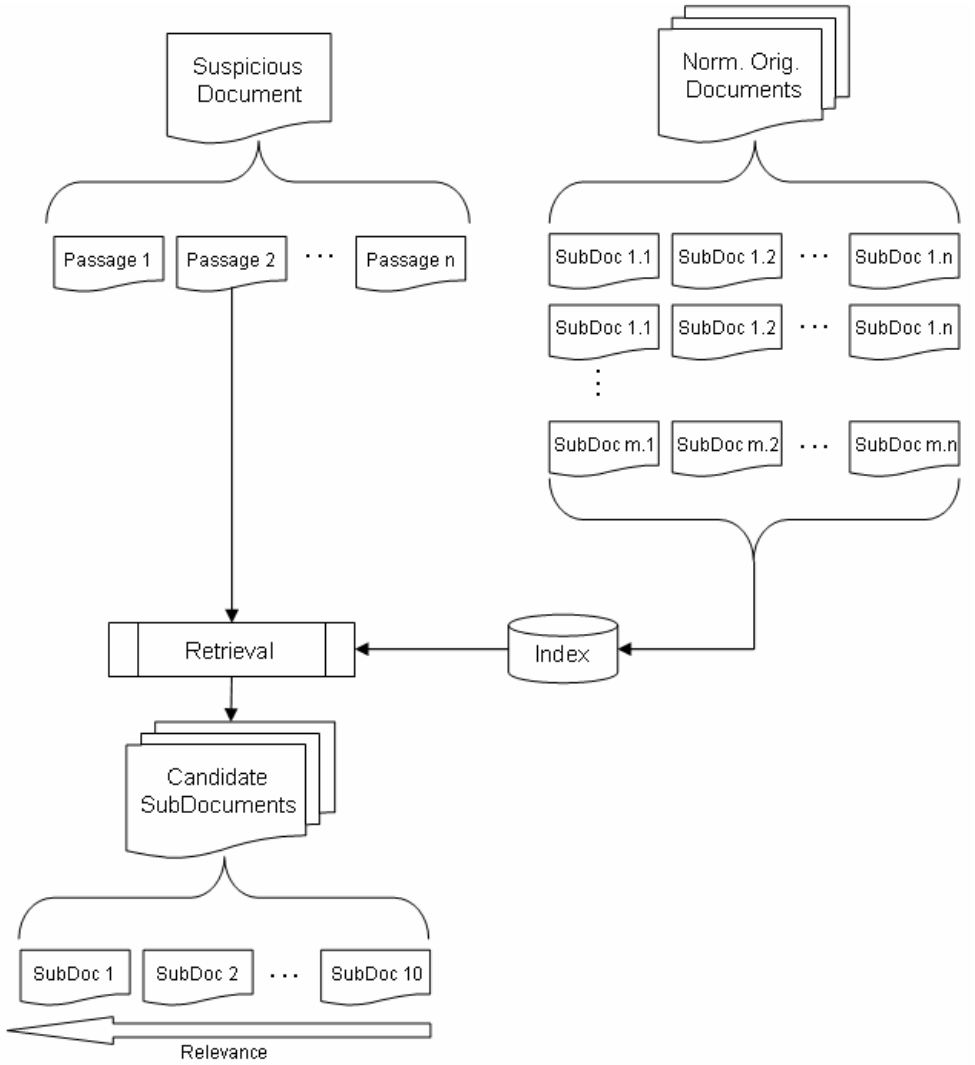

Figure 3.11: Overall retrieval process.

3.3 Feature Selection and Classifier Training

The main goal of this phase is to build a classification model that can learn how to differentiate between a plagiarized and a non-plagiarized text passage. To accomplish this, a training collection ***D''*** composed of suspicious and source documents is used. It is important to notice that the use of classification algorithms is very common in the area of intrinsic plagiarism analysis (ARGAMON; LEVITAN, 2005, KOPPEL; SCHLER, 2004), however, we do not know of any research applying them to the external plagiarism analysis task.

Classification algorithms have been successfully applied to solve problems like medical diagnoses, weather forecast, fraud detection, etc. According to (JIAWEI, 2005), classification is a form of data analysis that aims at extracting models that can describe important data classes. Classification algorithms are composed by two steps. During the first step (the training phase), the classification algorithm builds a classifier by analyzing a set of training instances along with their associated class labels. A training instance, **X**, is an n-dimensional feature vector $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ where each dimension x_i contains the values of the training instances for that feature. Besides, along with each training instance, a predefined class label must be provided. In the second step, the classifier built is used for classification. However, before using the classifier, its predictive accuracy must be estimated in order to check how well it predicts the class label of a set of test instances. If the classifier accuracy is considered acceptable, it can be used to classify instances where the class label is unknown.

In order to build the classification model, we first have to select a set of features to be considered during the training of the classifier. It is important to remember that the goal of the classifier is to decide whether or not a suspicious passage is plagiarized from a candidate subdocument. Therefore, each time a plagiarism analysis is performed between a suspicious passage and one of its candidate subdocuments, we only have two pieces of text to extract information from. After some preliminary tests, we decided to take into account the following features:

- *The cosine similarity between the suspicious passage and the candidate subdocument*: the main reason for using this measure is that the order of the terms does not affect its final similarity score. This is very important since the text passages compared may be originated from different languages or may have been be obfuscated to confuse the plagiarism detector.
- *The similarity score assigned by the IR system to the candidate subdocument*: the higher the score, the higher the chances of a plagiarism offense.
- *The position of the candidate subdocument in the rank generated by the IR system in response to the suspicious passage used as query*: the candidate subdocuments that are actually plagiarized tend to be in the top positions of the rank.
- *The length (in characters) of the suspicious and the candidate subdocument*: passages of similar sizes get higher similarity scores than passages with different sizes. However, even if there is some significant difference, it does not necessarily mean that the suspicious passage does not have some plagiarized text from the candidate subdocument. Therefore, if we get a medium similarity score, but there is some significant difference in length between the passages, there is still a possibility of a plagiarism offense.

Note that these features are the ones that presented the best results during the preliminary experiments. However, it does not mean that other features can not be used as well.

Once the features are selected, we are able to train the classifier. To accomplish this task, we must have some examples of plagiarism offenses to give as input to the classification algorithm. Ideally, we should use real cases of plagiarism, but since it would be very difficult to assemble a plagiarism collection of real world cases, we decided to use an artificial plagiarism corpus. The corpus consists of two different

collections: one with suspicious documents and one with source documents. The plagiarism cases are all properly annotated, i.e., for each document in the suspicious collection, we are able to identify the location of each plagiarized passage and its respective location in the source document. Based on these plagiarism annotations we can provide the necessary information in order to train the classifier.

To generate the training instances to give as input to the classifier we select some random suspicious documents from the collection. For each suspicious document, we proceed exactly as described in Section 3.2. As a result, each passage of each suspicious document generates a list of the top ten candidate subdocuments. Therefore, based on each pair [*suspicious passage, candidate subdocument*], we can extract the information necessary to create the training instances, i.e., we can compute the similarity score, we can get the score assigned by the IR system as well as the rank of the candidate subdocument, and we can find out the length of both suspicious passage and candidate subdocument. Besides, for each pair [*suspicious passage*, *candidate subdocument*], we must inform if the suspicious passage is, in fact, plagiarized from the candidate subdocument. To do this, we simply check out the plagiarism annotations provided with the corpus. As soon as we generate the training instances, we can train the classifier and then, proceed to the plagiarism analysis phase.

3.4 Plagiarism Analysis

Once the classification model is built, we need to create the test instances (extracted from ***D'***) in order to allow the trained classifier decide whether the suspicious document has, in fact, plagiarized passages from one or more of the source documents.

As described in Section 3.2, each suspicious document is split into several passages (one for each paragraph). Each one of the suspicious passages is submitted to the IR system. Thus, for each passage submitted we have a list of the top ten candidate subdocuments which are more likely to be the source of plagiarism. After that, for each suspicious passage p_i and candidate subdocument c_j, we are able to create the test instance with the following information:

- The cosine similarity between p_i and c_j;
- The similarity score assigned by the IR system for the candidate subdocument c_j;
- The position of the candidate subdocument c_j in the IR system rank;
- The length of p_i;
- The length of c_j;

Once the instance is created, the trained classifier is able to decide whether the suspicious passage p_i is, in fact, plagiarized from the candidate subdocument c_j. The plagiarism analysis phase is depicted in Figure 3.12.

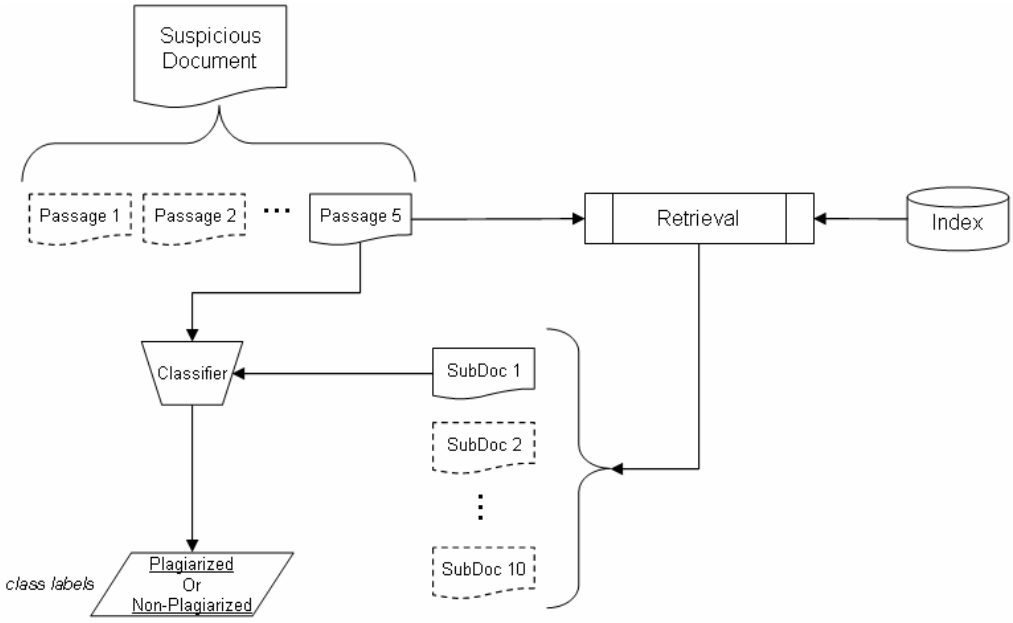

Figure 3.12: Plagiarism analysis phase.

3.5 Result Post-Processing

The goal of this phase is to join the contiguous plagiarized passages detected by the method in order to decrease its final granularity score. The granularity score is a measure that assesses whether the plagiarism method reports a plagiarized passage as a whole or as several small plagiarized passages. An example of a result file before the post-processing technique is shown in Figure 3.13.

```
<?xml version="1.0" encoding="UTF-8"?>
<document reference="A.txt">
 <feature name="detected-plagiarism"
          this_offset="1000" this_length="500"
          source_reference="B.txt"
          source_offset="3000" source_length="500"
 />
 <feature name="detected-plagiarism"
          this_offset="1500" this_length="300"
          source_reference="B.txt"
          source_offset="3500" source_length="300"
 />
</document>
```

Figure 3.13: Example of a result file before the post-processing technique.

Figure 3.13 represents the results of the plagiarism analysis of the suspicious document *A*. The method detected two plagiarized passages. Both plagiarized from document *B*. The first plagiarized passage starts at the 1000th character of the suspicious document *A* and it has a length of 500 characters. This passage was plagiarized from document *B*. The corresponding source passage starts at the 3000th character of source document *B* and it also has length of 500 characters. It is possible to see that the second detection indicates that there is a plagiarized passage located right after the one described in the previous detection. Therefore, instead of reporting two plagiarized passages in document *A*, we can combine these two detections into a single one. The result file after the post-processing is shown in Figure 3.14.

```
<?xml version="1.0" encoding="UTF-8"?>
<document reference="A.txt">
 <feature name="detected-plagiarism"
          this_offset="1000" this_length="800"
          source_reference="B.txt"
          source_offset="3000" source_length="800"
 />
</document>
```

Figure 3.14: Example of a result file after applying the post-processing technique.

Note that the final result file with the detections of the suspicious document *A* now contains only one plagiarized passage, leading to a better granularity score (this measure is explained in more detail in Section 5.3.3).

In order to combine the contiguous detected passages, we use the following heuristics:

(1) Organize the detections in groups divided by the source document (*source_reference* attribute);

(2) For each group, sort them in order of appearance in the suspicious document (i.e., ascending order of the *this_offset* attribute);

(3) Combine adjacent detections that are at most a pre-defined number of characters (in both suspicious and source document) distant from each other;

(4) Keep only one detection (the one with the largest length in the source document) per plagiarized passage, i.e., do not report more than one possible source of plagiarism for the same passage in the suspicious document.

4 ECLAPA TEST COLLECTION

One of the main problems in the evaluation of CLPA systems is the absence of an adequate test collection. Thus, due to the lack of resources to enable comparison between different cross-language plagiarism detection techniques, we decided to create an artificial cross-language plagiarism corpus to evaluate the proposed method. Note that in the evaluation chapter we also used the collections created for the PAN competitions. However, these collections are not specific for cross-language plagiarism analysis and only a small percentage of the cases are cross-lingual. This is one of the reasons why none of the groups participating in the PAN'09 competition tried to detect this type of plagiarism offense. We do not know if a group tried to detect the cross-language plagiarism cases in the PAN'10 competition since its lab reports were not published yet. Another advantage of the test collection described here is that it has an analogous monolingual version which enables a direct comparison of mono and cross-language plagiarism detection.

In order to create the test collection with artificial cross-language plagiarism cases, we used the documents of the Europarl Parallel Corpus (KOEHN, 2005), which is a collection of parallel documents composed of proceedings of the European Parliament. It contains documents for each of the former 11 official languages of the European Union, which means that there are 10 parallel corpora for each language. Although it was first conceived to aid research in statistical machine translation, it has been used to evaluate methods like word sense disambiguation, information extraction, and several other natural language problems.

To build the collection described here, we used the English-Portuguese and the English-French language-pairs to simulate cross-language plagiarism offenses among these three languages. We named the test collection ECLaPA (Europarl Cross-Language Plagiarism Analysis) and it can be freely downloaded from http://www.inf.ufrgs.br/~viviane/eclapa.html.

The ECLaPA test collection is composed of two corpora, one containing only monolingual plagiarism cases and the other containing only multilingual plagiarism cases. Both corpora contain exactly the same plagiarism cases. The documents in the monolingual corpus are all written in English. In the multilingual corpus the suspicious documents are all written in English, but the source documents are written in Portuguese or in French. Thus, in order to simulate plagiarism cases between these languages, we made a script that randomly selects passages from a Portuguese or French document, locates the equivalent English passages, and inserts them in an English document. Details on the ECLaPA test collection are shown in Table 4.1.

Table 4.1: Characteristics of the Test Collection.

---		#Docs	Size	Documents In		
				English	Portuguese	French
Monolingual Corpus	Suspicious	300	89MB	300	0	0
	Source	348	102MB	348	0	0
Multilingual Corpus	Suspicious	300	89MB	300	0	0
	Source	348	115MB	0	174	174

From the 300 suspicious documents in each corpus, 100 do not contain plagiarism cases. Also, from the 348 source documents in each corpus, 100 were not used as source of plagiarism (in the multilingual corpus, 50 Portuguese documents and 50 French documents). Each corpus has a total of 2169 plagiarism cases, from which about 30% are short passages (less than 1500 characters), 60% are medium passages (from 1501 to 5000 characters), and 10% are large passages (from 5001 to 15000 characters). Each suspicious document can have up to five different sources of plagiarism, and from each source, it can have up to 15 plagiarized passages. In the following subsections the necessary steps to create the test collection are described in more detail.

4.1 Pre-Processing

The first step in the creation of the plagiarism test collection was to normalize the two language pairs used: the English-Portuguese pair and the English-French pair. This normalization was necessary since there were some differences between the documents in each language-pair, i.e., some documents were only present in one of the language-pairs. Therefore, these documents had to be discarded. This ensures that each language pair has the exactly same number of documents. After discarding the documents we ended up with 648 parallel documents (one for each language).

We also pre-processed the remaining documents to eliminate some unnecessary meta-information that were present inside the documents (like the “CHAPTER” and the “SPEAKER” tags). An example of a document before and after being pre-processed is shown in Figure 4.1. Note that part of the sentences were omitted due to space constraints. After all the documents were pre-processed we could proceed to the creation of the plagiarism corpus, which is described in the next section.

Document before being pre-processed
<CHAPTER ID=1> Resumption of the session <SPEAKER ID=1 NAME="President"> I declare resumed the session of the European Parliament adjourned […] period. <P> Although, as you will have seen, the dreaded 'millennium bug […] dreadful. You have requested a debate on this subject in the course of the next […] part-session. In the meantime, I should like to observe a minute' s silence, as a [...] European Union. Please rise, then, for this minute' s silence. <P> (The House rose and observed a minute' s silence) …
Document after being pre-processed
Resumption of the session I declare resumed the session of the European Parliament adjourned […] period. Although, as you will have seen, the dreaded 'millennium bug […] dreadful. You have requested a debate on this subject in the course of the next […] part-session. In the meantime, I should like to observe a minute' s silence, as a [...] European Union. Please rise, then, for this minute' s silence. (The House rose and observed a minute' s silence) …

Figure 4.1: Document before and after being pre-processed.

4.2 Simulating Plagiarism Offenses

Once the set of documents to simulate the cross-language plagiarism offenses was defined, we divided it into two different groups: the documents that will be used as source of plagiarism (the source documents) and the documents that will be used to insert the plagiarized passages (the suspicious documents). Therefore, from the 648 documents available, we randomly selected 300 to be used as suspicious documents and 348 to be used as source documents. All the suspicious documents are written in English, while half of the source documents are written in Portuguese and half are written in French. From the 300 suspicious documents, 100 will not contain plagiarism cases, 100 will contain plagiarized passages from documents written in Portuguese, and 100 will contain plagiarized passages from documents written in French. Furthermore, from the 348 source documents, 100 documents were not used as source of plagiarism

(50 documents from each language). Figure 4.2 shows the structure of the corpus as described above.

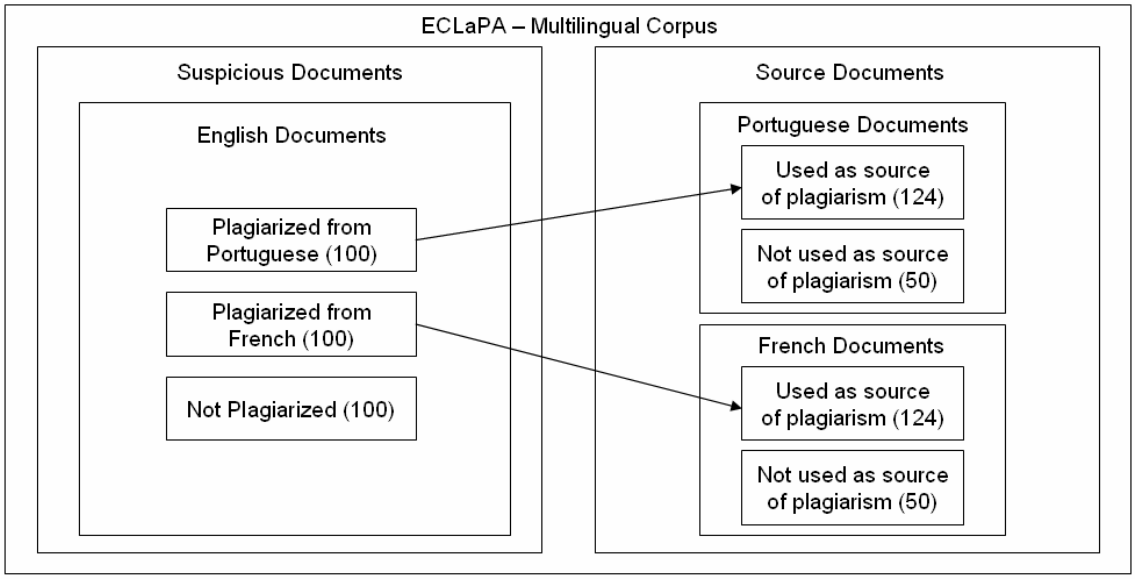

Figure 4.2: Structure of the multilingual corpus.

Thus, to simulate the cross-language plagiarism cases, for each one of the 200 suspicious documents that must contain plagiarism cases, we applied the following heuristic:

(1) Randomly select the source documents that will be used to extract the plagiarized passages. Note that each suspicious document can have plagiarized passages from up to five different source documents. Table 4.2 shows the statistics on how many source documents were used per suspicious document in the corpus.

Table 4.2: Number of source documents per suspicious document.

# Source Documents per Suspicious Document	**Corpus Statistic**
1	30%
2	30%
3	20%
4	10%
5	10%

(2) For each source document selected, we picked a random number of passages to be inserted in the suspicious document. Table 4.3 shows the statistics on how many passages were selected per source document in the corpus.

Table 4.3: Number of passages selected per source document.

# Passages per Source Document	Corpus Statistic
1 – 2	20%
3 – 7	65%
8 – 10	10%
11 – 15	5%

(3) For each passage, we randomly defined its length (in characters). Table 4.4 shows the statistic about the length of the plagiarized passages in the corpus.

Table 4.4: Length of the plagiarized passages.

Length (in characters)		Corpus Statistic
Short	500 – 1500	30%
Medium	1501 – 5000	60%
Large	5001 – 15000	10%

(4) Once the passage length is defined, we select a random piece of text from the source document with the defined length. Note that when the passage is selected, we also have to store the offset (in characters) of the passage in the source document. This information is important because we have to provide the plagiarism annotations after the collection is created.

(5) After the passage is extracted from the source document, we inserted it in a random position of the suspicious document. Here, we also have to store the offset where the plagiarized passage was inserted in the suspicious document. It is important to notice that the suspicious document is written in English, while the source document from where the passage was extracted is written in Portuguese or French. Thus, before inserting it in the suspicious document, we have to get its English translation from its respective parallel English document.

(6) Finally, when all the plagiarized passages are inserted in the suspicious document, we create the corresponding file with the required plagiarism annotations. The annotations are in the same format defined during the PAN competition (PAN-2009, 2009). This format is described in the evaluation chapter.

After the cross-language plagiarism corpus was created, we decided to create an equivalent monolingual plagiarism corpus in order to provide a baseline for comparison. The corpus was created exactly as the multilingual corpus described above. However, we replaced the source documents of the multilingual corpus (written in Portuguese and French) with their respective parallel English documents. Therefore, we created a

monolingual corpus that contains exactly the same plagiarism cases present in the multilingual corpus. The structure of the monolingual corpus is shown in Figure 4.3.

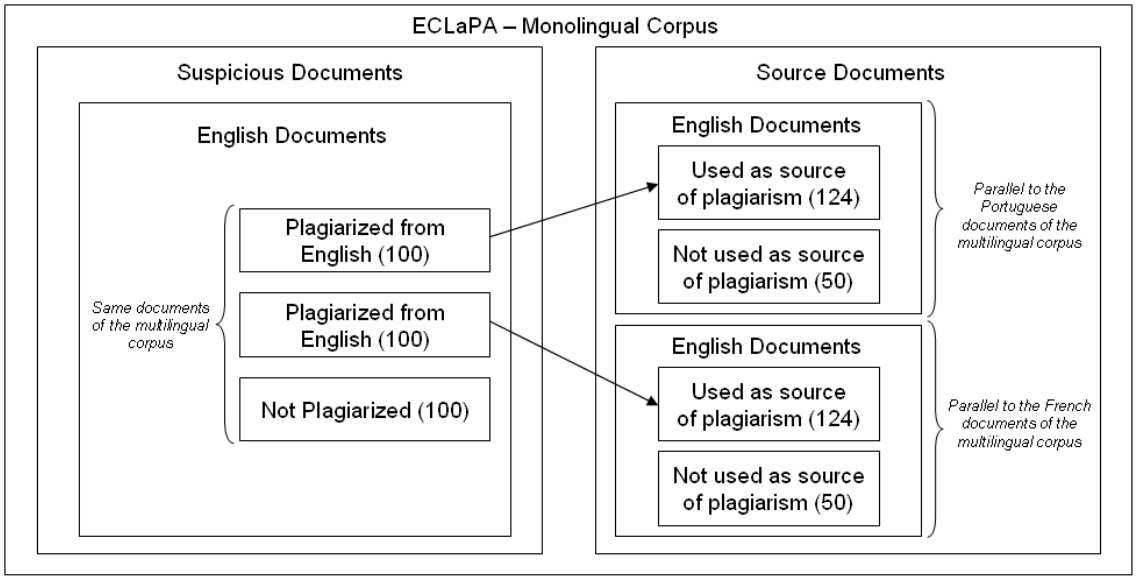

Figure 4.3: Structure of the monolingual corpus.

Note that in Figure 4.3 we kept the same structure from Figure 4.2 to emphasize that the documents used in each corpus were the same. The only difference is that the source documents are also written in English in the monolingual corpus.

An example of a multilingual plagiarized passage and its equivalent monolingual plagiarized passage are shown in Figure 4.4. The text underlined is the one that was plagiarized.

Suspicious passage (the same for both multilingual and monolingual corpus)
How, though, is the Union to be enlarged if we do not do the job properly now, i.e. before enlargement takes place? After the Barrot affair, I know there is a great desire to stop discussing any difficulties there may be with this Commission but, given that I got that one right, I would like to think that Mr Borrell will take the letter that I hand-delivered to him a couple of weeks ago rather more seriously. There are very grave doubts about the hearing of Commissioner Kallas, who, as you know, is in charge of the anti-fraud drive within the European Union. He gave an incorrect date, there was a mistranslation of a question and, in my view, he gave some very misleading information to this Parliament. I have written to Mr Kallas asking for some correct answers, and Mr Borrell received a copy of that letter. We consider it to be indispensable and urgently needed and hope that it actually comes up with the goods that will enable us to carry out enlargement of the European Union in the not too distant future. […]

Multilingual Source Passage
Monsieur le Président, je regrette que M. Borrell n'assure pas la présidence. Après l'affaire Barrot, je sais qu'il existe un profond désir de cesser de débattre de toute difficulté concernant la nouvelle Commission, mais, comme je ne me trompe pas cette fois-ci, j'aime à penser que M. Borrell prendra plus au sérieux la lettre que je lui ai remise en mains propres il y a deux semaines. De sérieux doutes entourent l'audition du commissaire Kallas, qui, comme vous le savez, est chargé de la lutte antifraude dans l'Union européenne. Il a fourni une date incorrecte, une question a été mal traduite et, à mon sens, il a fourni des informations trompeuses à cette Assemblée. J'ai adressé un courrier à M. Kallas lui demandant de fournir des réponses correctes, dont M. Borrell a reçu une copie. Par votre entremise, je demande donc à M. Borrell de veiller à ce que nous obtenions des réponses correctes du commissaire Kallas. En effet, si M. Borrell n'agit pas de la sorte, la réputation de ce Parlement et de l'ensemble de la procédure d'audition tombera encore plus en discrédit. […]
Monolingual Source Passage
Mr President, I am sorry that Mr Borrell is not in the Chair. After the Barrot affair, I know there is a great desire to stop discussing any difficulties there may be with this Commission but, given that I got that one right, I would like to think that Mr Borrell will take the letter that I hand-delivered to him a couple of weeks ago rather more seriously. There are very grave doubts about the hearing of Commissioner Kallas, who, as you know, is in charge of the anti-fraud drive within the European Union. He gave an incorrect date, there was a mistranslation of a question and, in my view, he gave some very misleading information to this Parliament. I have written to Mr Kallas asking for some correct answers, and Mr Borrell received a copy of that letter. So, through you, I am asking Mr Borrell to ensure that we get some correct answers from Commissioner Kallas, for, if Mr Borrell does not do that, then this Parliament and the whole hearings process will fall further into disrepute. […]

Figure 4.4: Example of a plagiarized passage.

As shown in Figure 4.4, we did not do any kind of text obfuscation (to disguise the plagiarism cases) in the monolingual plagiarized passages. We decided to do this way because since the monolingual corpus is supposed to be used as a baseline, the plagiarism detection in this corpus tends to be more accurate if there is no obfuscation introduced in the plagiarized passages. Also, in this way we can measure more precisely the performance loss due to the normalization phase of the proposed method.

5 EVALUATION

In this chapter we provided an evaluation of the proposed approach for cross-language plagiarism analysis described in Chapter 3. We begin by presenting, in Section 5.1, the PAN competition which aims at evaluating plagiarism detection systems. In Section 5.2, all resources used in the experiments are described. Section 5.3 introduces the evaluation measures that are used to analyze the results which are presented in Section 5.4.

5.1 PAN Competition

As stated before, the area of plagiarism analysis lacks a common evaluation framework to enable a fair comparison between the different existent techniques. Since this is a new area of research, the experiments reported in the literature were done over small test collections which most of the times were assembled by the authors. With the goal of solving this problem, in 2009, the 1st International Competition on Plagiarism Detection (PAN-2009, 2009) took place (the PAN'09 competition). The aim was to provide a common basis for the evaluation of plagiarism detection systems.

The PAN'09 competition was divided into two tasks: external plagiarism detection and intrinsic plagiarism detection. However, as mentioned before, the proposed method is designed to detect only external plagiarism cases. Thus, we participated only in one of the tasks.

During the competition the organizers released a training corpus (for each task) to allow the competitors to get familiar with the competition format. In the training corpus, all plagiarism cases were annotated. Thus, the training corpus was also used to tune up the plagiarism detectors of the participating groups. In our case, we were only interested in the corpus of the external plagiarism detection task, which is a large-scale corpus containing artificial plagiarism offenses. The artificial plagiarism cases were created using text obfuscation techniques like shuffling words, replacing a word by its synonym, text translation, etc. Note that cross-language plagiarism cases were also present in the corpus, however, the focus was on monolingual plagiarism and none of the methods used during the competition were designed to detect this type of offense.

After some time, the PAN'09 organizers released the competition corpus (with no plagiarism annotation available), which had the same structure and format of the training corpus. The competitors applied their plagiarism detection methods over the competition corpus and submitted their results. At the end of the competition, the organizers also released the annotated files of the competition corpus along with the competition results.

We ended up in the seventh place in the competition (among ten groups). However, due to time constraints, we were not able to analyze all the documents in the collection since our method was not in its final version. For instance, we were not using the term extraction technique (described in Section 3.2), which led our plagiarism detector to take too long analyzing each suspicious document. Another fact that contributed to our low score was that we were not using any kind of post-processing technique in the detection results.

Since several groups participated in the PAN'09 competition, it turned to be a very good way to promote the area of plagiarism analysis. It also enabled the researchers to compare the performance of their methods against other approaches. Thus, considering the success of the previous competition, in 2010, the second edition of the PAN competition took place.

The PAN'10 competition (PAN-2010, 2010) had the same goals of the previous competition. However, there was an important difference from the previous edition: the organizers decided to create only one corpus for both the external and intrinsic plagiarism analysis tasks. Thus, since our method was not designed to detect intrinsic plagiarism cases, we could only detect the external plagiarism cases of the corpus (which correspond to about 70% of the cases). It is important to notice that, to the best of our knowledge, there is no method capable of detecting both types of plagiarism cases. For instance, in the PAN'09 competition, only one group participated in the two available tasks (applying a different method for each task). We do not have this kind of information about the PAN'10 competition since the lab reports have not been published yet. Another important difference from the previous competition was the introduction of hand-made simulated plagiarism cases.

In the PAN'10 competition, we also ended up in the seventh place (among eighteen groups). However, since we analyzed the whole corpus this time, we achieved a much higher score than in the previous edition.

In the next sections we describe the results achieved while analyzing the corpus of both competitions as well as the results achieved while analyzing the ECLaPA test collection. Besides, to check in what situations the proposed method performs better, we conducted an in-depth analysis of our detection results.

5.2 Resources

In this section we describe all resources used during the evaluation of the proposed method.

5.2.1 Test Collections

During the evaluation of the method, we used three different plagiarism corpora:

(1) *The ECLaPA Test Collection*: the artificial cross-language plagiarism corpus described in Chapter 3 which was created to evaluate the proposed method. As mentioned before, it contains suspicious documents written in English and source documents written in Portuguese and French. For the sake of comparison, an equivalent parallel monolingual plagiarism corpus is also available with both the suspicious and source documents written in English. Further information about the corpus can be seen in Chapter 4.

(2) *The PAN'09 competition corpus*: the corpus used during the PAN'09 competition. Note that this corpus is not specific to cross-language plagiarism cases. In fact, most of them are monolingual. However, the usage of this corpus makes it possible to compare the performance of the proposed method against other approaches. Information about the competition corpus is shown in Table 5.1.

Table 5.1: Characteristics of the PAN`09 Competition Corpus.

-	#Docs	Size	Documents In		
			English	German	Spanish
Source Documents	7215	1,15GB	6764	305	146
Suspicious Documents	7214	1,43GB	7124	0	0

According to Table 5.1, the suspicious documents are all written in English, while the reference collection is composed of documents written in English, German, and Spanish. As reported in (POTTHAST, et al., 2009), the corpus has a total of 36,475 plagiarism cases. It is also important to mention that only half of the suspicious documents contain, in fact, plagiarism cases. The length of the plagiarism cases ranges from 50 to 5000 words. 50% of the documents have 1-10 pages, 35% have 11-100 pages, and 15% have 101-1000 pages.

(3) *The PAN`10 competition corpus*: the corpus used during the PAN'10 competition. This corpus is also not specific to cross-language plagiarism cases. It has almost the same features as in the PAN`09 corpus. However, besides artificial plagiarism cases, it also contains simulated cases, i.e., hand-made (but not real) plagiarism cases. Information about the competition corpus is shown in Table 5.2.

Table 5.2: Characteristics of the PAN`10 Competition Corpus.

-	#Docs	Size	Documents In		
			English	German	Spanish
Source Documents	11148	1,64GB	10483	476	189
Suspicious Documents	15925	3,16GB	15925	0	0

According to Table 5.2, the suspicious documents are all written in English, while the reference collection is composed of documents written in English, German, and Spanish. As reported in (POTTHAST, et al., 2010b), the corpus has a total of 68,558 plagiarism cases. It is also important to mention that only half of the suspicious documents contain, in fact, plagiarism cases. The length of

the plagiarism cases ranges from 50 to 5000 words. 50% of the documents have 1-10 pages, 35% have 11-100 pages, and 15% have 101-1000 pages.

The three corpora described above contain all the plagiarism cases annotated, making it possible for us to check whether we correctly detected a plagiarized passage. Since during the assembling of the ECLaPA test collection we adopted the same annotation format used during the PAN competition, we could easily work with all corpora in a uniform way. An example of the annotation provided with all three corpora is shown in Figure 5.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<document reference="suspicious-document00001.txt">
<feature
  name={"artificial-plagiarism", "simulated-plagiarism"}
  translation={"true", "false"}
  obfuscation={"none", "low", "high"}
  this_offset="1269"
  this_length="3430"
  source_reference="source-document07076.txt"
  source_offset="422"
  source_length="3450"
/>
</document>
```

Figure 5.1: Annotation provided with the corpora.

As shown in Figure 5.1, the annotations provided are in the XML format. There is one XML file for each suspicious document. Inside each file it is possible to identify the name of the document that the file refers to (through the *reference* attribute). For each plagiarized passage inside the suspicious document, there is one *feature* element with the following attributes:

- *name*: indicates whether the plagiarized passage was generated by an automatic process (*artificial-plagiarism*) or by hand (*simulated-plagiarism*). Note that simulated plagiarism was only present in the PAN'10 corpus.
- *translation*: indicates whether the plagiarized passage was plagiarized from a document written in the same language or not. In both the PAN competition corpora this type of plagiarism was generated using an automatic translation tool. In contrast, in the ECLaPA corpus, the translations were extracted from the parallel documents available.
- *obfuscation*: indicates in what extent the plagiarized text passage was obfuscated by the plagiarist, ranging from *none* to *high*. The higher is the obfuscation level, the more difficult it is to detect the plagiarized passage. It is important to notice that when the plagiarized passage is extracted from a document written in a different language (*translation="true"*), the obfuscation level is always set to *none*.
- *this_offset*: indicates the starting position (in characters) of the plagiarized passage inside the suspicious document.
- *this_length*: indicates the length (in characters) of the plagiarized passage inside the suspicious document.

- *source_reference*: indicates the name of the document that was used as source of the plagiarism offense.
- *source_offset*: indicates the starting position (in characters) of the plagiarized passage inside the source document.
- *source_length*: indicates the length (in characters) of the plagiarized passage inside the source document.

Note that the definition described above refers to the external plagiarism cases. As said before, in the PAN'10 corpus there were also intrinsic plagiarism cases. For these cases, the only difference in the XML annotation is that the attributes *source_reference*, *source_offset*, and *source_length* are omitted.

As the description of the plagiarism cases, the detection results are also defined in the XML format. The structure of the file is very similar to the structure presented above. Figure 5.2 shows an example of a file with the detection results.

```
<?xml version="1.0" encoding="UTF-8"?>
<document reference="suspicious-document00001.txt">
<feature
  name="detected-plagiarism"
  this_offset="1269"
  this_length="3430"
  source_reference="source-document07076.txt"
  source_offset="422"
  source_length="3450"
/>
</document>
```

Figure 5.2: Detection result of a suspicious document.

As shown in Figure 5.2, there is one XML file for each suspicious document analyzed. Inside each file it is possible to identify the name of the document that the file refers to (through the *reference* attribute). For each plagiarized passage detected, there is one *feature* element with the following attributes:

- *name*: indicates a detection in the suspicious document.
- *this_offset*: indicates the starting position (in characters) of the detected passage inside the suspicious document.
- *this_length*: indicates the length (in characters) of the detected passage inside the suspicious document.
- *source_reference*: indicates the name of the document that was used as source of the plagiarism offense.
- *source_offset*: indicates the starting position (in characters) of the detected passage inside the source document.
- *source_length*: indicates the length (in characters) of the detected passage inside the source document.

5.2.2 Other Resources

5.2.2.1 Information Retrieval System

As mentioned in Section 3.2, we use an IR system in order to retrieve, based on the passages extracted from the suspicious documents, the subdocuments in the reference collection that are more likely to be used as source of plagiarism. To accomplish this task, we used the Terrier (Terabyte Retriever) Information Retrieval Platform (OUNIS, et al., 2005) as our IR system.

Terrier is an open source search engine developed at University of Glasgow. It is written in Java and it provides various indexing and querying APIs, allowing us to easily integrate it with our method.

According to (OUNIS, et al., 2006), Terrier aims at providing a test-bed framework for driving research and facilitating experimentation in IR. Thus, it was designed as a tool to evaluate, test, and compare models and ideas, and to build systems for large-scale IR. It implements several IR methods and techniques enabling us to evaluate our method under different configurations.

In the experiments described in this chapter, we used the TF-IDF (term frequency times inverse document frequency) weighting scheme (JIAWEI, 2005) as well as stop-word removal (a list of 733 words included in the Terrier platform) and stemming (Porter Stemmer (PORTER, 1997)), which are all available in the Terrier platform. Note that other IR systems could be used as well.

5.2.2.2 Data Mining Software

As described in Section 3.3, we use a classification algorithm in order to build a classification model that is able to distinguish between a plagiarized and a non-plagiarized text passage. To accomplish this task, we used the Weka (Waikato Environment for Knowledge Analysis) Data Mining Software (WEKA, 2009).

Weka is an open source data mining software developed at University of Waikato. It is written in Java and, as Terrier, it also provides an API that can be easily integrated with our method.

According to (FRANK, et al., 2005), Weka was originally developed aiming at processing agricultural data, motivated by the importance of this application area in New Zeland. However, its machine learning methods have grown so quickly that the workbench is now commonly used in all forms of data mining applications (e.g., bioinformatics). The Weka workbench is an organized collection of machine learning algorithms and data pre-processing tools. The workbench includes methods for all the standard data mining problems: regression, classification, clustering, association rule mining, and attribute selection.

For the experiments described in this chapter, we tested several of its classification algorithms, including BayesNet, J48, NaiveBayes, and AdaBoostM1. These tests showed that the J48 classification algorithm (QUINLAN, 1993) had the best results. Thus this is the algorithm used to build the classifier of the proposed method.

5.2.2.3 Language Guesser

During the first step of the language normalization phase (described in Section 3.1), we must identify the language in which each document was written in order to translate

them all to English. To accomplish this task, we used an open source Java API[2] which provides an interface to several functionalities available on Google Translator (GOOGLE-TRANSLATOR, 2009).

5.2.2.4 Automatic Translation Tool

In the second step of the language normalization phase (described in Section 3.1), we translate all the non-English documents of the collection to English. In order to do that, we used the LEC Power Translator 12 (LEC, 2008) as our automatic translation tool. The reason why we decided to use this translator is that, instead of translating each document at a time, it was the only one that enabled us to translate an entire directory of documents. This is an important feature since we do not know of any translator that provides an API to be called by other systems. It is important to notice that we could also use the Google Translator Java API. However, using it to translate a large amount of data through the Internet can be very time consuming.

5.3 Evaluation Metrics

In order to evaluate the proposed method we employed the same evaluation metrics used in both PAN competitions (refer to (POTTHAST, et al., 2010b, POTTHAST, et al., 2009) for further information on them): recall, precision, and granularity. Figure 5.3 below is used as an example to help explain the metrics. As it is possible to see, there are three plagiarized passages to be detected ($\mathbf{s_1}$, $\mathbf{s_2}$, and $\mathbf{s_3}$). However, the plagiarism detector reported four detections ($\mathbf{r_1}$, $\mathbf{r_2}$, $\mathbf{r_3}$, and $\mathbf{r_4}$). In the next subsections we describe how to calculate each metric based on the example showed in Figure 5.3.

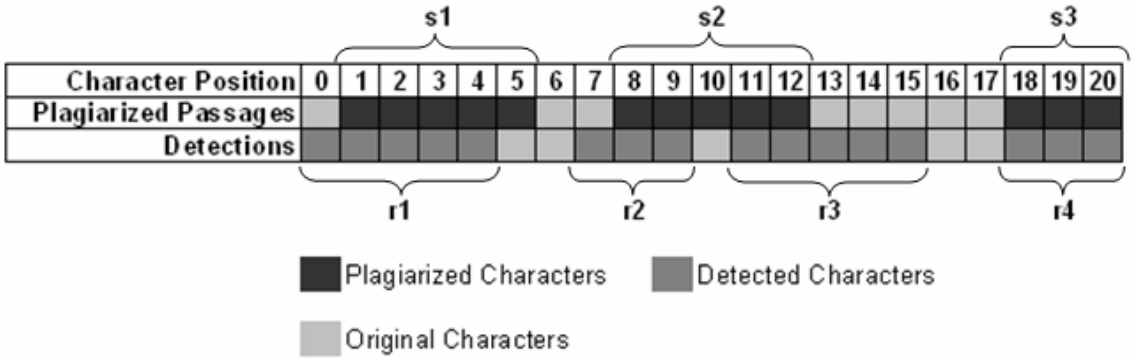

Figure 5.3: Plagiarized passages and their respective detections.

5.3.1 Recall

Let ***s*** be a plagiarized passage (i.e., a contiguous sequence of plagiarized characters) from document ***d***, and let ***S*** be the set of all plagiarized passages. Recall measures the number of plagiarized characters that are actually detected by the plagiarism detector. According to the example in Figure 5.3, it is possible to see that the plagiarism detector detected all three plagiarized passages ($\mathbf{s_1}$, $\mathbf{s_2}$, and $\mathbf{s_3}$). However, not all the plagiarized characters were detected. The formula to calculate the recall measure is given in Equation 5.1, where $\boldsymbol{s_i}$ is a plagiarized passage from the set ***S*** of all plagiarized passages.

[2] http://code.google.com/p/google-api-translate-java/

$$\text{Recall} = \frac{1}{|S|}\sum_{i=1}^{|S|}\left(\frac{\#Detected\_chars\_of\_s_i}{|s_i|}\right)$$

Equation 5.1: Recall.

Applying this measure to the example shown in Figure 5.3 we have:

Recall = 1/3 * (4/5 + 4/5 + 3/3) = 1/3 * (0.8 + 0.8 + 1.0) = 1/3 * 2.6 = 0.86

5.3.2 Precision

Let ***r*** be a detection (i.e., a contiguous sequence of possible plagiarized characters) found by the plagiarism detector, and let ***R*** be the set of all detections. Precision measures the number of characters detected that are actually plagiarized. According to the example in Figure 5.3, it is possible to see that the plagiarism detector detected all three plagiarized passages ($\mathbf{s_1}$, $\mathbf{s_2}$, and $\mathbf{s_3}$). However, not all the detected characters were, in fact, plagiarized. The formula to calculate the precision measure is given in Equation 5.2, where $\boldsymbol{r_i}$ denotes a detection from the set ***R*** of all detections.

$$\text{Precision} = \frac{1}{|R|}\sum_{i=1}^{|R|}\left(\frac{\#Plagiarized\_chars\_of\_r_i}{|r_i|}\right)$$

Equation 5.2: Precision.

Applying this measure to the example showed in Figure 5.3 we have:

Precision = 1/4 * (4/5 + 2/3 + 2/5 + 3/3) = 1/4 * (0.8 + 0.67 + 0.4 + 1.0) = 1/4 * 2.87 = 0.72

5.3.3 Granularity

The granularity score is a measure that assesses whether the plagiarism method reports a plagiarized passage as a whole or as several small plagiarized passages. In the example showed in Figure 5.3 it is possible to see that, although there are only three plagiarized passages to be detected, the plagiarism detector reported four detections. An ideal plagiarism detector should report only one detection per plagiarized passage. The formula to calculate the granularity measure is given in Equation 5.3, where $\boldsymbol{S_R}$ is the subset of ***S*** for which detections exist in ***R***. It is important no notice that to be considered correct, a detection ***r*** must report at least one correct character in both the plagiarized and the source passage of its corresponding ***s***.

$$\text{Granularity} = \frac{1}{|S_R|}\sum_{i=1}^{|S_R|}\left(\#Detections\_of\_s_i\_in\_R\right)$$

Equation 5.3: Granularity.

Applying this measure to the example showed in Figure 5.3 we have:

Granularity = 1/3 * (1 + 2 + 1) = 1/3 * 4 = 1.33

5.3.4 Overall Score

The overall score is a measure that combines recall, precision, and granularity to provide an absolute ordering between the performances of the different methods. This measure was created mainly to facilitate the creation of the PAN competition rank. The formula to calculate the overall score is given in Equation 5.4, where *F* is the harmonic mean of precision and recall. Note that the logarithm function is applied to decrease the impact of the granularity score in the overall score.

$$\text{Overall Score} = \frac{F}{\log_2(1+G)}$$

Equation 5.4: Overall Score.

Applying this measure to the example showed in Figure 5.3 we have:

F = (2 * 0.86 * 0.72) / (0.86 + 0.72) = 1.24 / 1.58 = 0.78

Overall Score = 0.78 / log2(1 + 1.33) = 0.78 / l.22 = 0.64

5.4 Experimental Results

In this section we present the results achieved after analyzing the test collections described earlier. As mentioned before, in order to analyze each test collection, we used the Terrier Information Retrieval Platform (OUNIS, et al., 2005) as our IR system. In particular, we used the TF-IDF weighting scheme as well as stop-word removal (a list of 733 words included in the Terrier platform) and stemming (Porter Stemmer (PORTER, 1997)). To build the classifier, we used the Weka Data Mining Software (WEKA, 2009). More specifically, we applied the J48 classification algorithm (QUINLAN, 1993). Our task is to detect all the plagiarized passages in the suspicious documents and their corresponding text passages in the source documents. The documents were divided into several subdocuments before translation in order to keep the original offset and length of each paragraph in the original document. As mentioned in Section 3.1, during the language normalization phase, we have to identify the language of each suspicious and source document in order to translate them all to English. To accomplish this task, we used the LEC Power Translator 12 (LEC, 2008) as our translation tool and the Google Translator (GOOGLE-TRANSLATOR, 2009) as our language guesser. It is important to notice that in all of the available test collections, the multilingual documents are only present in the reference collection and all the suspicious documents are written in English. Thus, we only applied the language normalization phase in the reference collection of each corpus.

The next section describes the parameters of the method along with their respective values that were used during the experiments. In the remaining sections, the results achieved during the analysis of each test collection are presented.

5.4.1 Detection Parameters

During the employment of the method, some parameters must be defined. Below we present these parameters and the respective values that were used to produce the results showed throughout this section. Note that the values presented below were all originated from exhaustive preliminary tests. It is also important to mention that the change of any of these values is a tradeoff between result quality and processing time.

- *Subdocument length*: defines the minimum length, in characters, that a subdocument must have in order to be indexed by the IR system. After some preliminary tests, we decided to index only the subdocuments with length greater than 250 characters. The definition of this kind of restriction aims at speeding up the retrieval process, since the IR system will have fewer subdocuments to lookup in the index.
- *Subdocuments retrieved per suspicious passage*: the IR system retrieves at most 10 candidate subdocuments for each suspicious passage submitted. This restriction is also defined to speed up retrieval. The rationale behind this is that the candidate subdocuments that are actually plagiarized tend to be in the top positions of the IR rank.
- *IDF threshold*: instead of using all the terms of the suspicious passage to query the index, we discarded the terms which had an IDF value lower than 2.41. This is one of the most important parameters of the method, since if we consider all the terms of the suspicious passage, the time spent in the retrieval phase increases significantly. Besides, only the terms that are actually relevant tend to remain, leading to almost no performance loss due to this restriction.
- *IR score threshold*: defines the minimum score that a subdocument must receive (by the IR system) in order to be considered during the plagiarism analysis phase. After some preliminary tests, we decided to discard the subdocuments that received a score lower than 11. Note that this parameter depends on the IR system used as well as the weighting scheme. As mentioned before, in the experiments presented here, we used the Terrier IR system as well as the TF-IDF weighting scheme.
- *Merge threshold*: defines the maximum distance, in characters, that two plagiarized passages must be from each other in order to be merged during the post-processing phase. In the experiments described here we merged the contiguous plagiarized passages that were at most 3000 characters distant from each other.

The parameters described above are summarized in Table 5.3:

Table 5.3: Method Parameters.

Retrieval Parameters	
Subdocument length (in characters)	250
Subdocuments retrieved per suspicious passage	10
IDF threshold	2.41
IR score threshold	11
Post-Processing Parameters	
Merge threshold (in characters)	3000

5.4.2 ECLaPA – Experimental Results

In this section we present the results achieved after analyzing the ECLaPA test collection. Note that this test collection provides two corpora: one containing only monolingual plagiarism cases and the other one containing only cross-language plagiarism cases. Thus, we analyzed each corpus separately.

Once all the documents in the reference collections were divided into subdocuments and translated into English, we indexed them. Information about the indexes is shown in Table 5.4.

Table 5.4: ECLaPA - Information about the indexes.

---	**Monolingual**	**Multilingual**
# Subdocuments Indexed	134,406	143,861
# Terms	8,379,357	8,786,678
# Unique Terms	37,262	45,370
Size (MB)	38	41

Note that it was not necessary to apply the language normalization phase in the monolingual corpus since all the documents were already written in English. After the reference collections were indexed, we were able to start analyzing each one of their suspicious documents.

In order to build the classification model, a training collection with the same characteristics as the test collection was created. The training instances were randomly selected from the suspicious documents. For each suspicious document the top ten candidate subdocuments were retrieved. Based on each pair [*suspicious passage, candidate subdocument*], we extracted the information necessary to create the training instances. Given that the training collection also comes with the plagiarism annotations, we could easily check if the training instances were positive or negative examples of plagiarism. After extracting the necessary information, we generated the ARRF (Attribute-Relation File Format) file containing the training instances according to the Weka file format (an example of this file is shown in the Appendix A). After generating the training instances, we applied the J48 algorithm to build the classification model. Once the classifier is trained, we analyzed each suspicious document of the test collections in order to find the plagiarized passages.

We compared the performance of the method when detecting the plagiarism cases of the multilingual corpus against its performance when detecting the plagiarism cases of the monolingual corpus. Since both corpora contain the same plagiarism cases, we believe this experiment can provide us an idea of how well our method handles cross-language plagiarism. The final results of the experiment are shown in Table 5.5.

According to Table 5.5, the cross-language experiment achieved 86% of the performance (final score) of the monolingual baseline. This is comparable with the state of the art in CLIR. It is also possible to see that the recall was the most affected measure when dealing with cross-language plagiarism. We attribute the 22% drop in recall to the loss of information incurred by the translation process. As a result, the similarity score assigned by the IR system decreased, leading to more subdocuments being discarded

during the retrieval phase. Our post-processing step allowed for perfect granularity in both settings.

Table 5.5: ECLaPA - Experimental Results.

---	**Monolingual**	**Multilingual**	**% of Monolingual**
Recall	0.8648	0.6760	78.16%
Precision	0.5515	0.5118	92.80%
F-Measure	0.6735	0.5825	86.48%
Granularity	1.0000	1.0000	100%
Final Score	0.6735	0.5825	86.48%

To analyze in which situations the method performs better, we investigated to what extent the length of the plagiarized passage affects the results. Table 5.6 shows the results of the analysis. We divided the plagiarized passages according to their textual length (in characters): short (less than 1500 characters), medium (from 1501 to 5000 characters), and large (from 5001 to 15000 characters).

Table 5.6: ECLaPA - Detailed Analysis.

---	**Monolingual**			**Multilingual**		
	Short	Medium	Large	Short	Medium	Large
Detected	435	1289	239	242	1190	239
Total	607	1323	239	607	1323	239
%	71	97	100	39	90	100

According to Table 5.6, when considering only the monolingual plagiarism cases, our method detected 90% of the passages (1963). As for the multilingual plagiarism cases, the method detected 77% (1671) of the passages. As expected, the length of the plagiarized passage affects the results considerably. The larger the passage the easier the detection. All large plagiarized passages were detected in both mono and multilingual settings. However, only 39% of short passages in the multilingual corpus were detected. We compared the detection of plagiarism from documents in Portuguese and in French. The results were almost identical (overall score of 0.62 and 0.61 respectively). The time spent by the method to analyze the test collections is shown in Section 5.4.5.

5.4.3 PAN'09 – Experimental Results

In this section we present the results achieved after analyzing the PAN'09 test collection. It is important to mention that the results presented here are not the official results we got during the competition. As stated before, due to time constraints, we were not able to analyze all the documents of the collection at the time of the competition.

To analyze the suspicious documents of the PAN'09 competition, we proceeded the same way described in the previous section. The only difference is that we analyzed a different corpus. Thus, after we have all the documents in the reference collection divided into subdocuments and translated into English, we indexed the collection. Information about the resulting index is shown in Table 5.7.

Table 5.7: PAN'09 - Information about the index.

# Subdocuments Indexed	1,296,971
# Terms	81,991,405
# Unique Terms	640,426
Size (MB)	436

To train the classifier, we used the training corpus of the competition. To accomplish this, we selected 50 suspicious documents to be used to create the training instances. Then, we generated the ARRF file containing the training instances. After, we applied the J48 algorithm to build the classification model. Once the classifier is trained, we proceeded to the analysis of each suspicious document of the competition corpus.

Since none of the participating groups of the PAN`09 competition tried to detect the cross-language plagiarism cases of the competition, we tried to compare our method against its monolingual version. We know that this is not the ideal test scenario, since the documents used in the two experiments are not the same. However, we believe this experiment can provide us an idea of how well our method handles cross-language plagiarism when compared to its overall performance when analyzing monolingual plagiarism cases. Therefore, in order to evaluate the performance of the method when analyzing documents containing cross-language plagiarism, we decided to perform two different experiments. The first experiment considers only the documents that contain monolingual plagiarism offenses, i.e., documents written in English which have text passages plagiarized from other documents also written in English. The second experiment considers only the documents that contain cross-language plagiarism offenses, i.e., documents written in English which have text passages plagiarized from documents written in German and/or Spanish. The final results of the experiments are shown in Table 5.8.

According to Table 5.8, the cross-language experiment achieved 73% of the performance (final score) of the monolingual baseline. The recall and the precision of the cross-language experiment achieved, respectively, 59% and 89% of their monolingual counterparts. This shows that recall was the most affected part of the method when dealing with cross-language plagiarism, while precision had only 11% performance loss.

To analyze in which situations the method performs better, we investigated how well it handles text obfuscation and in what level the length of the plagiarized passage affects its overall performance. Table 5.9 shows the results of the analysis. Due to the plagiarism annotation in the suspicious documents, we could identify whether a plagiarized text passage was obfuscated (and to which extent). We also divided the plagiarized passages according to its textual length (in characters): short (less than 1500

characters), medium (from 1501 to 5000 characters), and large (greater than 5000 characters).

Table 5.8: PAN`09 - Experimental results.

---	**Monolingual**	**Multilingual**
Recall	0.6066	0.3580
Precision	0.6326	0.5684
F-Measure	0.6194	0.4393
Granularity	1.0453	1.0000
Final Score	0.6000	0.4393

Table 5.9: PAN`09 - Detailed analysis.

Short Plagiarized Passages						
-	**Monolingual**			**Multilingual**		
Obfusc.	**Detect.**	**Total**	**%**	**Detect.**	**Total**	**%**
None	676	3191	21	91	915	10
Low	536	3190	16	---	---	---
High	416	3025	13	---	---	---
Medium Plagiarized Passages						
-	**Monolingual**			**Multilingual**		
Obfusc.	**Detect.**	**Total**	**%**	**Detect.**	**Total**	**%**
None	2960	3757	78	442	578	76
Low	2389	3704	64	---	---	---
High	2044	3758	54	---	---	---
Large Plagiarized Passages						
-	**Monolingual**			**Multilingual**		
Obfusc.	**Detect.**	**Total**	**%**	**Detect.**	**Total**	**%**
None	7022	7097	99	177	192	92
Low	6839	6934	98	---	---	---
High	89	134	66	---	---	---

According to Table 5.9, when considering only the monolingual plagiarism cases, our method detected 22,971 passages out of 34,790 (i.e., 66%). When considering multilingual plagiarism cases, the method detected 710 passages out of 1685 (i.e., 42%). It is also possible to see that, as expected, the level of text obfuscation affects the results

considerably, especially when handling short passages. Regarding the textual length of the plagiarized passage, the larger is the passage the easier is the detection (when analyzing large plagiarized passages with no kind of text obfuscation, the method detected 99% of the plagiarized passages). It is also worth mentioning that in this test collection, in the case of the multilingual plagiarized passages, there are much more short passages than medium and large ones. Due to this fact, the percentage of multilingual passages detected was affected considerably. Note that multilingual plagiarism cases did not suffer any kind of text obfuscation, but again, the text translation itself can be considered as a kind of low level text obfuscation. The time spent by the method to analyze the test collection is shown in Section 5.4.5.

5.4.4 PAN'10 – Experimental Results

In this section we present the results achieved after analyzing the PAN'10 test collection. Note that the results presented here are the official results we got during the competition. It is also worth mentioning that differently from the previous two test collections analyzed this one also contains intrinsic plagiarism cases. Therefore, since our method was designed to detect only external plagiarism analysis cases, the recall measure ended up getting negatively affected.

In order to analyze the competition corpus, we proceeded the same way described in the previous two sections. Information about the index is shown in Table 5.10.

Table 5.10: PAN'10 - Information about the index.

# Subdocuments Indexed	1,861,401
# Terms	124,049,701
# Unique Terms	788,901
Size (MB)	623

As in the PAN`09 competition, we used the provided training corpus in order to build the classification model. Table 5.11 shows our overall result in the competition as well as the result of the analysis when considering only the external plagiarism cases. Note that since the competition corpus had both external and intrinsic plagiarism cases mixed up, the recall value ended up getting affected since the applied method was designed to detect only external plagiarism cases.

Table 5.11: PAN`10 - Experimental results.

---	**Competition**	**Only External Cases**
Recall	0.4036	0.4966
Precision	0.7242	0.7242
F-Measure	0.5183	0.5892
Granularity	1.0024	1.0017
Final Score	0.5175	0.5881

With the final score of 0.5175 our group got the seventh place in the competition. Table 5.12 shows an in-depth analysis of the results. We provide an overall analysis considering the results of the competition and we also analyze our results while detecting only the external plagiarism cases (which is the focus of the applied method). To analyze in which situations the method performs better, we investigated how well it handles text obfuscation and in what level the length of the plagiarized passage affects its overall performance. We divided the plagiarized passages according to their textual lengths: short (less than 1500 characters), medium (from 1501 to 5000 characters), and large (greater than 5000 characters).

Table 5.12: PAN`10 - Detailed analysis.

Short Plagiarized Passages						
-	**Competition**			**Only External**		
Obfuscation	**Detected**	**Total**	**%**	**Detected**	**Total**	**%**
None	78	9395	0.83	78	4088	1.90
Low	63	3798	1.65	63	3798	1.65
High	37	3729	0.99	37	3729	0.99
Translated	194	2417	8.02	194	1754	11.06
Simulated	211	2362	8.93	211	2362	8.93
Medium Plagiarized Passages						
-	**Competition**			**Only External**		
Obfuscation	**Detected**	**Total**	**%**	**Detected**	**Total**	**%**
None	2509	9907	25.32	2509	5911	42.44
Low	1832	4722	38.79	1832	4722	38.79
High	1415	4752	29.77	1415	4752	29.77
Translated	980	2358	41.56	980	1851	52.94
Simulated	268	624	42.94	268	624	42.94
Large Plagiarized Passages						
-	**Competition**			**Only External**		
Obfuscation	**Detected**	**Total**	**%**	**Detected**	**Total**	**%**
None	6755	8733	77.35	6755	6785	99.55
Low	6343	6363	99.68	6343	6363	99.68
High	6171	6275	98.34	6171	6275	98.34
Translated	2630	3123	84.21	2630	2709	97.08
Simulated	0	0	100.00	0	0	100.00

According to Table 5.12, during the competition the method detected 29,486 out of 68,558 plagiarized passages (i.e., 43%). When ignoring the intrinsic plagiarism cases,

the method detected 29,486 out of 55,723 plagiarized passages (i.e., 53%). It is possible to see that the method performed poorly while detecting short plagiarized passages. This is partially explained by our decision of indexing only the subdocuments with length greater than 250 characters (to speed up retrieval). Table 5.12 also shows that, other than translation, the intrinsic plagiarism cases did not suffered any kind of obfuscation. While detecting medium plagiarized passages, the performance of the method decreased as the level of obfuscation increased (none to high). It is worth noticing that the translated and the simulated plagiarized passages did not seem to have a negative impact in the performance of the method, since the percentage of the passages detected is not lower than for the other types of obfuscation. Finally, when detecting large plagiarized passages the method detected almost all of them, regardless of the type of obfuscation (note that that were no large simulated plagiarized passage). The time spent by the method to analyze the test collection is shown in Section 5.4.5.

5.4.5 Processing Time Analysis

In this section we present information about the time spent by the method to analyze each test collection.

Table 5.13: Processing Time Analysis.

---	ECLaPA - Monolingual	ECLaPA - Multilingual	PAN'09	PAN'10
Number of Suspicious Documents	300	300	7214	11148
Number of Source Documents	348	348	7215	15925
Total Analysis Time	1 hour and 50 minutes	1 hour and 53 minutes	~ 88 hours	~ 230 hours
Average Time / Suspicious Document	22,2 seconds	22,6 seconds	44 seconds	52 seconds
KB Analyzed / Minute	828KB	806KB	273KB	236KB
Suspicious Document Average Size	303KB	303KB	204KB	206KB

According to Table 5.13, it is possible to see how long the method took to analyze the suspicious documents of each test collection. Note that the main factor that influences the time spent during the analysis is the number of source documents. This is due to the fact that the IR system takes more time to retrieve the candidate subdocuments as the number of source documents increases. This fact is clearly visible when looking at the amount of kilobytes analyzed per minute in each collection. During the analysis of the ECLaPA test collections the method processed approximately three times more data than when analyzing the collections of the PAN competitions.

It is also important to notice that the time spent during the translation of the non-English documents is not considered in Table 5.13. However, note that the source documents only need to be translated once (before the indexing process). During the experiments the translation tool translated around (depending on the language of the document) 85 kilobytes of text per minute.

6 CONCLUSIONS

This work proposes and evaluates a method for CLPA. The evaluation experiments show that it is a viable approach to the task of cross-language plagiarism analysis. The proposed method employs techniques and strategies taken from different areas of research like monolingual plagiarism analysis, cross-language information retrieval, and data mining.

The cross-language plagiarism analysis method proposed is language independent, capable of handling documents written in several different languages. The only resources necessary to accomplish this are a language guesser, an automatic translation tool that is able to translate from one language to another, and an appropriate stemming algorithm to be used in the documents translated to the default language (English was chosen as the default language in this work).

The method is divided into five main phases, where any phase can be easily modified in order to test other different strategies. The main difference of our method compared to the existing ones is that we used a classification algorithm in order to decide whether a text passage is plagiarized or not. To the best of our knowledge, there is no research in the area of external plagiarism analysis that employs this kind of strategy.

We evaluated our method using three freely available test collections. Two of them were created for the PAN competitions (PAN'09 and PAN'10), which is an International Competition on Plagiarism Detection. However, only a small percentage of these two collections contained cross-language plagiarism cases. Therefore, we decided to create an artificial test collection especially designed to contain this kind of offense. We named the test collection ECLaPA. Since it can be freely downloaded, it enables a fair comparison between different cross-language plagiarism methods.

During the experiments, we evaluated in what situations the method performs better. As expected, the length of the plagiarized passage affects the results considerably. The larger the passage the easier the detection. It is also possible to see that the level of text obfuscation affects the results considerably, especially when handling short passages. The method performed poorly while detecting short plagiarized passages, especially in the PAN'10 competition corpus. This is partially explained by our decision of indexing only the subdocuments with length greater than 250 characters (to speed up retrieval). Finally, when detecting large plagiarized passages the method detected more than 90% of them (in all test collections), regardless of the type of obfuscation.

6.1 Contributions

Here is a list of the main contributions of this work:

- The definition of a new cross-language plagiarism analysis method that employs a classification algorithm to build a model that is able to distinguish between a plagiarized and a non-plagiarized text passage. This approach was not used to tackle the problem of external plagiarism analysis so far.
- The evaluation of the method against three plagiarism test collections. This enables future methods to be compared against ours.
- The creation of a plagiarism test collection especially designed to contain cross-language plagiarism cases. This collection can be freely downloaded and used to assess the performance of future methods.

6.2 Published Papers

This dissertation has resulted in two published papers:

1. Pereira, R.C., V.P. Moreira, and R. Galante, A New Approach for Cross-Language Plagiarism Analysis, in Proceedings of the CLEF 2010 Conference on Multilingual and Multimodal Information Access Evaluation, M. Agosti, et al., Editors. 2010, Springer: Padua, Italy.

2. Pereira, R.C., V.P. Moreira, and R. Galante, UFRGS@PAN2010: Detecting External Plagiarism, in Proceedings of the PAN 2010 Lab on Uncovering Plagiarism, Authorship, and Social Software Misuse, M. Agosti, et al., Editors. 2010, Springer: Padua, Italy.

6.3 Future Work

Although we achieved good results during the experiments conducted, there are still some points that can be improved. One of them is the performance of the method while detecting short plagiarized passages. This low performance is partially due to the fact that we restricted the length of the passages indexed by the IR system (in order to speed up retrieval). Thus, if we could in some way improve the time spent during the analysis of each suspicious document (like analyzing each suspicious passage in a different computer), this length restriction may be avoided.

Another interesting point is to try to find other features to be used during the training phase of the method. Thus, the classifier built would have more information to decide whether the suspicious passage is plagiarized or not.

Finally, it would also be very interesting to test the performance of the method while detecting plagiarism between documents written in unrelated languages, like English versus Chinese and/or Japanese. Many real plagiarism cases happen between these pairs of languages.

REFERENCES

ARGAMON, S.; LEVITAN, S. Measuring the Usefulness of Function Words for Authorship Attribution. In: **Association for Literary and Linguistic Computing/ Association Computer Humanities**, University Of Victoria, Canada, 2005.

BARRÓN-CEDEÑO; ROSSO, P. On Automatic Plagiarism Detection Based on n-Grams Comparison. In: **Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval**, Toulouse, France, p.696-700, Springer-Verlag, 2009.

BARRÓN-CEDEÑO, A.; ROSSO, P.; BENEDÍ, J.-M. Reducing the Plagiarism Detection Search Space on the basis of the Kullback-Leibler Distance. In: **Proc. 10th Int. Conf. on Comput. Linguistics and Intelligent Text Processing**, LNCS(5449), p.523-534, Springer-Verlag, 2009.

BARRÓN-CEDEÑO, A., et al. On Cross-lingual Plagiarism Analysis using a Statistical Model. In: **Proceedings of the ECAI'08 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse**, Patras, Greece, 2008.

CESKA, Z.; TOMAN, M.; JEZEK, K. Multilingual Plagiarism Detection. In: **Proceedings of the 13th international conference on Artificial Intelligence: Methodology, Systems, and Applications**, Varna, Bulgaria, p.83-92, Springer-Verlag, 2008.

FRANK, E., et al. Weka - a machine learning workbench for data mining. In: **Oded Maimon and Lior Rokach**, p.1305-1314, Springer, 2005.

FUJII, A.; ISHIKAWA, T. Japanese/English cross-language information retrieval: exploration of query translation and transliteration. In: **Computers and the Humanities**. 2004. 389-420p.

GEY, F.; JIANG, H. English-German cross-language retrieval for the GIRT collectionexploiting a multilingual thesaurus. In: **Text REtrieval Conference**, Gaithersburg, Maryland, p.219-234, 1999.

GOOGLE TRANSLATOR. Available at: <http://www.google.com/translate_t>. Visited on Oct. 29, 2009.

GREFENSTETTE, G. Cross-Language Information Retrieval. 1998. Kluwer Academic Publishers. 182p.

GROZEA, C.; GEHL, C.; POPESCU, M. ENCOPLOT: Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection. In: **Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse**, San Sebastian (Donostia), Spain, p.10-18, 2009.

HULL, D. A.; GREFENSTETTE, G. Querying Across Languages, a Dictionary-based approach to Multilingual Information Retrieval. In: **19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**, Zurich, Switzerland, 1996.

JIAWEI, H. Data Mining: Concepts and Techniques. 2005. Morgan Kaufmann Publishers Inc.

KASPRZAK, J.; BRANDEJS, M.; KŘIPAČ, M. Finding Plagiarism by Evaluating Document Similarities. In: **Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse**, San Sebastian (Donostia), Spain, p.24-28, 2009.

KOEHN, P. Europarl: A Parallel Corpus for Statistical Machine Translation. In: **MT Summit**, 2005.

KOPPEL, M.; SCHLER, J. Authorship Verification as a One-Class Classification Problem. In: **Proceedings of the 21st International Conference on Machine Learning**, Banff, Canada, ACM, 2004.

LATHROP, A.; FOSS, K. Student Cheating and Plagiarism in the Internet Era. A Wake-Up Call. 2000. Libraries Unlimited, Inc., P.O. Box 6633, Englewood. 255p.

LEC POWER TRANSLATOR. Available at: <http://www.lec.com/power-translator-software.asp>. Visited on Oct. 31, 2008.

MALYUTOV, M. B. Authorship Attribution of Texts: A Review 2006. Springer. 362-380p.

MANNING, C. D.; RAGHAVAN, P.; SCHUTZE, H. Introduction to Information Retrieval. 2008. Cambridge, United Kingdom. 547p.

MAURER, H.; KAPPE, F.; ZAKA, B. Plagiarism - A Survey. In: **Journal of Universal Computer Science**, p.1050-1084, 2006.

MCCABE, D. L. Cheating among college and university students: A North American perspective. In: **International Journal for Educational Integrity**, 2005.

ORENGO, V. M.; HUYCK, C. R. Portuguese-English cross-language information retrieval using latent semantic indexing. In: **Cross-Language Evaluation Forum**, Roma, Italia, Heidelberg: Springer, 2002.

OUNIS, I., et al. Terrier Information Retrieval Platform. In: **Proceedings of the 27th European Conference on Information Retrieval (ECIR 05)**, 3408, p.517-519, Springer, 2005.

OUNIS, I., et al. Terrier: A High Performance and Scalable Information Retrieval Platform. In: **Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)**, Seattle, Washington, USA, 2006.

PAN 2009. Available at: <http://www.webis.de/pan-09>. Visited on Apr. 28, 2009.

PAN 2010. Available at: <http://www.uni-weimar.de/medien/webis/research/workshopseries/pan-10/>. Visited on Jun. 29, 2010.

PETERS, C.; FERRO, N. CLEF 2009 Ad Hoc Track Overview: TEL & Persian tasks. In: **Working Notes of CLEF 2009**, 2009.

PORTER, M. F. An algorithm for suffix stripping. In: **Readings in information retrieval**. 1997. Morgan Kaufmann. 313-316p.

POTTHAST, M. Wikipedia in the pocket: indexing technology for near-duplicate detection and high similarity search. In: **Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval**, Amsterdam, The Netherlands, p.909 - 909, ACM, 2007.

POTTHAST, M., et al. Cross-language plagiarism detection. In: **Language Resources and Evaluation**, 2010a.

POTTHAST, M., et al. An Evaluation Framework for Plagiarism Detection. In: **Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)**, Beijing, China, Association for Computational Linguistics, 2010b.

POTTHAST, M., et al. Overview of the 1st International Competition on Plagiarism Detection. In: **Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse**, San Sebastian (Donostia), Spain, p.1-9, 2009.

POULIQUEN, B.; STEINBERGER, R.; IGNAT, C. Automatic Identification of Document Translations in Large Multilingual Document Collections. In: **Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP'03)**, Borovets, Bulgaria, p.401-408, 2003.

QUINLAN, J. R. C4.5: programs for machine learning. 1993. Morgan Kaufmann. 302p.

RIVEST, R.L. The md5 message-digest algorithm. 1992. Available at: <http://theory.lcs.mit.edu/~rivest/rfc1321.txt>. Visited on Apr. 16, 2009.

ROIG, M. Avoiding plagiarism, self-plagiarism, and other questionable writing practices: A guide to ethical writing. 2010.

STEIN, B.; EISSEN, S. M. Z. Intrinsic Plagiarism Analysis with Meta Learning. In: **SIGIR'07 - Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection**, Amsterdam, 2007.

STEIN, B.; EISSEN, S. M. Z. Near Similarity Search and Plagiarism Analysis. In: **From Data and Information Analysis to Knowledge Engineering**. 2006. Springer. 430-437p.

TEXT RETRIEVAL CONFERENCE (TREC). Available at: <http://trec.nist.gov/index.html>. Visited on Aug. 19, 2009.

WEKA DATA MINING SOFTWARE. Available at: <http://www.cs.waikato.ac.nz/ml/weka/>. Visited on Sep. 01, 2009.

APPENDIX A - EXAMPLE OF THE WEKA ARRF FILE

We present here an example of the ARRF file created in order to be used during the training phase of the proposed method. This file is used as input to the classification algorithm. The ARRF file below contains ten training instances, where five of them represent information of a plagiarized passage and the other five represent information of a non-plagiarized passage.

```
@relation plagiarism-cases

@attribute length_A real
@attribute length_B real
@attribute rank real
@attribute score real
@attribute similarity real
@attribute isPlagiarism {true, false}

@data
432,422,1,28.6984,0.4923,true
474,365,1,32.1514,0.4074,true
400,383,1,37.1284,0.4674,true
449,264,1,25.7244,0.2551,true
463,349,1,22.6944,0.4190,true
422,251,3,12.7824,0.1690,false
422,328,4,14.5399,0.1322,false
514,296,1,15.5506,0.0787,false
514,413,2,14.4376,0.1487,false
514,335,3,11.3420,0.2144,false
...
```

According to the ARRF file above, it is possible to see the declaration of the five features used during the training phase:

- @attribute length_A: represents the length of the suspicious passage;
- @attribute length_B: represents the length of the candidate subdocument;
- @attribute rank: represents the position of the candidate subdocument in the IR system rank;
- @attribute score: represents the score assigned by the IR system to the candidate subdocument;
- @attribute similarity: represents the cosine similarity between the suspicious passage and the candidate subdocument;

It is also possible to see the definition of the feature that represents the class that must be predicted:

- @attribute isPlagiarism: indicates whether the data represents a plagiarized instance or a non-plagiarized one;

Note that after the @data keyword, the training instances are listed. Note that the value of each feature is separated by comma. For example, the first training instance contains the following information:

- The length of the suspicious passage is 432 characters;
- The length of the candidate subdocument is 422 characters;
- The candidate subdocument was in the first position of the IR system rank;
- The IR system assigned a score of 28.6984 to the candidate subdocument;
- The cosine similarity between the suspicious passage and the candidate subdocument is 0.4923;
- This is an instance that represents a true plagiarism case;

APPENDIX B - CONTRIBUIÇÕES

Este trabalho propõe e avalia um novo método para Análise de Plágio Multilíngue. O objetivo do método é detectar casos de plágio em documentos suspeitos baseado em uma coleção de documentos ditos originais (essa tarefa é conhecida como análise de plágio externo). A principal diferença do trabalho proposto em relação aos métodos existentes é a aplicação de um algoritmo de classificação para construir um modelo capaz de distinguir entre um trecho de texto plagiado e um trecho de texto não plagiado.

Uma vez que o método tem por objetivo detectar plágio entre documentos escritos em idiomas diferentes, uma ferramenta de tradução automática é utilizada para traduzir os documentos suspeitos e os documentos originais para um mesmo idioma (de forma que seja possível analizá-los de forma uniforme). Após a fase de normalização, um algoritmo de classificação é utilizado para construir um modelo capaz de diferenciar entre um trecho plagiado e um não plagiado. Para isso, um conjunto de *features* é selecionado para ser utilizado durante o treinamento do classificador. Ao término do treinamento, um sistema de Recuperação de Informações é utilizado para recuperar, baseado nos trechos extraídos dos documentos suspeitos, os documentos originais com maior chance de terem sido utilizados como fonte de plágio. Apenas após a recuperação desses documentos, uma análise detalhada de plágio é realizada. Ao final, o resultado é pós-processado a fim de combinar trechos plagiados contíguos.

Levando em consideração os passos descritos acima, o método proposto pode ser subdividido em cinco fases principais:

(1) *Normalização do Idioma*: nesta fase, tanto os documentos suspeitos quanto os documentos originais são traduzidos para um mesmo idioma.

(2) *Recuperação dos Documentos Candidatos*: nesta fase, trechos extraídos dos documentos suspeitos são utilizados para descobrir quais documentos originais são os mais prováveis de terem sido utilizados como fonte de plágio. Esta fase é muito importante já que não seria viável uma análise detalhada entre os documentos suspeitos e todos os documentos originais.

(3) *Seleção de Features e Treinamento do Classificador*: nesta fase, usando uma coleção de treinamento, um conjunto pré-definido de *features* é selecionado para construir o modelo de classificação. Baseado no classificador construído o método é capaz de decidir se um trecho suspeito é plagiado ou não.

(4) *Análise de Plágio*: nesta fase, cada trecho extraído dos documentos suspeitos é comparado com seu conjunto de documentos candidatos a fim de avaliar se o trecho suspeito é, de fato, plagiado.

(5) *Pós-Processamento do Resultado*: nesta fase, os trechos plagiados contíguos são combinados em um único trecho com o objetivo de reportar um trecho plagiado como um todo ao invés de diversos trechos plagiados pequenos.

Considerando o método descrito, assim como os passos necessários para sua validação, as principais contribuições deste trabalho são:

- A definição de um novo método para análise de plágio multíngue que emprega um algoritmo de classificação para construir um modelo capaz de distinguir entre uma trecho de texto plagiado e um trecho de texto não plagiado. Essa abordagem não tinha sido utilizada na análise de plágio externo até o momento.

- A avaliação do método utilizando três coleções de testes disponíveis. Isso possibilita que a performance de novos métodos possa ser comparada com a performance do método descrito nesse trabalho.

- A criação de uma coleção de teste especialmente construída para conter casos de plágio multilíngue. Essa coleção possibilita um meio de comparação entre diferentes métodos de análise de plágio multilíngue.