الجـــــــــمـهـوريـــــة الجـــــــــزائريـة الديمـــــــــــقراطيـة الشـــــــعبيـة
POPULAR DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التـــــــــــــعليـم العـــــــــالي و البـــــــــحث العـــــــــلمـي
Ministry of Higher Education and Scientific Research
جـــــــــــامعة غـــردايـــــــة
University of Ghardaia
كليـة العلـــوم و التكنولـوجيـا
Faculty of Science and Technology
قسم الرياضيات و الإعلام الآلي
Department of Mathematics and Computer Science

THESIS

Presented to obtain the **DEGREE** of **MASTER**

In : Computer Science

Speciality : Intelligent Systems for Knowledge Extraction

By BOUREKOUA Samia **And** OULAD ALI Bouchra

Subject

PLAGIARISM DETECTION

Publicly supported 28/06/2018 before the jury composed of :

Dr. Chaker Abdelaziz KERRACHE	MA	Univ. Ghardaia	President
Dr. Slimane OULAD NAOUI	MC	Univ. Ghardaia	Director of thesis
Dr. Slimane BELLAOUAR	MC	Univ. Ghardaia	Examinator
M. Messaoud BETKA	MA	Univ. Ghardaia	Examinator

College year 2017/2018

اهداء

أحمد الله عز وجل على منه وعونه لإتمام هذا العمل

الى من بلغ الرسالة وأدى الأمانة وعلم البشرية ونصح الأمة الى نبي الرحمة ونور العالمين

محمد صلى الله عليه وسلم

الى من قال الله فيهما عز وجل

واخفض لهما جناح الذل من الرحمة وقل ربي ارحمهما كما ربياني صغيرا سورة الاسراء الاية 21

الى الذي بفضل الله ثم بفضله اليوم أخط عبارات هذا الاهداء الى الذي ضحى بالغالي والنفيس من أجلي وعلمني معنى الكفاح

الى الذي سهر على تعليمي بتضحيات جسام مترجمة في تقديسه للعلم وكان نعم الاب الحنون والمعطاء الى القلب الكبير

" أبي الغالي" أطال الله في عمره

الى التي بحنانها وبفيض فؤادها رعتني وسهرت الليالي وأفنت شبابها لإسعادي الى التي صبرت على كل شيء الى القلب

الناصع البياض

" أمي الغالية " أطال الله في عمرها

الى سندي وقوتي بعد الله تعالى الى الشمعة المتقدة التي تنير ظلمة حياتي

الى من اثروني على أنفسهم اخوتي " فاطمة الزهراء، مبروكة، علي، فريال، عبد القادر"

الى مكمن البراءة أبناء اختي " هتون، فاروق "

الى اخي الذي لم تلده امي "العيد الحيلي" الى ينبوع الصفاء "خالتي عائشة"

الى من تحلو بالإخاء وتميزوا بالوفاء والعطاء الى من اعتز بالانتساب إليهم عائلتي " أولاد علي" خاصة " أعمامي

وعماتي"

الى مكمن العطف والحنان "خالي وخالاتي"

الى الروح الحية معلمي رحمه الله كنت خير معلم رسول "بن حويط علي"

الى توأم روحي ورفيقة دربي الى صاحبة القلب الطيب والنوايا الصادقة "مروة "

الى الاخوات اللواتي لم تلدهن أمي الى من تحلو بالإخاء وتميزوا بالوفاء والعطاء الى من معهم سعدت وبرفقتهم في دروب

الحياة الحلوة والحزينة سرت الى من كانوا معي في طريق النجاح والخير الى من عرفت كيف أجدهم وعلموني ان لا

أضيعهم صديقاتي

" ابتسام، حبيبة، سامية،وهيبة ,عزيزة, حنان ,ريم ,هدى , ايمان"

الى كل من تسعهم ذاكرتي ولم تسعهم مذكرتي، الى كل هؤلاء أهدي ثمرة جهدي

بشرى

Dedication

All praise to Allah, today we fold the days' tiredness and the errand summing up between the cover of this humble work.

To the utmost knowledge lighthouse, to our greatest and most honoured prophet Mohamed - May peace and grace from Allah be upon him

To the spring that never stops giving, to my mother and father who weaves my happiness with strings from their merciful heart...

To my husband to her help and understand a thousand thanks...

To my love, my children: ZAKI, SAFA, AROI, ROKAIA. God save them.

To my dear sister and dear brothers

To all my friends

To those who reworded to us their knowledge simply and from their thoughts made a lighthouse guides us through the knowledge and success path, To our honoured teachers and professors.

SAMIA

Acknowledgements

We would first like to thank our thesis advisor Dr Slimane Oulad Naoui, his door was always open whenever we ran into a trouble spot or had a question about our research or writing, He steered us in the right direction whenever he thought we needed it.

We would also like to thank the experts who were involved in the validation survey for this research project "Dr Bellaouar Slimane, Dr Kerrache Abd ElAziz, Mr Betka Messaoud".

We would also like to acknowledge all the teachers of MI in the University of Ghardaiia especially " Dr Bellaouar Slimane, Dr Kerrache, Mr Kechida Khaled, Mr Mahdjoub Youcef, Mr Bouhani Abdelkader,Mr Chabi Samir, Me Brahim Nacira" for theirs work with us every year.

Getting through our dissertation required more than academic support, and We have many, many people to thank for listening to and, at times, having to tolerate our over the past five years. we cannot begin to express our gratitude and appreciation for their friendship. have been unwavering in their personal and professional support during the time I spent at the University. For many memorable evenings out and in, we must thank everyone.

Finally, We must express our very profound gratitude to our parents our sisters and brothers and all families members,We are also grateful to our friends who have supported us along the way. for providing us with unfailing support and continuous encouragement throughout our years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Abstract

Nowadays, the use of computers, internet and digital technologies in general has increased remarkably. This situation leads to proliferation, simplicity, free access to digital documents and open databases on the Web. Plagiarism is the use of the work or the idea of someone else without acknowledgement, which is illegal especially in the research area. Consequently, the development of automatic plagiarism detection methods has gained a great interest over the last three decades. Data mining can help to build and improve the efficiency of the plagiarism detection systems thanks to its popular techniques such classification, clustering and outlier detection.
The aim of this work is to present in-depth study of principal plagiarism detection approaches. In the experimentation phase, report our JAVA implementation of an intrinsic plagiarism detection method based on character 3-gram. The obtained results show the usefulness of the method to recognize plagiarized passages. However, in the cases of huge corpora, its efficiency should be further studied.

Keywords: Plagiarism detection, Intrinsic plagiarism, Data mining, Similarity.

Résumé

De nos jours, l'utilisation des ordinateurs, de l'Internet et des technologies numériques en général a considérablement augmenté. Cette situation a entraîné la prolifération, la simplicité, et l'accès gratuit aux documents numériques et bases de données ouvertes sur le Web.
Le plagiat est l'utilisation du travail ou l'idée de quelqu'un d'autre sans Confession, ce qui est illégal surtout dans le domaine académique. Par conséquent, le développement de méthodes de détection automatique du plagiat a gagné un grand intérêt au cours des trois dernières décennies. La fouille de données peut aider à construire et améliorer l'efficacité des systèmes de détection de plagiat grâce à ses techniques populaires telles la classification, la segmentation et la détection des anomalies.
Le but de ce travail est de présenter une étude approfondie des principales méthodes de détection du plagiat. Dans la phase d'expérimentation, nous rapportons notre implémentation JAVA d'une méthode de détection intrinsèque du plagiat intrinsèque basée sur le caractère 3-gram. les résultats obtenus montrent l'utilité de la méthode pour reconiser les passages plagiés. Cependant, dans le cas d'énormes corpus, son efficacité devrait être reconsidérée.

Mots clés: Detection de plagiat, Plagiat interne , Fouille de données, Similarité

ملخص

على ضوء التقدم التكنولوجي وانتشار الاجهزة الرقمية والانفجار المعرفي الكبير الذي عرفه العالم المعاصر، مما أدى إلى سهولة و مجانية الوصول للمستندات الرقمية وقواعد البيانات المفتوحة على الويب. ظهر الانتحال و هو استخدام عمل أو فكرة شخص آخر دون اعتراف، وهذا أمر غير قانوني خاصة في مجال البحث العلمي. و لذلك اكتسب تطوير طرق الكشف التلقائي عن الانتحال اهمية كبيرة على مدى العقود الثلاثة الماضية. يمكن أن يساعد التنقيب عن البيانات في بناء وتحسين كفاءة أنظمة الكشف عن الانتحال وذلك بفضل تقنياته الشائعة مثل التصنيف والتجميع والكشف المبكر. الهدف من هذا العمل هو تقديم دراسة معمقة لأساليب الكشف الرئيسية عن الانتحال. في المرحلة التجريبية، قمنا بانجاز تطبيق باستعمال لغة البرمجة جافا والخاص بطريقة من طرق الكشف عن الانتحال الداخلي باستعمال ثلاثة أحرف، أظهرت النتائج المتحصل عليها فائدة الطريقة المستعملة للتعرف على الاجزاء المنتحلة، ومع ذلك في حالة مجموعة المعطيات الضخمة ينبغي اعادة النظر في فاعليتها.

الكلمات المفتاحية التنقيب في البيانات، الكشف عن الانتحال، الانتحال الداخلي، التشابه.

Contents

List of Figures

List of Tables

Introduction

The explosion in the information and technology of our modern life and its consequences in a lot of fields makes a revolution in getting of data with easier way, especially with the web; which is a great situation for spread of knowledge to everyone, but also added the cases of fraud and theft of information.

As survey of 43,000 high school students conducted in 2010 by the Josephson Institute [48] found that:" 59% admitted to cheating within the past year, with 34% doing it twice or more". A study by Bowers, as cited in McCabe et al.[16] in 1964 found that 75% of college students cheated. This case leads to kill the work hard and creativity.

Plagiarism is the use of the work or the idea of someone else without acknowledgement. Usually, an act of plagiarism could be recognized manually by relying on human cognition on the seemingly-similar texts or on the writing style that changes clearly. However, this kind of recognition demands an intense memory on all articles, book and any other types of writings which have been read. Another requirement is that the process of reading should have occurred recently. Otherwise, it would be forgotten.

With the incredible improvement on the computer network and a large amount of source material available on the Web, the task of recognizing Plagiarism is beyond the reach of all human knowledge. To make it worse, prove a job as an act of plagiarism requires proof of source documents. This situation gives We need automatic detection of plagiarism, which motivated us to study this approach.

Plagiarism is very often done intelligently, for example by paraphrasing or obscuring the texts so that only a small part of the document is found to be similar.
The detection of plagiarism can be done as extrinsic which compare a suspicious document against a collection of references, whereas intrinsic which do not need any resources just the suspicious document because it is based in chunking the original document and study its style writing variation. Intrinsic case is more difficult when extrinsic is the most popular.

The detection of plagiarism is essentially a similarity based task, because in the two cases we are interested in finding similar/dissimilar document segments. So, plagiarism detection can be seen as a trivial data mining and machine learning tasks, such as classification, clustering and outlier detection.

The thesis is organized as follows: in the first chapter we give an overview of the

principle concepts of data mining, its tasks, techniques and some application fields. We deal in the second chapter with plagiarism in detail, its taxonomy, detection, some tools, methods and evaluation. In the third chapter we describe the related work of the famous extrinsic and intrinsic plagiarism detection methods. We present in the last chapter our implementation, we describe the corpus [PAN09] then present the chosen method which is about intrinsic plagiarism detection case.

Chapter 1

Data Mining

The growing power of technology and deal of data sets perform data mining to improvement from tapes and disks to advanced algorithms and large databases. Data mining has a long history, early Data Mining methods Bayes' Theorem and Regression analysis which were mostly identifying patterns in data, that were the first start. The motivation was the richness of data, another hand information poor.

In the 19^{th} century, data mining began to be recognized and used in the research community by statisticians, data analysts, and the management information systems (MIS) communities. By the end of 1990's, data mining was already a famous technique used by the organizations after the introduction of customer loyalty cards. This opened a big door allowing organizations to record customer purchases and data, the resulting data could be mined to identify customer purchasing patterns. The popularity of data mining has continued to grow rapidly over the last years in many areas.

In this chapter we present a brief overview about the high line of Data Mining and its tasks.

1.1 Basic Definitions

Knowledge Discovery in Databases

Knowledge Discovery in Databases (KDD) is " The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" [20].

Data Mining

" Is a step in the Knowledge Discovery in Databases (KDD) process that consists of applying data analysis and discovery algorithms that produce a particular enumeration of

patterns (or models) over the data."[20]

Data Mining is related with a lot of fields like Artificial Intelligence, Statistics, Machine Learning, Databases and data warehousing, High performance computing Visualization, etc.

Data Warehouse

" Data Warehouse is a repository of information collected from multiple sources, stored under a unified schema, and that usually resides at a single site. Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and periodic data refreshing." [28].

Machine Learning

" Machine learning investigates how computers can learn (or improve their performance) based on data. A main research area is for computer programs to automatically learn to recognize complex patterns and make intelligent decisions based on data." [28]
the types of machine learning which are very related with data mining are: Supervised and Unsupervised learning.

Supervised Learning

" Is basically a synonym for classification. The supervision in the learning comes from the labeled examples in the training data set. " [28].

Unsupervised Learning

" Is essentially a synonym for clustering. The learning process is unsupervised since the input examples are not class labeled. Typically, we may use clustering to discover classes within the data. " [28].

1.2 KDD Process

The knowledge discovery process is shown in Figure of page 6 as an iterative sequence of the following steps [28]:

1. **Data cleaning** : also known as data cleansing, it is a phase in which noise data and irrelevant data are removed from the collection.

2. **Data integration** : at this stage, multiple data sources, often heterogeneous, may be combined in a common source.

3. **Data selection** : at this step, the data relevant to the analysis is decided on and retrieved from the data collection.
4. **Data transformation** : also known as data consolidation, it is a phase in which the selected data is transformed into forms appropriate for the mining algorithm.
5. **Data mining** : an essential step where intelligent methods are applied to extract data patterns. It is the core of the KDD process.
6. **Pattern evaluation** in this step, strictly interesting patterns representing knowledge are identified based on given measures.
7. **Knowledge presentation** : is the final phase in which the discovered knowledge is visually represented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results.

1.3 Data Mining Tasks

Data mining tasks can be classified into two categories[20]: predictive and descriptive; Descriptive mining tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make predictions.

1.3.1 Classification

Classification is a supervised learning task that consists of assigning, according to their properties, a collection of objects to a set of predefined classes [50].

For example personal email sorting. A user may have folders like talk announcements, electronic bills, email from family and friends, and so on, and may want a classifier to classify each incoming email and automatically move it to the appropriate folder. It is easier to find messages in sorted folders than in a very large inbox. The most common case of this application is a spam folder that holds all suspected spam messages.

1.3.2 Prediction

Task predicts the possible values of missing or future data. Prediction involves developing a model based on the available data and this model is used in predicting future values of a new data set of interest [28]. For example, a model can predict the income of an employee based on education, experience and other demographic factors like place of stay, gender etc. Also prediction analysis is used in different areas including medical diagnosis, fraud detection etc.

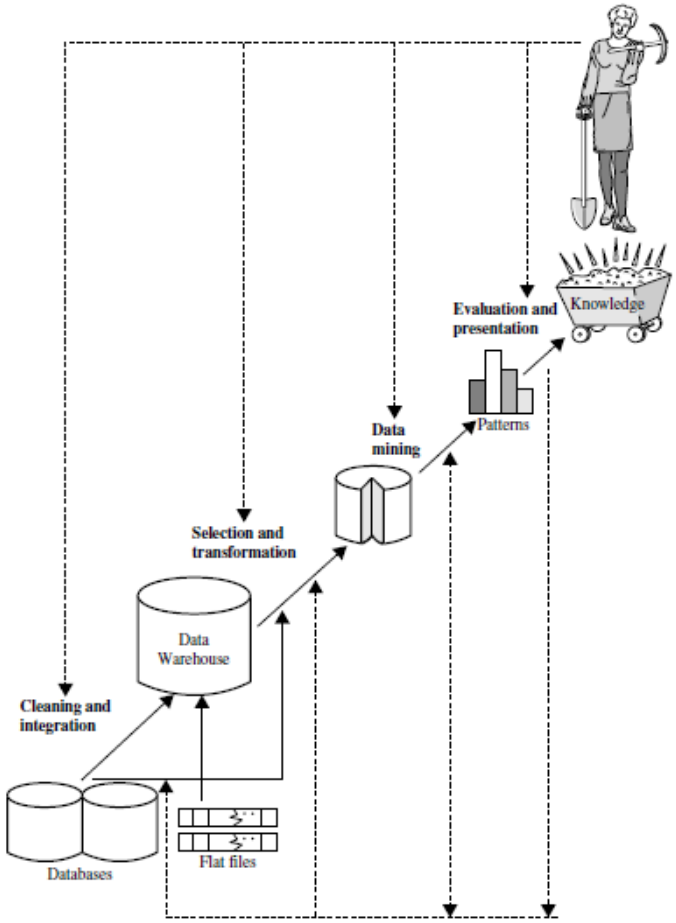

Figure 1.1: Data mining as a step in the process of knowledge discovery [28]

1.3.3 Association Rule

The discovery of correlation or connection of objects, such kind of correlation or connection is termed as *association rule*. An association rule reveals the associative relationships among objects, i.e., the appearance of objects in a database is strongly related to the appearance of another set of objects. For example, in a telecommunication database, an association rule that " call waiting" is associated with " call display", denotes as " call waiting $\rightarrow$ call display", says if a customer subscribes to the " call waiting" service, he or she very likely also has " call display" [28].

1.3.4 Clustering

Is a common descriptive task, it deals with finding structure in a collection of unlabelled data [28].(see figure 1.2), is a process which partitions a given data set into homogeneous

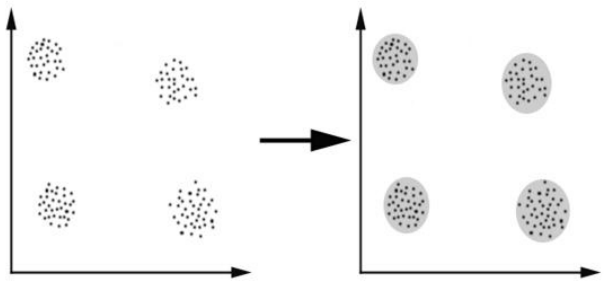

Figure 1.2: four clusters formed from the set of unlabeled data [28].

groups based on given features such that similar objects are kept in a group, Whereas dissimilar objects are in different groups

1.3.5 Outlier Analysis

An outlier is a data object that deviates significantly from the normal objects [29] as if it were generated by a different mechanism.
Outlier analysis is used in various types of dataset, such as graphical dataset, numerical dataset, Text dataset, and can also be used on the pictures etc. The identification of outlier can lead to the discovery of useful and meaningful knowledge. Finding outliers from a collection of patterns is a popular problem in the field of data mining. A key challenge with outlier analysis and detection is that it is not a well formulated problem like clustering so outlier detection as a branch of data mining requires more attention.

1.3.6 Summarization

Is the generalization of data. A set of relevant data is summarized which result in a smaller set that gives aggregated information of the data[21]. For example, the shopping done by a customer can be summarized into total products, total spending, offers used, etc. Such high level summarized information can be useful for sales or customer relationship team for detailed customer and purchase behavior analysis. Data can be summarized in different abstraction levels and from different angles.

Different data mining tasks are the core of data mining process. Different prediction and classification data mining tasks actually extract the required information from the available data sets.

1.4 Data Mining Techniques

Data mining combine different techniques and algorithms from various disciplines. Here we will present some algorithms which are used in plagiarism detection.

Supervised learning algorithms

k-Nearest Neighbors, *Support vector machine (SVM)*, *Naive Bayes*, *Random forest*, *Decision trees*, *Neural Networks etc.*
All classification and regression algorithms come under supervised learning.

- *Support Vector Machine (SVM)*: SVM is one of the powerful techniques for classification, regression and outlier detection with an intuitive model. Support Vector Machines [28] are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. There are two kinds of SVM classifiers:

  1. *SVM Linear Classifier:* In the linear classifier model, we assume that training examples are plotted in space. These data points are expected to be separated by an apparent gap. It predicts a line dividing two classes. The primary focus while drawing the line is on maximizing the distance from line to the nearest data point of both class. The drawn line is called a maximum-margin hyperplane, as shown in figure 1.3 [14].
  2. *SVM Non-Linear Classifier:* In the real world, our dataset is generally dispersed up to some extent. To solve this problem separation of data into different classes on the basis of a straight linear hyperplane can not be considered a good choice. For this, Vapnik [70] propose Non-Linear Classifiers by applying the kernel trick[1] to maximum-margin hyperplanes. In Non-Linear SVM Classification, data points are plotted in a higher dimensional space, Example figure 1.4.

  The positive points of The support vector machine produce very accurate classifiers and less over-fitting, robust to noise.

  On the other hand SVM is a binary classifier. To do a multi-class classification, pairwise classifications can be used (one class against all others, for all classes). and computationally expensive, thus runs slow.

- k-Nearest Neighbors: Is one of the elementary supervised classifiers, based on stores all available cases and classifies new cases established on a similarity measure [28]. The k-nearest-neighbor is an example of a lazy learner algorithm , meaning that it does not build a model using the training set until a query of the data set is performed.The general steps are showing in Algorithm 1.

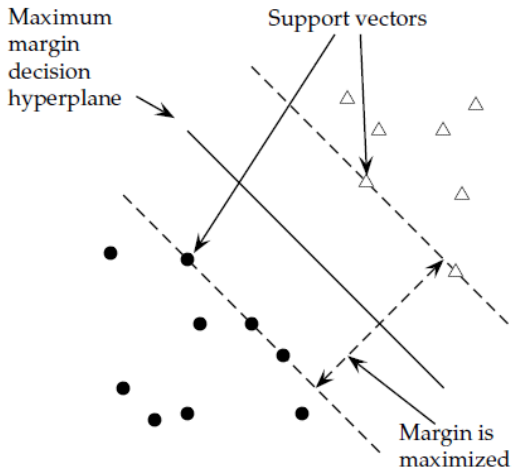

Figure 1.3: The linearly separable case [14].

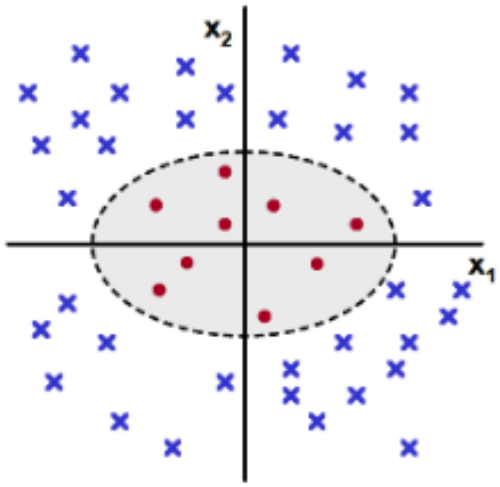

Figure 1.4: Non-linear separable case [14].

Algorithm 1 k-Nearest Neighbors

1: *Classify* (**X**, **Y**, s) //**X**: training data, **Y**: class labeles of **X**, s: unknown sample.
2: **for** *i=1* **to** n **do**
3: *Compute distance* $d(X_i, s)$
4: **end for** .
5: *Compute set* I *containing indices for the* k *smallest distances* $d(X_i, s)$
6: **return** *majority label for* $\{Y_i$ *where* $i \in I\}$

K-nearest neighbors need not neither prior knowledge about the structure of data in the training set, nor required retraining if the new training pattern is added to the existing training set. But when the training set is large, it may take a lot of space. and for every test data, the distance should be computed between test data and all the training data, thus a lot of time may be needed for the testing.

- Naïve Bayes Classifier: is a popular algorithm in machine learning, it is particularly useful for textual data analysis [14]. It works on conditional probability. Conditional probability is the probability that something will happen, given that something else has already occurred. using the conditional probability, we can calculate the probability of an event using its prior knowledge. It considers all the features to be unrelated, so it cannot learn the relationship between features.

- Neural Networks (NN): [26] are a class of systems modeled after the human brain. As the human brain consists of millions of neurons that are interconnected by synapses, neural networks are formed from large numbers of simulated neurons, connected to each other in a manner similar to brain neurons. Like in the human brain, the strength of neuron interconnections may change (or be changed by the learning algorithm) in response to a presented stimulus or an obtained output, which enables the network to learn.

Unsupervised learning algorithms

All clustering algorithms come under unsupervised learning algorithms [71].
K – means clustering, Hierarchical clustering, Hidden Markov models

❶ *K – means clustering:* is one of the simplest unsupervised learning algorithms that solve the well known clustering problem [17]. The idea is to classify a given set of data into k number of disjoint clusters, where the value of k is fixed in advance. The algorithm consists of two separate phases: the first phase is to define k centroids, one for each cluster. The next phase is to take each point belonging to the given data set and associate it to the nearest centroid. Euclidean distance is generally considered to determine the distance between data points and the centroids. When all the points are included in some clusters, the first step is completed and an early grouping is done. At this point we need to recalculate the new centroids, as the inclusion of new points may lead to a change in the cluster centroids. Once we find k new centroids, a new binding is to be created between the same data points and the nearest new centroid, generating a loop. As a result of this loop, the k centroids may change their position in a step by step manner. Eventually, a situation will be reached where the centroids do not move anymore. This signifies the convergence criterion for clustering.
Pseudocode for the k-means clustering algorithm is listed as Algorithm 2 [17].

Algorithm 2 K-means clustering

1: **Input**: *k(the number of clusters),D(a training set)*
2: **Output**: *a set of K clusters*
3: **Method**: *Randomly choose k clusters*
4: **Repeat**:
5: *(re)assign each object to the cluster to which it is the most similar, based on the mean value of the object in the cluster.*
6: *update the cluster means (calculate the mean value of the object for each cluster)* .
7: **Until** *no change*.

This algorithm generally effective in producing good results. The major issue is that it produces different clusters for different sets of values of the initial centroids. Quality of the final clusters heavily depends on the selection of the initial centroids. The k-means algorithm is computationally expensive and requires time proportional to the product of the number of data items, number of clusters and the number of iterations.

1.5 Data Mining Applications

Data Mining uses in various areas [28] including Market Basket Analysis, Health Care and Insurance, Bio-Informatics, Education, Manufacturing Engineering and Research analysis, etc,.

Market Basket Analysis

Market basket analysis is a modelling technique based upon a theory that if you buy a certain group of items you are more likely to buy another group of items. This technique may allow the retailer to understand the purchase behaviour of a buyer. This information may help the retailer to know the buyer's needs and change the store's layout accordingly. Using differential analysis comparison of results between different stores, between customers in different demographic groups can be done.

Health Care and Insurance

Data mining holds great potential to improve health systems. It uses data and analytics to identify best practices that improve care and reduce costs. Researchers use data mining approaches like multi-dimensional databases, machine learning, soft computing, data visualization and statistics. Mining can be used to predict the volume of patients in every category. Processes are developed that make sure that the patients receive appropriate

care at the right place and at the right time. Data mining can also help healthcare insurers to detect fraud and abuse.

Bio-Informatics

Data Mining approaches seem ideally suited for Bioinformatics, since it is data-rich. Mining biological data helps to extract useful knowledge from massive datasets gathered in biology, and in other related life sciences areas such as medicine and neuroscience. Applications of data mining to bioinformatics include gene finding, protein function inference, disease diagnosis, disease prognosis, disease treatment optimization, protein and gene interaction network reconstruction, data cleansing, and protein sub-cellular location prediction.

Education

There is a new emerging field, called Educational Data Mining, concerns with developing methods that discover knowledge from data originating from educational Environments. The goals of EDM are identified as predicting students' future learning behaviour, studying the effects of educational support, and advancing scientific knowledge about learning. Data mining can be used by an institution to take accurate decisions and also to predict the results of the student. With the results the institution can focus on what to teach and how to teach. Learning pattern of the students can be captured and used to develop techniques to teach them.

Manufacturing Engineering

Knowledge is the best asset a manufacturing enterprise would possess. Data mining tools can be very useful to discover patterns in complex manufacturing process. Data mining can be used in system-level designing to extract the relationships between product architecture, product portfolio, and customer needs data. It can also be used to predict the product development span time, cost, and dependencies among other tasks.

Research analysis

History shows that we have witnessed revolutionary changes in research. Data mining is helpful in data cleaning, data pre-processing and integration of databases. The researchers can find any similar data from the database that might bring any change in the research. Identification of any co-occurring sequences and the correlation between any activities can be known. Data visualisation and visual data mining provide us with a clear view of the data.

The challenges in data mining are efficient and effective data mining in large databases poses numerous requirements and great challenges to researchers and developers,The issues involved include data mining methodology, user interaction, performance and scalability, and the processing of a large variety of data types. Other issues include the exploration of data mining applications and their social impacts.

1.6 Conclusion

Data mining is the task of discovering interesting patterns from large amounts of data, where the data can be stored in databases, data warehouses, or other information repositories. It is a young interdisciplinary field, drawing from areas such as data warehousing, statistics, machine learning, etc.

In this chapter, we gave a whole idea about data mining which is play an interest case in data science. In the next chapter, we will see the concept of Plagiarism Detection.

Chapter 2

Plagiarism

Nowadays, technology offers the facility to copy text and other kinds of information easier than ever before. The advancement of information technology and more particularly the Internet augmented the availability of information considerably. The unspecified use of original work is considired as one of the major problems in the modern time. Reason for which automatic methods for the detection of plagiarism were developed playing the part of a possible counter measure. In this chapter, we focus on the basic concepts of plagiarism in order to understand this problem.

2.1 Definition

According to American Association of University Professors, plagiarism is defined as follwing: "Taking over the ideas, methods, or written words of another, without acknowledgment and with the intention that they be taken as the work of the deceive"[59] .
Lancaster in[38] state that "Plagiarism describes the process of using the words or ideas of another without suitable acknowledgement". There are several methods of plagiarising, some of them include[42]:

- Copy – paste plagiarism: copying word to word textual information which the text content is copied from one or several sources.
- Paraphrasing: change grammar, use synonyms of words, reorganization of sentences of original work and finally the removal of some parts of the text.
- Translated plagiarism: content translation and use without reference to original work.
- Artistic plagiarism: presenting same work using different media: text, images ..etc.
- Code plagiarism: using program codes without permission or reference.

- No proper use of quotation marks: the exact parts of take contents are not identified.
- Misinformation of references: reference is addition to incorrect or non existing source.
- Idea plagiarism: using similar ideas which are not common knowledge, it is difficult to detect this kind of plagiarism because it is more advanced than the other.
- self-plagiarism: an author uses his own published work in a new research paper for publication.
- Ignored or expired links to resources: the quotations or reference marks are addition but failing to give information or up-to-date links to sources.

2.2 Plagiarism Taxonomy

We can classify the plagiarism into different ways. Based on the type of person who is excuting the plagiarism, plagiarism can be academic or profissional[38]

- **Academic plagiarism**: it accurate where an academic such as student, researcher uses a plagiarized work for professional development, for example presenting plagiarized papers to conferences or journals
- **Professional plagiarism**: refers to plagiarism in workspace, like copying a report from a concurrent.

An other classification can appear based on types of material being examined:

- **Source code plagiarism**: refers to plagiarism in programs written in languages such as Delphi and Java.
- **Free text plagiarism**: refers to plagiarism in text written in natural language such as Arabic.

Based on the plagiarist's behavior (i.e. student's or researcher's way of executing plagiarism), We can divide plagiarism into two typical types[3]: literal plagiarism and intelligent plagiarism in Figure 2.1.

- **Literal Plagiarism**: in this kind, the plagiarists makes an exact copy/paste of the text from internet for example (exact copy), or does a few alteration near copy (insertion, deletion, substitution), or modified copy (phrase reordering, syntax) and don't lose the time in the modifications to hide the crime of plagiarism.
  " Any verbatim text taken from another source must be enclosed in quotation marks and be accompanied by a citation to indicate its origin"[59]

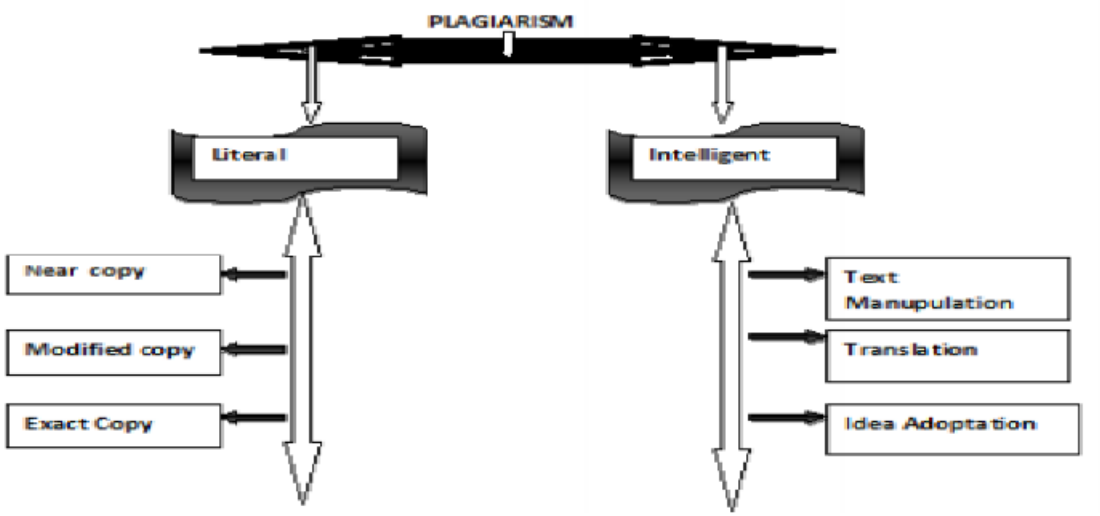

Figure 2.1: Taxonomy of Plagiarism [46]

- **Intelligent Plagiarism**: plagiarists try to hide and modify the original work in various ways to appear as their own, including text manipulation, translation and adoption of ideas[46].

  - Text Manipulation: changes appeared in the document, as the replacement of the word by their synonyms, antonyms, sentences are restructured, small sentences are added, summarizing the text in a shorter form using sentence reduction. the idea remains unchanged
  - Translation: text is translated from one language to another, with no suitable referencing to the source. translation can be automatic or manual.
  - Idea Adoption: Idea copying refers to the employ of others thoughts, such as results, contributions, and conclusions, without citing the source of thoughts. It is a main crime to take others thoughts.

These are not the only types of plagiarism possible, we can see plagiarism of music, images, video,..etc. In ours studies, we are interested in textual plagiarism.

2.3 Plagiarism Detection Types

Based on the homogeneity or heterogeneity of the language of the compared textual documents, the detection of plagiarism can be divided into two basic types: monolingual and multilingual[3].

1. **Mono-Lingual Plagiarism detection** : the automatic identification and extraction of plagiarism in a homogeneous language setting refers to Mono-lingual plagiarism detection . e.g. English-English plagiarism. Most detection methods are of this category.

2. **Multi-Lingual Plagiarism detection** : in this approach, a text fragment wich is created in a language is considered a plagiarism of a text in another language if its content is reused[4].

According to Potthast et al.[57], we can divides plagiarism detection into two main tasks :

1. **Intrinsic Plagiarism detection**

   In this case, the plagiarism detection is made without any reference collection; It handles the same document. Consequently, The emergence of this type of plagiarism discovery is strongly related to verification of copyright, and can be considered as generalization of authorship verification and attribution [37]. Unlike intrinsic plagiarism detection, an authorship verification is given some parts of writing examples of an author, for example author A, and its task is to determine whether or not a text is written by A. In term of similarities and differences between intrinsic plagiarism detection and authorship verification, $Halvani$ in [27] summarizes that intrinsic plagiarism detection is not addressing who the writer is as authorship verification, but rather the suspicious sections. Besides, the context for intrinsic plagiarism detection and authorship verification is different, but they share slightly similar technical background.

   Eissen and Stein[75] was the first who introduced the concept of intrinsic plagiarism. because the effect of plagiarism approach depends on quality of the quantified linguistic features, they introduce features that can be used to determine a respectable part of style information.
   The main idea of intrinsic plagiarism detection consists of dividing a document into natural parts (sentences, paragraphs or sections), then analysing the style variation, known as stylometry analysis.
   Stylometric features quantify style aspects, which have been succefully implemented to distinguish texts with respect to authorship in the past [63]. The categories for stylometric features can be constructed that quantify the characteristic trait of an author's writing style as Text statistic (invest at the character level), Syntactic features, Part-Of-Speech (quantify the word class.), Closed class word (count special words: difficult word, number of stop word, foreign words), Structural features (examine text organisation: chapter lengths, paragraph lengths).

   In order to work out these features more systematically, Potthast et al [54] define a building block of intrinsic plagiarism detection into four stages which comprise

chunking strategy, writing style retrieval model, an outlier detection algorithm and post-processing. *The chunking strategy* defines a boundary for feature extractions. The chunk length should be chosen in approximately equal size [27], otherwise it would influence the accuracy of the final result. *The retrieval model* is a model function that maps feature representations and their similarity measure. *The outlier detection* attempts to identify chunks that are noticeably different from the rest. This is done either by measuring the deviation from the average document style or chunk clustering[54]. *Post-processing* merged overlapping and consecutive chunks that have been identified as outliers[54].

In comparison with external plagiarism detection, intrinsic plagiarism detection is more difficult [27] since there is no available reference document except the suspicious one. This leaves no further possibilities to uncover plagiarism case except to detect suspicious sections, and even if suspicious sections are found, there is still no guarantee that these sections are truly plagiarized [27]. But the emergence of intrinsic plagiarism detection approach is to anticipate a case where the reference material is not always available or the amount of reference is very large [75]. This makes intrinsic plagiarism detection approaches increasingly important. The figure 2.2 illustrates this concept.

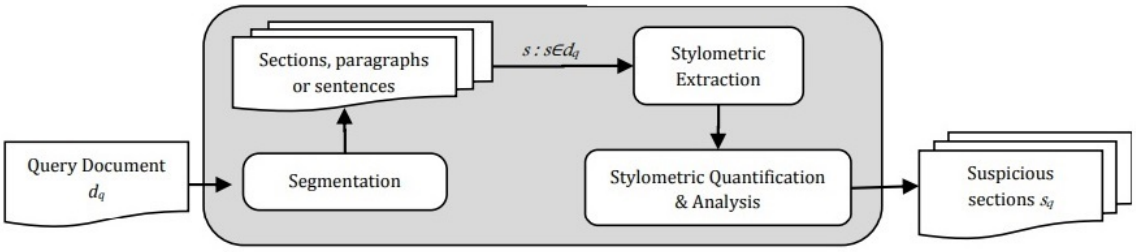

Figure 2.2: intrinsic plagiarism detection [9]

2. **Extrinsic Plagiarism detection**: plagiarism is evaluated by extrinsic plagiarism detection referencing to one or more source document in the corpus. the computer capacity is used by this task in order to find similar documents inside a corpus and retrieve plagiarized document. Most external plagiarism detection system follow a three-stage retrieval process as illustrated in Figure 2.3[66].

   - Heuristic retrieval: the goal of this stage is to reduce the number of comparisons between suspicious document and source documents when there is a large collection of source documents. the output of this step is condidate documents

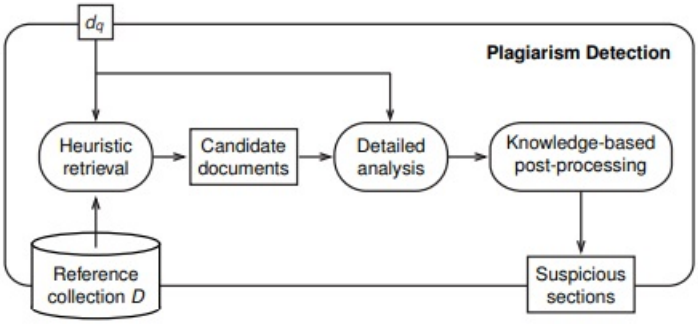

Figure 2.3: Generic retrieval process for external plagiarism detection [66]

these probably conten plagiarized fragements. most of External Plagiarism Detection apply fingerprinting or sub-string matching in this stage[22]

- Detailed Analysis: The aim of this stage is to identify pairs of the possibly similar passages and to discard the rest of passages that are highly dissimilar[66].
- Knowledge-based Post-processing: It is analyzed whether the same passages identified in the previous step have been properly quoted[66].

2.4 Avoiding Plagiarism

To avoid or minimize the risk of plagiarism, two methods exist: Plagiarism prevention and Plagiarism Detection [33]

2.4.1 Plagiarism Prevention

In order to avoid plagiarism must create a collaborative effort to recognize and control the fight against plagiarism at all levels, It is necessary to create laws that punish the plagiarists, students should be educated about the appropriate use and admission of all forms of intellectual material, possibility of representation ,finely, efficient procedures are installed for monitoring and detecting plagiarism.
In fact, Plagiarism prevention is not easy to achieve and takes a long time to implant in but its effects are long term [33]

2.4.2 Plagiarism Detection

Plagiarism detection can be excuted manually or automaticly with the help of software, manually detection is hard and slow unlike automatic detection which is easier, simpler

and faster to detect. Culwin and Lancaster[15] define a four stage process for detecting plagiarism which are shown in figure 2.4

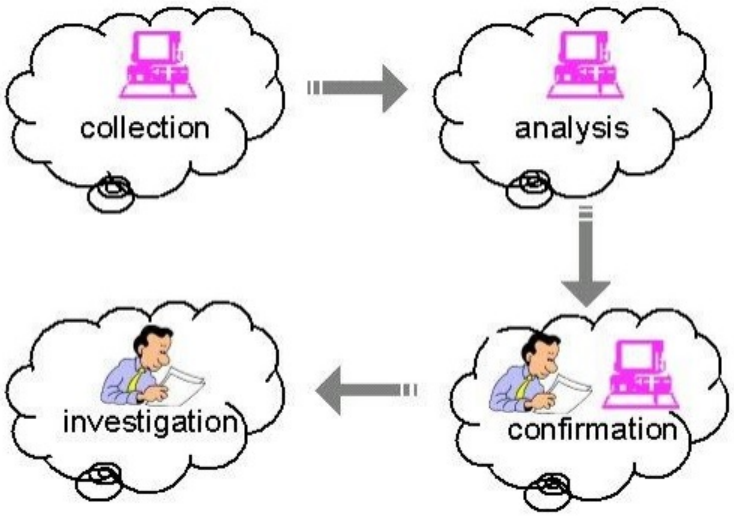

Figure 2.4: Process of detecting plagiarism [15]

- Collection stage: In this stage, assignments or works of student or researcher are uploaded to the web engine, the web engine proceeds as an interface between the students and the system

- Analysis stage: the corpus of submitted documents is run through a computerized similarity engine that produces some sort of measure of which submitted documents are potentially plagiarized. The effectiveness of the detection will depend on the methods used in this stage.

- Verification (confirmation) In this step, the man enters for decision if the similarity reported represents plagiarism or not. Any similarity regarded as plagiarism is examined further at the investigation stage.

- Investigation stage, separates the detection of similarity from any decisions about the impact of this similarity. two important things are considered when automating the process, first, the efficiency of the algorithm used in searching similar submission, Second, is the time which is used in the confirmation and investigation stages and how this could be reduced

As we saw earlier in the plagiarism taxonomy, plagiarism can be textual or code source. the way in which the textual features are used to characterize the documents is the principle of classification textual plagiarism detection[13]. In the following section, we focus on textual features used in extrinsic, intrinsic, and multi-lingual plagiarism detection.

2.5 Textual Features

There are different textual features like lexical feature, syntactic feature, semantic feature and structural feature, which can be used to detect similarity between two documents[3].

2.5.1 Lexical features

Lexical features work at the character or word level. Character-based n-gram (CNG) represents document as sequences of n characters, the same of word-based n-gram(WNG)where document is represented as sequences of n words.
lexical features are used in both extrinsic, intrinsic plagiarism detection methods, but it not used in multi-language plagiarism detection methods.

2.5.2 Syntactic features

In extrinsic plagiarism detection methods, the document is represented as chunks, sentences, phrases, and part of speech(POS), Basic POS tags (POS tagging[1][3].) include verbs, nouns, pronouns, adjectives, adverbs, prepositions, conjunctions and interjections. In intrinsic plagiarism detection methods, syntactic features work at the sentence level.

2.5.3 Semantic features

In extrinsic plagiarism, Semantic features operate at the level of word classes, synonyms, antonyms, hypernyms, and hyponyms. Thesaurus dictionaries and lexical databases, WordNet[2] are used to give more clarity in the semantic meaning of the text. also POS tagging is used.
In intrinsic plagiarism, synonyms, hypernyms,.. etc, functional words, and/or semantic dependencies are used.
In cross-linguage plagiarism detection methods, syntactic features are usually combined with semantic or statistical features, table3.3 refer to textual features for that kind of methods.

[1]POS tagging is the task of marking up the words in a text or more precisely in a statement as corresponding to a particular POS tag

[2]WordNet is an online lexical database. English nouns, verbs, adjectives, and adverbs are organized into sets of synonyms called synsets ...

	Exemples	Required Tools and Resources
Syntactic features	Word n-gram(1-gram)	Tokenizer
	Chunks/fragment	Tokenizer,[Sentence splitter,POS tagger], Text chunker(Windowing)
	Word positions	Tokenizer, Sentence splitter, Compressor (e.g. Lempel-Zif)
	Part-of-speech and phrase structure	Tokenizer, Sentence splitter, POS tagger
	Sentence	Tokenizer, Sentence splitter, POS tagger, Text chunker, Partial parser
Semantic features	Synonyms, hypernyms,etc	Tokenizer, [POS tagger], Bilingual thesaurus
	Semantic dependencies	Tokenizer, Sentence splitter, POS tagger, Text chunker,Partial parser, Semantic(bilingual) parser
Statistical features	Language-specific	Tokenizer, [Stemmer, Lemmatizer], Statistical(bilingual), dictionaries,Machine translators

Table 2.1: Types Of Cross-Language Text Features with Computational Tools Required for thier Implementation [3].

2.5.4 Structural features

With extrinsic plagiarism detection methods, structural features might characterize documents as headers, sections, sub-sections, paragraphs, sentences..etc. HTML web-pages and XML files are example for structural document which is used by this type of features.

2.6 Plagiarism Detection Methods

According to the work of Maurer et al.[42], methods of Plagiarism detection can be categorized into three main categorizes :

- The first tries to detect the style writing and find the inconsistent changes to this style.

- The second category is the most used. It is based on the comparison between several documents and identification of overlapping parts between these documents

- The third category receives an input document and then search the suspicious passage in the Web manually or in an automated way.

According to[44], plagiarism detection methods can be classified by type of similarity assessment as shown in figure 2.5. The leaves of the tree present the document models that the methods typically use for comparing documents
In local similarity assessment methods, the analysis of matches achieved on restricted

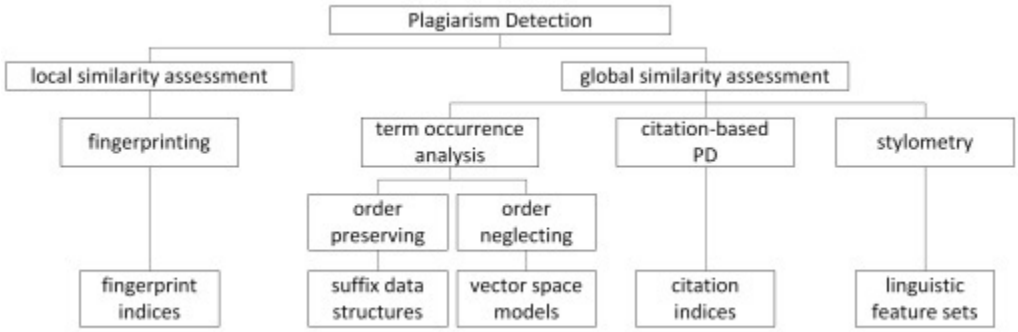

Figure 2.5: Classification of plagiarism detection methods [44]

text segments, On the other hand, in global similarity assessment methods, the analysis of characteristics achieved on Longer Text or the full document [44].

2.6.1 Fingerprinting

A document fingerprint is a set of integers wich represents some key content of document. each of these integers is called minutia, In order to generate fingerprint, the text is devised into substrings (chunks) and a hashing function is applied on each selected subsring wich produce one minutia, an index of minutia is created for quick access when querying, the comparison is realised on the fingerprint rather than the whole text [60]. The figure 2.6 presents an example of fingerprinting concept .
To construct fingerprinting, there are four factors which must be considered:

- Hach function: it used to map the substring on integers. It is important to select the hash function in such a way as to minimize the collisions due to mapping different chunks to the same hash[65].

- Fingerprint granularity: it is the size of substring. a fingerprint is more susceptible to false matches if there is fine granularity , whereas large granularity fingerprinting becomes very sensible to changes.[65]

- Selection Strategy: to select substring from ducument, many algorithms are selected[60], such as winnowing[60], There are four type of selection strategies in this state: full fingerprinting, positional strategies, frequencybased strategies, and structure-based strategies. more details on these strategy in[60].
- Fingerprint resolution: is the number of minutiae used to construct a document fingerprint, it might be fixed or variable[60] .

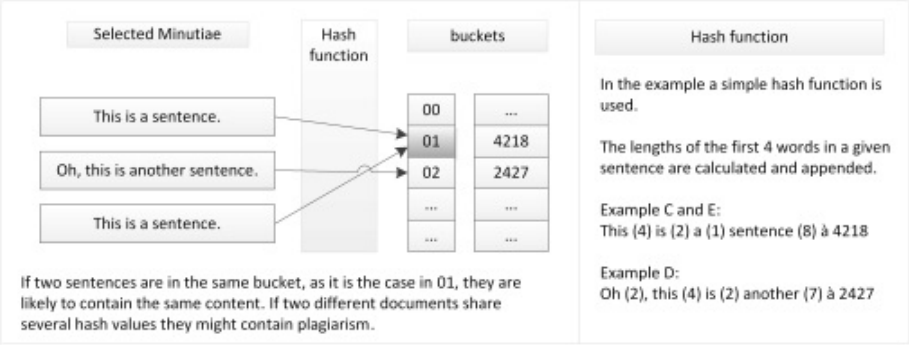

Figure 2.6: Concept of Fingerprinting [44]

2.6.2 Term occurrence analysis

In this category, there are two types[44]:

2.6.2.1 String matching

String matching is the most common approach in computer. It refers to searching for a given character sequence in a text. The plagiarism detection algorithm must caculate suffix document models for the suspicious document and the entire reference collection. the most algorithm used in this case is the Brute Force algorithm. In the "brute force" one checks all the characters of the text with the first character of the pattern. Once has a correspondence between them, we shift the comparison between the second character of the pattern with the following character of the text[19].
The major disadvantages of string matching in a plagiarism detection context are the difficulty of detecting disguised plagiarism, which is attributable to the exact matching approach, and the high computational cost required[44]. An example of string matching is shown in figure 2.7.

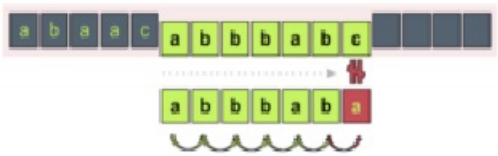

Figure 2.7: String matching representation [19]

2.6.2.2 Vector space model

The documents are represented as a vector and the similarity calculation can then be count on the traditional measure cosine similarity. Vector space model is an algebraic model for representing documents as vector identifiers for example, indexing terms.
Documents and queries are represented as vectors: each $W_{i,j}$ is a weight for the term j of the document i. The classification of documents in a search by keyword can be calculated, using the theory of similarity by comparing the difference in angles between each vector of the documents[19]. In practice, it is easier to calculate the cosine of the angle between the vectors as shown in figure 2.8.

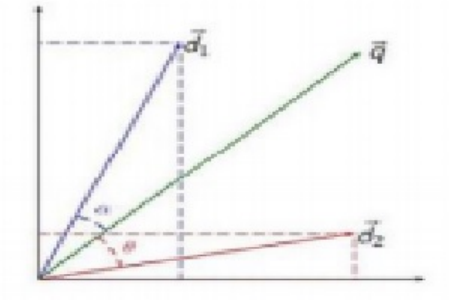

Figure 2.8: Graph of degree of similarity [19]

2.6.3 Citation analysis

It is based on the analysis of citations, it is the only one plagiarism detection approach that is not based on the textual similarity between the documents, It examines citations and references in texts for identify similar patterns in the sequences of citations. This approach is appropriate for texts scientific, academic or other documents that contains citations.

The plagiarism detection by citation analysis is a recent method. citations patterns are quotes sequences that are shared between two documents A and B, as well as the citations potentially unmatched intermediaries[24]. The figure 2.9 below illustrates the concept of citation.

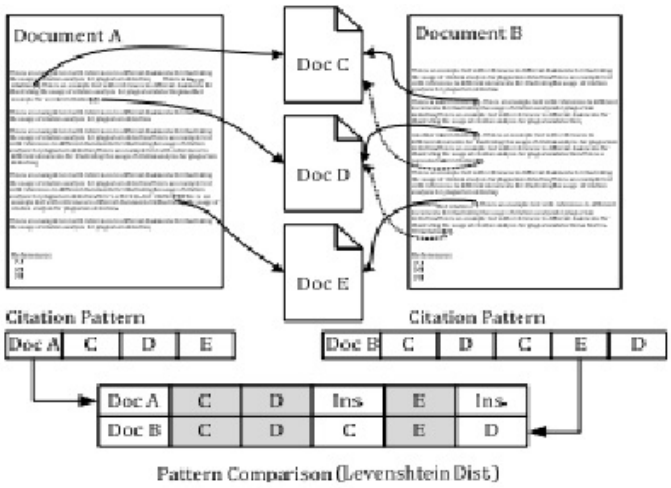

Figure 2.9: Identifying citation patterns for CbPD [24]

2.6.4 Stylometry

Stylometry offers statistical methods for quantify and analyse the writing style of an author, this method uses stylometry to build quantitative style models for segments of a text. The goal is to identify segments that are stylistically different from other segments, it is a potential indicator of plagiarism. It analyzes the structural segments of the text,paragraphs and chapters. It breaks down a text in fixed length segments based on characters or words[24].

2.7 Plagiarism detection tools

There are many tools to detect plagiarism on the Internet, some are free and the others are not free. The goal is to detect plagiarism in student papers and thus allow them to correct them in order to eliminate plagiarism. For teachers, these tools allow them to see how dependent their students are on themselves. There are some criteria used to compare the

tools[34] as shown in follow:

1. Supported languages: which languages are supported by this tool(java, pascal, c..etc, and natural language texts).
2. Presentation of the results: a good presentation should contain the following elements: *Summary*(her,the successful analyses, the parameters used for running the detection, and a chart showing the distribution of similarities over the result should be shown), *Matches*("The matches should be listed sorted by similarity, in a comprehensive way. This can be done pairwise, or in clusters. It should also be possible to set a certain threshold on the minimum similarity to include in the result overview"[34], and *Comparison tool*(It is useful if there is an editor capable of displaying the two files that marked as similar next to each other)
3. Extendible: If it is possible to add other languages that supported by this tool.
4. Exclusion of code: signifies Whether the tool can ignore base code [41].
5. Local: signifies that the tool can work without access to another web service.
6. Submission as groups of files: If the tool can consider a group of files as a submission[41].
7. Open source:"If the source code was released under an open source license"[41]

We will mention some of these tools below

- Plagiarisma: it is free access, it supports 190 language, and it do not store uploaded content. to input files, there are three ways: copy and paste, test by entering URL, and uploading file. http://plagiarisma.net
- Turnitin: it is a textual plagiarism detection, a commercial online service, it developed in 2006 from iParadigm, the suspected document is uploaded by the user to the system database, a complete fingerprint of the document is created and stored by a system, Proprietary algorithms are used to query the three main sources: one is the current and extensively indexed archive of Internet with approximately 4.5 billion pages, books and journals in the ProQuest^{TM} database; and 10 million documents already submitted to the Turnitin database.[49].it is available in http://www.turnitin.com
- Plagscan: it is a paid service, PlagScan has designed for schools, universities, and companies. in order to detect plagiarism in a document, you must first inscribe, you can select one document or more, the results was shown in report. available in www.plagscan.com

- DupliChecker: started in 2006, it is free and is an extrinsic plagiarism detection, the user can access by inscribing only once, but registered user can check for plagiarism for 50 times in a day.User can use several ways such as copy paste, uploading file or by submitting URL to check content's originality[13]. www.DupliChecker.com
- MOSS: a Measure of Software Similarity, it is free service, it is used to detect source code plagiarism. This service takes batches of documents as input and attempts to present a set of HTML pages to specify the sections of a pair of documents where matches detected. The tool support programming languages in C, C++, Java, Pascal, Ada, ML, Lisp, or Scheme programs[13].

2.8 Evaluation of Plagiarism Detection Methods

In order to evaluate the development of a research field, we need an evaluation framework that allows to qualitatively compare various approaches over years. In plagiarism detection, As IR task, a dataset and a set of measures compose the evaluation framework necessary to develop and, perhaps even more important, compare different approaches under a common setting.
Potthast et al.[56] in their survey of evaluation resources and strategies in automatic plagiarism detection on 275 papers (where 139 deal with plagiarism detection in text, 123 deal with plagiarism detection in code, and 13 deal with other media types) they found that 80% of the research work has been carried out considering a local collection of documents and 15% of the papers perform their experiments over the Web in the experimental plagiarism detection task in text . they explain this results by the facts that the Web cannot be utilized easily as a corpus.
In the other hand, about the evaluation strategy followed by the researcher, they found that 43% of research is performing an evaluation based on the precision and recall measures. However, 35% follows a manual strategy, where an expert reviews the cases in order to note the effectiveness of the model.
So, the evaluation is done on two things:

- **Corpora**
  Differents corpora are created in order to used in the test of plagiarism detection algorithms PAN-PC[3] competition is a yearly competition on digital forensics which provides such evaluation framework, this frame contains corpus (PAN-PC-09 corpus, PAN-PC-10 corpus, PAN-PC-12 corpus..etc)and performance measures that address the specifics of plagiarism detection, it allows participants to evaluate their approaches using documents corpus.

[3]PAN is an acronym for "**P**lagiarism Analysis, **A**uthorship Identification, and **N**ear-Duplicate Detection"

- **Evaluation measures**
  Let d_q be a plagiarized document, a plagiarized section s forms a contiguous sequence of plagiarized characters in d_q, d_q describe a sequence of characters each of which is either labeled as plagiarized or nonplagiarized. A plagiarized section s forms a contiguous sequence of plagiarized characters in d_q. The set of all plagiarized sections in d_q is denoted by **S**, where $\forall s_i, s_j \in \mathbf{S} : i \neq j \longrightarrow (s_i \cap s_j) = 0$. Likewise, the set of all sections $r \subset d_q$ found by a plagiarism detection algorithm is denoted by R. an illustration is given by [57] in figure 2.10

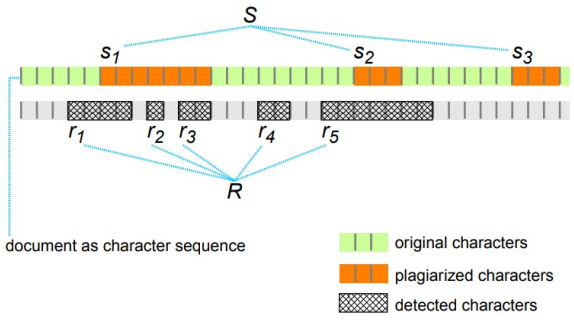

Figure 2.10: A document as character sequence, including plagiarized sections S and detections R returned by a plagiarism detection algorithm [57].

To evaluate perfarmonce for plagiarism detection system, we use th following measures that cited by [4]:
Precision and Recall: is the fraction of the true positive part in each actual and detected case respectively. Their formulas are given in the equations (2.1) and (2.2) .

- *Precision*:
$$prec(S, R) = \frac{1}{|R|} \sum_{r \in R} \frac{|\cup_{s \in S} (s \sqcap r)|}{|r|} \tag{2.1}$$
where $\sqcap$ computes the positionally overlapping characters.
- *Recall*:
$$rec(S, R) = \frac{1}{|S|} \sum_{s \in S} \frac{|\cup_{r \in R} (s \sqcap r)|}{|s|} \tag{2.2}$$
- *Granularity*: quantifies whether the contiguity between plagiarized text passages is properly recognized[56].
$$gran(S, R) = \frac{1}{|S_R|} \sum_{s \in S_R} |C_s| \tag{2.3}$$
where $S_R = \{s | s \in S \wedge \exists r \in R : s \cap r \neq 0\}$.
and $C_s = \{r | r \in R \wedge s \cap r \neq 0\}$

- *F-measure*:
  Precision and recall can be used to compute the F-Measure as defined in equation
  $$F = 2\frac{prec.rec}{rec + prec} \tag{2.4}$$
- *Plagdet*: The measures precision, recall, and granularity are combined to an overall score named Plagdet:
  $$plagdet(S, R) = \frac{F}{\log_2(1 + gran)} \tag{2.5}$$
  where F denotes the F-Measure.

In the example shown in previous figure, the calculation of those measure gives:

$prec(S, R) = \frac{1}{|R|}(\frac{|r_1 \sqcap s_1|}{|r_1|} + \frac{|r_2 \sqcap s_2|}{|r_2|} + \frac{|r_3 \sqcap s_3|}{|r_3|} + \frac{|0|}{|r_4|} + \frac{|r_5 \sqcap s_5|}{|r_5|})$

$prec(S, R) = \frac{1}{5}(\frac{2}{4} + \frac{1}{1} + \frac{2}{2} + \frac{3}{7}) = 0.5857$

$rec(S, R) = \frac{1}{|S|}(\frac{|(s_1 \sqcap r_1) \cup (s_1 \sqcap r_2) \cup (s_1 \sqcap r_3)|}{|s_1|} + \frac{|s_2 \sqcap r_5|}{|s_2|} + \frac{0}{|s_3|})$

$rec(S, R) = \frac{1}{3}(\frac{5}{7} + \frac{3}{3}) = 0,5714$

$F(S, R) = 2(\frac{0,5857.0,5714}{0,5857+0,5714}) = 0,5785$

$gran(S, R) = \frac{1}{2}(3 + 1) = 2$

$plagdet(S, R) = \frac{0,5785}{\log_2(1+2)} = 0,3650$

2.9 Conclusion

In this chapter, we talked about the concept of plagiarism, its types, the different methods of detection, and the various measures used to evaluate the strength of these methods. In the next chapter, We will talk about the related work on extrinsic and intrinsic plagiarism detection methods.

Chapter 3

Related Work

The field of plagiarism is very broad, a lot of work is done in the field of plagiarism detection methods and each of them complements the other, In this chapter we give most of the works that are done on the extrinsic plagiarism detection methods and those of intrinsic plagiarism detection.

3.1 Extrinsic Plagiarism Detection Methods

Several methods have been developed concerning extrinsic plagiarism detection, we are trying to give the most known and used.

3.1.1 Character-Based Methods

The most plagiarism detection algorithms use this category, they manipulate lexical and syntaxic features to find similarity beteewn a query document and a reference collection [3].
In this case, string matching can be exact or approximate. In exact matching, each letter in both strings must be matched in the same order. Several detection algorithms are developed based on character n-gram or word n-gram use exact matching string, Daniel Micol et al. [45] tries to find longest commune substring using character n-gram in order to extract plagiarized fragments.
Grozea et al.[11] use character 16-gram matching, they won first place in the first international competition on plagiarism 2009, they use the ENCOPOLT algorithm to detect the exact matching. The principle of this algorithm is: If it has two sequences A and B in the input, it gives in the output a list(x,y) of position in A,B where there is exactly the same N-gram through the next steps :

- Extract the n-gram of A and B
- Sort these two lists of n-grams
- Compare these lists in a modified mergesort algorithm. Whenever the two smallest N-grams are the equal, output the position in A and the one in B.

Basile et al. [9] use 8-gram matching.

On the other hand, approximate matching string show degree of similarity/dissimilarity between tow string, for example the characters 6-gram x= "aaabbc" and y= "aaabbd" are highly similar because all letters match except the last one. String similarity matrix is the proximity measure available to support the approximate string matching, the table3.3 refers to different kinds of string similarity metrics.
Scherbinin et al. [62] used Levenshtein distance to compare word n-gram and combine adjacent similar grams into sections, Su et al. [7] combined Levenshtein distance, and simplified SmithWaterman algorithm for the identification and quantification of local similarities in plagiarism detection.

Similarity Matric	Dscription	Exemples
Hamming distance	Defines number of characters different between two string x and y of equal length	x="aaabbcc" y="aaabbcd" d(x,y)=1
Levenshtein distence	defines minimum edit distance which transforms x into y. edit operations include: - delete a char, cost 1 - insert a char, cost 1 - substitute one char for another, cost 1	x="aaabbcc" y="aaabbcd" z="aaabbcde" w="aaabbc" d(x,y)=1 -d(x,z)=2 d(x,w)=1
Longest common sequence (LCS) distance	measure the length of the longest pairing of chars that can be made between x and y with respect to the order of the chars allows insertions, cost 1 allows deletions, cost 1	x="aaabbcc" y="aaabbcd" d(x,y)=6

Table 3.1: String similarity metrics [3].

3.1.2 Vector-Based Methods

Lexical and syntax features are released and compared as tokens rather than strings, the similarity can be calculated using vector similarity measures like Jaccard, Dice's, Overlap, euclidean, cosine..etc. Sentences and chunks are presented as either term vectors or character n-grams vector, cosine coefficient and Jaccard coefficient are most used in research works[3]. The table3.2 refer to these vectors similarity .
Albero Barron et al. [7] use word n-gram when n =1..10, they split the suspicious document into sentences s_i , the last one is splited into word n-gram, and they split the source document into word n-gram, each sentence s_i is searched alone over the reference collection.
In order to determine if s_i is a candidate of being from $d \in D$, they compare the corresponding sets of n-grams. Due to the difference in size of these sets, an asymmetric comparison is carried out on the basis of the containment measure in equation (3.1) [40]

$$c(s_i / d) = \frac{|N(s_i) \bigcap N(d)|}{|N(s_i)|} \tag{3.1}$$

where $N(s_i)$ the set of n-gram in s_i, d the source document in reference collection
after considering every d in reference collection, if (3.1) is maximum and is greater than a given threshold, s_i becomes a candidate of being plagiarized from d.
The experimental result show that 2-gram and 3-gram are the best comparison unit for this system [7].
Zhang and al[74] used exponential Cosine distance as a measure of document dissimilarity that globally converges to 0 for small distances and to 1 for large distances. Barrón-Cedeno and al [5] use Jaccard coefficient Similarly to calculate the similarity between n-gram terms of different lengths n=1,2,...,6

3.1.3 Syntax-Based Methods

in order to measure text similarity and plagiarism detection, Some research works have used syntactical features. the authors [69] present a set of low-level syntactic structures that captures creative aspects of writing(such as whether authors tend toward passive or active voice) and show that information about linguistic similarities of works ameliorate recognition of plagiarism(over tfidf-weighted keywords alone)when combined with similarity measurements based on tfidf-weighted keywords.
Elhadi and Al-Tob [18] introduces an approach Using Text Syntactical Structures in Detection of Document Duplicates. This technique ordered and ranked the documents using POS tags. The proposed method minimize the text into a smaller set of syntactical (Pos) tags, each tag is replaced by a single character, this produce the set of strings of tags

Vector similarity metric	Description and Equation	Equation	Range
Matching coefficient	similar to Hammning distance between vector of equal lenght	M(x,y)= $\mid x \mid - \mid x \cap y \mid$	0 To $\mid X \mid$ where $\mid X \mid$=$\mid Y \mid$
Jaccard coefficient	defines number of shared terms against total number of terms	$J(x,y) = \frac{\mid x \cap y \mid}{\mid x \cup y \mid}$	0 To 1
Dice's coefficient	similar to Jaccard but reduces the effect of shared terms between vectors	$D(x,y) = \frac{2\mid x \cap y \mid}{\mid x \cup y \mid}$	0 to 2
Overlap(or containment) coeficient	if v_1 is subset of v_2 or the converse, then the similarity is a full match	$O(x,y) = \frac{\mid x \cap y \mid}{\min(\mid x \mid . \mid y \mid)}$	0 to 1
Cosine coeficient	find the cosine angle between two vectors	$Cos(x,y) = \frac{\Sigma(x,y)}{\sqrt{\Sigma(x_1)^2}\sqrt{\Sigma(y_1)^2}}$	0 to 1
Euclidean distance	measures the geometric distance between two vectors	$Ec(x,y) = \sqrt{\Sigma_i \mid x_i - y_i \mid^2}$	0 to ∞
Sguard Euclidean Distance	places progressively greater weight on vectors that are further apart	$SEc(x,y) = \Sigma_i (x_i - y_i)^2$	0 to ∞
Manhattan Distance	measures the average difference across dimensions and yields results similar to the simple Euclidean distance	$Manh(x,y) = \Sigma_i \mid x_i - y_i \mid^2$	0 to ∞

Table 3.2: Vector similarity metric [3].

which represent the document, in this stage they use TreeTagger[1] because it is easily accessible and has a large set of tags. The produced strings are used by Longest Comman Sequences (LCS) algorithm in order to produce similar groups string, nearly related documents should cluster together and have high similarity when run through some clustering tool. The results confirm that frequencies of tagged documents can serve as an indication of similarities between documents.

3.1.4 Semantic-Based Methods

A sentence can be exminated as a set of words presented in a certain order, in this state two sentences have the same semantic but differents in their structure, the active and passive

[1]is a tool that annotates a text with information on parts of the speech (kind of words: nouns, verbs, infinitives and particles) and information on lemmatization

voice are the exempel for this way. WordNet is used in this content to find the semantic similarity between words or sentences, *Li Yuhua et al* [39] propose an approach calculate a semantic similarity between two sentences using information from a lexical database and from corpus statistics, *Victor U Thompson et al*[68] combine methods to detecting the most commune techniques used in paraphrasing text, which are synonym replacing (lexical substitutions), word recording (syntactic alterations) and deletion/insertion (edit operations),they use semantic similarity, syntactic similarity and similarity when insertions/deletions are taking into account to detect plagiarism in paraphrase documents.
In the proposed paraphrase retrieval model, suspect and source text passage are splited into sentences and each one is pre-processing as follow:

1. sentence is tokenised into words
2. words are normalized to lower-case.
3. stop words are removed.
4. words are stemming

After this stage, each sentence in the suspect passage is compared with sentences in the sources passage using similarity methods for measuring semantic similarity, syntactic similarity and measuring similarity when insertions/deletions appeared. If the similarity score for each sentence is under than a predefined threshold, the sentence is discarded. In the other part, the similarity scores for the sentences are averaged for each methods and transfer into matching learning classifier where the suspect passage is classified as paraphrased or not.
In order to calculate the semantic similarity between pairs of sentences, they use equation (3.2) .

$$semanticsim(sp, sr) = \frac{count(lexical\ sibstitution)}{len(sp)} \tag{3.2}$$

More details on how calculate semantic similarity, syntactic similarity and measuring similarity when insertions/deletions appeared are in the original paper[68]. In order to implement and evaluate the proposed methods, they use two corpus: The crowd sourcing corpus (Burrows et al., 2012) and the Cloughs and Stevenson (Clough and Stevenson , 2011) corpora, both contain simulated paraphrased plagiarised texts.The results on the first corpus are presented in the table3.3
The results confirm that the best approach for detecting the paraphrase plagiarism problem is to develop methods for detecting the most common paraphrase plagiarism techniques and combine those methods.

Methods	Precision	Recall	F-1	AUC-ROC
semantic	0.800	0.923	**0.857**	0.915
syntactic	0.796	0.918	0.853	0.868
insert/delete	0.798	0.871	0.833	0.897
sem,syn	0.793	0.939	0.860	0.928
sem,ins/del	0.789	0.936	0.858	0.916
syn,ins/del	0.782	0.932	0.855	0.907
sem,syn,ins/del	**0.803**	0.938	**0.865**	**0.9.17**
Bar and all,(2012)			0.852	
Burrow and all,(2012)			0.837	
Baseline and all((GST)	0.768	0.922	0.838	0.887

Table 3.3: Results from the Experiments on the Crowd Paraphrase Corpus [68].

3.1.5 Fuzzy-Based Methods

In a fuzzy-based methods, the similarity between texts such as sentences is represented by value that range from one(exactly matched) to zero(entirely different).A fuzzy set is a set that contain words wich having de same meaning with each word in documents, there is a degree of similarity between the words in document and the fuzzy set [72], in [72] a term-term correlation matrix is created which contain words and their corresponding correlation factor that calculate the degree of similarity among different words,then, in order to calculate the degree of similarity between sentences, he compute correlation factors between pair of words from two different sentences in their respective documents, The term-to-term correlation factor defines a fuzzy similarity between two word.
Azahrani et al [2] propose an approach using a fuzzy semantic-based string similarity for extrinsic plagiarism detection, their algorithm is described as follow:
in the pre-processing stage, they remove stop words and use the PORTER stemmer algorithm, for generating k-shingles, the k was set to 3 (i.e. word-3-grams),to retrieve the condidate documents, they compute jacard similarity as following in equation (3.3) and a threshold value was Stabilized(0.1)

$$J(A,B) = \frac{\mid shingles\ of A \cap shingles\ of B \mid}{\mid shingles\ of A \cup shingles\ of B \mid} \tag{3.3}$$

For semantic-based analysis, WorldNet v3.0 using MySQL2 was used to query the Sunset table and extract synonyms of the words, and finally they use term-to-sentence correlation factor $\mu_{q,x}$ as shown in equation (3.4) to calculate fuzzy similarity between two sentences s_x, s_q with $s_x \in d_x$(candidate document) and $s_q \in d_q$(suspicious document) as follow in equations (3.5)

$$\mu_{q,x} = 1 - \Pi w_k \in s_x(1 - F_{q,k}) \tag{3.4}$$

$$sim(s_q, s_x) = \frac{\mu_{1,x} + \mu_{2,x} + ... + \mu_{q,x} + ... + \mu_{n,x}}{n} \tag{3.5}$$

where $w_x \in d_x$ and $F_{q,k}$ is a fuzzy similarity between w_q and w_k that they defined as follow:

$$F_{q,k} = \begin{cases} 1 & \text{if } w_k \text{ and } w_q \text{ are identical} \\ 0.5 & \text{if } w_k \text{ is in the synset of } w_q \\ 0 & \text{otherwise} \end{cases}$$

in order to judge that two sentences as equal (i.e. plagiarized), the minimum similarity score should be above a threshold value $(\alpha > 0.65)$ as follow.

$$EQ(s_q, s_x) = \begin{cases} 1 & \text{if } MIN(Sim(s_q, s_x), Sim(s_x, s_q)) \geq \alpha \\ 0 & \text{otherwise} \end{cases}$$

Their results on PAN'10 are shown as follow (recall= 0.1259,precision= 0.5761, granularity= 3.5828), The low recall was caused by several reasons, one of them is that algorithm was designed for extrinsic plagiarism task and did not concern intrinsic nor cross-lingual plagiarism

3.1.6 Structural-Based Methods

In fact, we found a few studies interested in this categorie, previouse methods use the flat features representation such as lexical, syntactic and semantic features, structural-based methods uses contextual similarity such as how the words are used in entire documents for exemple paragraph, section. In order to manipulate contextual information, they use tree-structure features representation, *Rahman et al*[58] propose a hierarchical tree-structured representation of documents that consist of text content only, they use only 'html' documents at this stage, they process as follows in order to extract tree-structure:
- the document is partitioned into paragraphs blocks using the html paragraph tag "<p>" and new line tag "
"
- subsequent paragraph blocks is merged in order to form a new page until the total number of words of the merged blocks exceeds a page threshold value 1000. There is no minimum threshold for the last page. The page blocks are formed.
- each page is split into a smaller blocks using more html tags: "<p>", "
", "<il>", "<td>", etc. Merge these subsequent blocks in the same fashion of Step 2 to form a new paragraph until total number of words of the merged blocks exceeds a page threshold value 100. The minimum threshold for the last paragraph of a page is kept 40; otherwise the paragraph is merged with the previous paragraph.
The authors of [12] use multilayer self-organizing map (MLSOM) With Tree-Structured Data to construct a new document retrieval (DR) and plagiarism detection (PD) system, where a document is represented as tree-structured and in order to handled this last, they use an MLSOM algorithm which is developed in the past for the application of image retrieval , but in their approach they use them as clustering algorithm. two approach to

detect plagiarism are proposed, the first is an extension of the DR method with additional local sorting. The second method uses document association on the bottom layer of ML-SOM, more details on these approachs in[12].

3.1.7 Methods for Cross-Lingual Plagiarism Detection

Gross-linguage plagiarism detection methods use textual features for cross-language to caculate similarity between section of suspisious document d_q wich is writing in language L1 and section of source documents d wich is writing in language L2. *Corezola Pereira et al.*[52], *Barrón Cedeño et al.*[6], use an automatic translation tool to translate the source and suspisious documents into in samse language in order to analyze them such as a monolangual methods.*Pottast et al*[55] propose a new multilingual retrieval model for analysis of cross-linguage similarity called Cross-Language Explicit Semantic Analysis(CL-ESA) which is based on monolingual retrieval model ESA, where in ESA a document d is represented as an *n* dimensional concept vector **d**:

$$\mathbf{d} = (\varphi(v, v_1^*), \varphi(v, v_2^*), .., \varphi(v, v_n^*))^T \tag{3.6}$$

where : v is the vector space model representation of d, v_i^* is the vector space model representation of the ith index document in D^*, D^* a document collection of so-called index documents, and φ the cosine similarity measure, if $\varphi(v, v_i^*)<\varepsilon$($\varepsilon$ is the noise threshold)the respective entry is set to zero. Let $\mathbf{d}_1$ be the concept vector representation of another document d_1. the similarity between **d** and $\mathbf{d}_1$ with ESA is defined as $\varphi(\mathbf{d}, \mathbf{d}_1)$
In CL-ESA, the principe is:
if $\mathbf{L} = \{L_1, L_2, ..., L_n\}$ a set of languages, $D^* = \{D_1^*, D_2^*, ..., D_n^*\}$ a set of index document collections where each D_i^* contains index documents of L_i, and if $d \in L$ and $d_1 \in L_1$, d, d_1 are represented as ESA vector **d** and $\mathbf{d}_1$ by using their index document collections $D^*, D_1^* \in D^*$ that corresponds to L, L_1^* respectively.The similarity between d and d_1 is quantified in the concept space, by computing the cosine similarity between **d** and $\mathbf{d}_1$, this model requires a comparable corpora such as wikipedia.
Potthast et al[53] in their survey present standard process of detection plagiarism in gross language as shown in figure 3.1
In heuristic retrieval, they quoted three approaches in order to retrieve candidate documents, the first use cross-language information retrieval(CLIR)where a query reformulation was created using keywords extracted from the suspicious document and translated into the corresponding language , the next two approaches depend on the results of machine translation and make use of either standard keyword retrieval (an IR solution)or hash coding as shown in figure 3.2
In detailed analysis, they outlined retrieval models to measure cross-language similarity: models based on language syntax using character n-grams features for languages that are syntactically similar such as European languages (CL-CNG[43]), models based on dictionaries, models based on comparable corpora(the cross-language explicit semantic

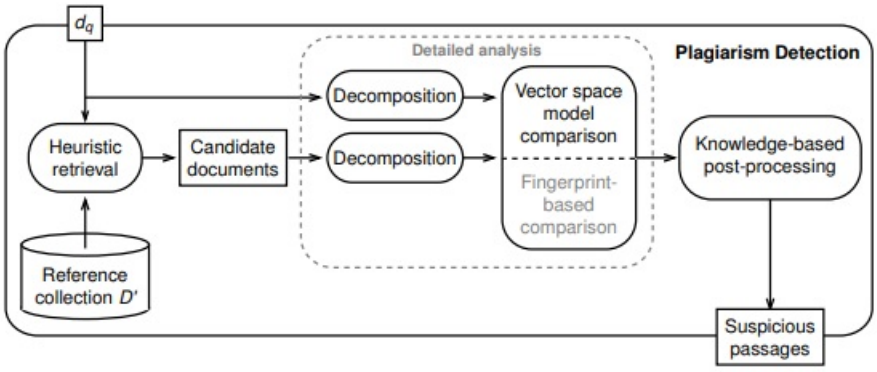

Figure 3.1: Retrieval process for cross-language plagiarism detection [67]

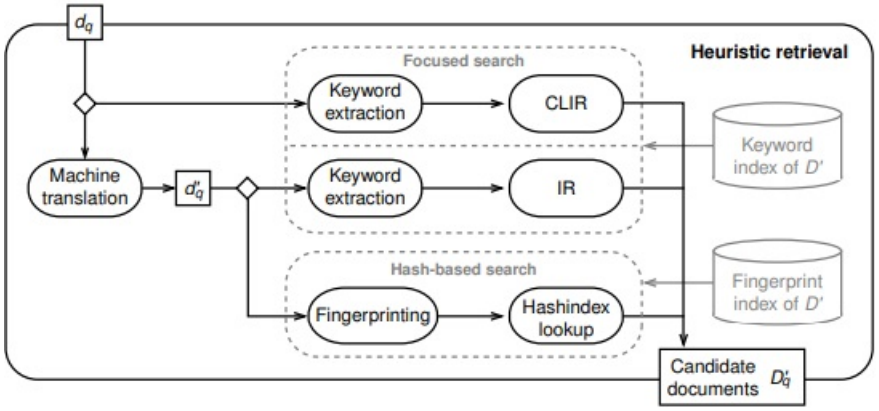

Figure 3.2: Retrieval process of the heuristic retrieval step of cross-language plagiarism detection [53]

analysis model (CL-ESA[55])), and models based on parallel corpora(the cross-language alignment-based similarity analysis model(CL-ASA[8])) as shown in figure 3.3
Pottast and all use CL-C3G, CL-ESA,CL-ASA in thier evaluation on two corpora, the comparable Wikipedia corpus and the parallel JRC-Acquis corpus,the results show that CL-ASA achieves good results on professional and automatic translations. CL-CNG outperforms CL-ESA and CL-ASA. However, unlike the former, CL-ESA and CL-ASA can also be used on language pairs whose alphabet or syntax are unrelated.

3.1.8 Citation-Based Methods

In these methods, they use citations and references for determining document similarities in order to identify plagiarism[25].

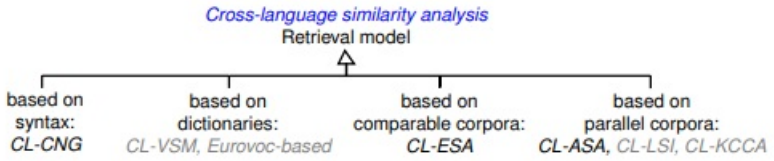

Figure 3.3: Taxonomy of retrieval models for cross-language similarity analysis [53]

Kessler[35] introduces the first citation-based approach wich is named bibliographic coupling. Two document A and B are called bibliographiclly coupled if they shared one or more than document in their cited documents. it allows the calculation of the coupling strength and used to identify related articles by academic search engines[23]. unique consideration of bibliographic coupling strength can not indicate potential plagiarism efficeintly [24]
Bela Gipp and all[23] propose a new approach called Citation Order Analysis (COA)wich is based on citation analysis On the contrary of traditional approach which are analyse document's words. First, The document is analysed and a series of heuristics applied to process the citations, including their position within the document (The citations were parsed using a modified version of parsCit (http://wing.comp.nus.edu.sg/parsCit) in combination with the authors self developed software, which is available upon request). second, Citations are corresponded with their entries in the bibliography. finally, The citation-based similarity of the documents is calculated. In the basic version, only the order is considered; in the more advanced version, the distance between two citations is evaluated as well. Even if a document is translated, the order of citations within sentences or paragraphs might change due to different sentence structures or writing styles document .
In other reaserch of Bela Gipp[24],they present three citation pattern analysis algorithms for citation-based plagiarism detection(CbPD), namely Greedly Citation Tallin(GCT), Citation chunking(CC) and Longest Commune Citation Sequence(LCCS). Citation sequence of document is similar to a string which is used to identify existing similarity functions.
GCT use principe of Greedly String Tillin(GST)algorithm which is work to find all matching substring with individually longest possible size in two sequences. The corresponding individually longest match in two sequence is called Tile, the last is represented as a tuple $t = (s_1, s_2, l)$ include the first position of a longest match in the first sequence (s_1), the first position in the second sequence (s_2) and (l) the length of the match.
In GCT algorithm, the extraction consist of matching citation instead of matching substring as shown in figure 3.4:
For more details on LCCS and CC, look at the original paper[24].

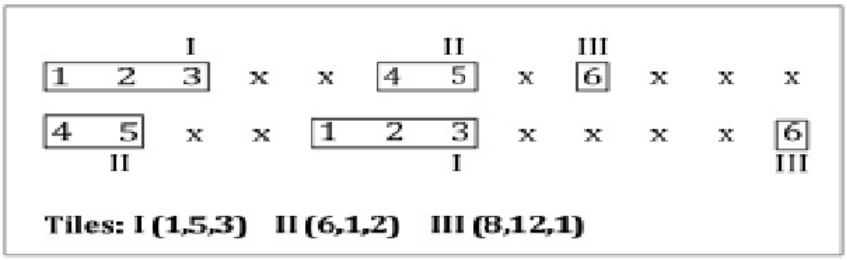

Figure 3.4: citaton Tiles [24]

3.2 Intrinsic Plagiarism Detection

Most researches in intrinsic plagiarism detection use either be lexical, character or syntactic features.

3.2.1 The Averaged Word Frequency class

The averaged word frequency class is a new vocabulary richness statistic that *Eissen and Stein* introduced which proved to be more powerful and stable concept for intrinsic plagiarism detection, it concern in the lexical features; The importance of a document's averaged word frequency class informs us about style complexity (readability, writing complexity and vocabulary richness) [11].Than the methods which are established on the ratio between the number of different words within a document, The frequency class of a word is related to Zipf's law [32].
The authors define word frequency class **c(w)** of a word **w** in a corpus **C** as [76]:

$$\lfloor \log_2(\frac{f(w^*)}{f(w)}) \rfloor \tag{3.7}$$

where:
$f(w^*)$: The most frequently used word in **C**.
$f(w)$: The frequency of the word **w** in **C**.

Remarkable Property:

- Computed in linear time in the number of words.
- Small variance with respect to text length.

In experimental analysis *Eisen and Stein* [76] try to given a solution for these questions:

1. Which vocabulary richness measure is suited best? which leads us to the question: How stable is a measure with respect to text length?

2. To which extent is the detection of plagiarized text portions possible?

- The first question can be reformulated as a document classification task, given a reference corpus with plagiarized and non plagiarized documents.

Yule's K [73] and *Honore's R* [30] are famous example of methods which measure the vocabulary richness are based on the ratio between the number of different words and the total number of words within a document.These measures depend on document/passage length[64, 53], so they are not suitable for compare passages of varying lengths and deliver unreliable results for short passages, which is a disqualifying criterion for plagiarism analysis.
According to comparison of Honore's R, Yule's K, and the average word frequency class is shown in the right plot of the figure in page 42 [76]; here, the analysed text portion varies between 10% and 100% of the entire document. Observe that the average word frequency class is stable even for small paragraphs, which qualifies the measure as a powerful instrument for intrinsic plagiarism analysis.

- As a result also showed that the introduced averaged word frequency class exceeds other measures in this respect.

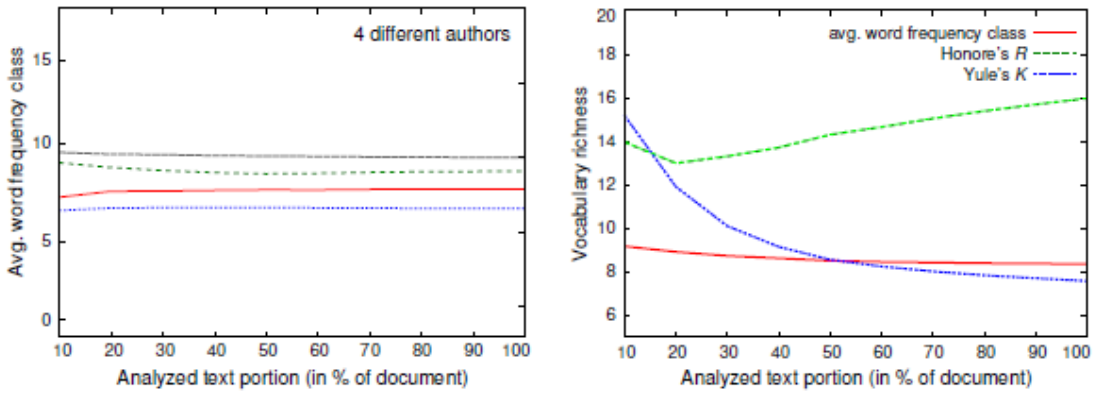

Figure 3.5: Average word frequency class of four different authors (left plot). The right plot shows the development of Honore's R, Yule's K, and the average word frequency class of a single-author document for different text portions [76].

3.2.2 N-gram Profiles

We say profile to the representation as a vector of n-gram which is a sequence of n characters or words and their frequencies.
The use of " n-gram profiles compares segments of the document against the whole document. This approach is based on the supposition that the document has a main author, who wrote the majority, if not all the text. So, the comparison between the style of a particular segment with the whole document style could reveal important variations, meaning that other authors are involved.

Stamatatos [64] presented attempts in character features to quantify the style variation within a document using *character n-gram profiles* and a style change function as the author is based on an appropriate dissimilarity measure originally propose for author identification.
In addition. *Stamatatos* proposes a set of heuristic rules that attempt to detect plagiarized-free document and plagiarized passages as well as to reduce the effect of irrelevant style changes within a document.

These style profiles are first constructed using a sliding window which is defined over the text length, then compare the text in the window with the whole document; after that, they use a function that quantifies the style changes within the document. the anomalies peak of that function may indicate the plagiarism sections(when there is a peak that means a plagiarized section).
The following dissimilarity measure normalized has been used:

$$Nd(A,B) = \frac{\sum_{g \in P(A)} (\frac{2(f_A(g) - f_B(g))}{f_A(g) + f_b(g)})^2}{4 \mid P(A) \mid} \tag{3.8}$$

where:

- $f_A(g)$ and $f_B(g)$ are the frequency of occurrence (normalized over text length) of the n-gram g in text A and text B, respectively.
- $|P(A)|$ is the size of the profile of text A. Then he defined the style change function Sc of a document D like that:

$$Sc(i,D) = Nd(w_i, D), i = 1... \mid w \mid \tag{3.9}$$

where:
w: sliding window of length l (in characters) and step s (in characters) and $\mid w \mid$ is the total number of windows.

If we have a text with x characters $\mid w \mid$ is computed as:

$$\mid w \mid = \lfloor 1 + \frac{x - l}{s} \rfloor \tag{3.10}$$

The evaluation results of *Stamatatos* show that it is capable to detect about half of the plagiarized sections. Otherwise, the precision remains low.

Oberreuter [47] proposed a method involving the lexical features to judge whether what is plagiarism by using words frequency features (word vector including stop words with term frequency weighting), as the author the words which an author uses are very important, because different authors tend to use different words to write their ideas, either in the same topic or not.

Algorithm 3 Intrinsic plagiarism detection evaluation of *oberreuter*

for c $\in$ C **do**
$d_c \longleftarrow 0$
build v_c using term frequencies on segment c
for word w $\in v_c$ **do**
$d_c \longleftarrow d_c + \frac{|freq(w,v) - freq(w,v_c|}{|freq(w,v) + freq(w,v_c|}$
end for
end for
style $\longleftarrow \frac{1}{C} \sum_{c \in C} d_c$
for c $\in$ C **do**
If $d_c \inf style$ - δ **then**
Mark segment c as outlier and potential plagiarized passages.
endif
end for

As presented in Algorithm3, the general footprint or style of the document is represented by the average of all differences computed for each segment and the complete document. Note that, every segment is compared against the whole document only in terms of the words present in the segment. Also, this algorithm takes into account the intuition; if certain words are only used on a certain segment, the comparison of that segment against the whole document would lead to a low value, because the frequency of those words would be the same in both the whole document and in the segment. Finally, all segments are classified according to its distance with respect to the document's style.

Kestemont et al. [36] use the 2500 most frequent char-3-grams, and *Rao et al.* [59] use char-3-grams as well as other well-known features that quantify writing style.

3.2.3 Kolmogorov complexity measures

Kolmogorov complexity measures as style features in intrinsic plagiarism analysis of syntactic features, are based on segment the text according to word classes then attach these word classes with a binary string. *Kolmogorov complexity* is used to define the complexity or degree of randomness of a binary string.
For approximate *Kolmogorov complexity* of a string x, it is possible to use any lossless compression algorithm.

Let A be any compression algorithm and *A(x)*: the results of compression x using A. The approximate *Kolmogorov complexity* of x using A, denoted K_c*(x)*, it defined as:

$$K_c(x) = \frac{Length(A(x))}{Length(x)} + q \tag{3.11}$$

where q is the length in bits of the program which implements A.

Seaward and Matwin [61] use that method as the authors, they test with complexity features based on the *Lempl-Ziv*[10] compression algorithm for detecting style variations within a single document. Thus revealing possible plagiarism passages. But The problem, as it was discovered, was the high degree of granularity required by the task. Complexity analysis does not do well with short text.

3.3 Conclusion

In this chapter, we presented the famous research works on extrinsic and intrinsic plagiarism detection methods, in the next chapter, we will implement the Stamatatos method in intrinsic plagiarism detection based on 3-gram.

Chapter 4

Implementation

After identifying in the previous chapter the most important works which accomplished in plagiarism detection methods, we will try in this chapter to implement one of those in intrinsic plagiarism detection using JAVA programming language.we choose the method of *Stamatatos (2009)* who participated with its in the PAN 09 competition and use the PAN-PC 09 corpus to evaluate this method. That Experimental is done in a PC Intel Core $i5$ with processor of 1.80 GHz and RAM 6.00 Go, the operation system is windows 8.1.

4.1 PAN-PC Corpus

In our Implementation, we use PAN-PC 09 corpus which is the first corpus developed in the PAN-PC serie competition, it contains 42223 text documents in which 94202 case of artificial plagiarism, the main statistics of this corpus are summarised in Table 4.1
The important parameters of the corpus are[57]:

Documents statistic				Obfuscation statistic	
Documents purpose		doc lenght			
source document	50%	short(1-10pp)	50%	none	35%
suspisious doc		midium(10-100pp)	35%	paraphrasing	
- with plagiarism	25%	long(100-1000pp)	15%	- automatic(low)	35%
- without plagiarism	25%			- automatic(high)	20%
				translation	10%

Table 4.1: Statistics of the PAN-PC-09 corpus [69]

- Document Length: documents in corpus are devised into three categories; small documents(1-10pages) represent 50%, medium documents (10-100pages) 35% and 15% large documents(100-1000p).

- Plagiarism Percentage: in the suspisious documents, 50% of them contain no plagiarism in all.
- Suspicious-to-Source Ratio: 50% of corpus are documents source, 50% are suspisious documents.
- Plagiarism Length: The plagiarized fragment have length between 50 words and 5000 words.
- Plagiarism Languages: most cases are monolingual plagiarism (90%) and the rest are multilingual plagiarism which is translated automatically from German or Spanish to English.
- Plagiarism Obfuscation: there is two main steps to generate plagiarism cases in corpus: *extraction-insertion* where a text fragment **s** is selected from a document than is insertde in another document, and *obfuscation* where a fragment **s** is modified before inserted it into a suspicious document [69], the degree of obfuscation vary from none to high.
  according to Potthast and al. [57], there are three heuristics to generate text fragment s_q from a text fragment s_x:
  1. Random Text Operations: modifications are applied to a text fragment s_x, words or short phrases in s_x are shuffled, removed, inserted, or replaced randomly.
  2. Semantic Word Variation: The vocabulary in s_x is substituted by the corresponding synonym, antonyms, hyponym or hypernym.
  3. POS-preserving Word Shuffling: The words in s_x are re-ordered such that its original POS sequence is preserved.

4.2 The Proposed System Work-flow

In manipulation we choose from intrinsic plagiarism detection the method proposed by *Stamatatos*, because the effectiveness of character n-gram in quantify writing style as the prouve of many study[51, 31].
As we see earlier intrinsic plagiarism detection based in building blocks (Chunking strategy, Writing style retrieval model, An outlier detection algorithm and Post-processing).

Chunking strategy

We employ a sliding window chunking with size around 1000 characters, The slide stepping of window ranges 200 characters; Each profile (ngram with its frequencies) window will be compared with the profile of the hole document.

Writing Style retrieval model

The feature representations here is character features which is character n-grams (3-gram). To produce the profiles (the representation 3-gram and its frequencies) we use two functions:

1. `public HashMap<String, Integer> gramGlobal(String path, int n)`

   which produces the profile of the hole document D.

```
                // produce the profile n-gram
   public HashMap<String, Integer> gramGlobal(String path, int n ) {


     HashMap<String, Integer> counter = new HashMap<>();
               // n-gram
   for (int i=0; (i <= path.length()-n);i++) {

     String ram=path.substring(i, i+n);
      //count the frequencies of n-gram
     if (counter.containsKey(ram)) {
       counter.put(ram, counter.get(ram) + 1);

           }
           else counter.put(ram, 1);


   }

   return counter;
 }

```

Listing 4.1: Document profile function

2. `public ArrayList<HashMap<String, Integer>> NewPro(String D, int n)`

   Produces an ArrayList which contains the profile of every segment of the document D.

```
public ArrayList<HashMap<String, Integer>> NewPro(String D, int n) {

     int s=0;

       ArrayList<HashMap<String, Integer>> SegmProfile = new ArrayList<
  HashMap<String, Integer>>();
        for (int i=0;i<(Math.abs(1+((D.length()-leng)/step)));i++) {
          HashMap<String, Integer> chunk =new HashMap<String, Integer>()
  ;
```

```
          int a=s;
           while( a<leng+s) {
                if(a+n<D.length()) {
          String ram=D.substring(a, a+n);
                if (chunk.containsKey(ram)) {
                  chunk.put(ram, chunk.get(ram) + 1);
                }
                else chunk.put(ram, 1);
             }
                 a=a+n;
           }
           s+=step;
           if (chunk.isEmpty())
             System.out.println("vide :  ");
           SegmProfile.add(chunk) ;
        }
        System.out.println("segmentprofile:  "+SegmProfile);
    return SegmProfile;

    }

```

Listing 4.2: Segment profile function

Similarity measures used *Stamatatos'* normalized distance measure Nd

$$Nd(A,B) = \frac{\sum_{g \in P(A)} \left(\frac{2(f_A(g) - f_B(g)}{f_A(g) + f_b(g)}\right)^2}{4 \mid P(A) \mid} \tag{4.1}$$

where $f_A(g)$ and $f_B(g)$ are the the frequency of n-gram g in text A and B in our case B is D and A is segment$w$ from D.

```
public double similarity(HashMap<String, Integer> w,
                         HashMap<String, Integer> B).
```

```
   // the similarity between the profile of the segment and the hole
   document
    public double similarity(HashMap<String , Integer> w,HashMap<String ,
   Integer> B){
      double val = 0;
      //System.out.print("b.."
        //+B+" "+w);
      for(String a:w.keySet() ) { //loop with gram
                  //test if B contain the key a (gram) of A .

        if(B.containsKey(a)) {
                  // measure normalized d1 stamatatos.
           val+= Math.pow(
```

```
                (2*(w.get(a)/(double)w.size()-B.get(a)/(double)B.size())
                /
                (w.get(a)/(double)w.size()+B.get(a)/(double)B.size()))
                ,2);
// diviser par le nbr de 3gram.

        }
      }
      double result=val/(4*w.size());

      return (result);
    }

```

Listing 4.3: Similarity function

where B is profile of D and w is a segment profile of D.

Outlier Detection

Measuring the deviation from the average document style

```
public HashMap<Integer, Double> StyleChange(String Global,int number)
```

```
public HashMap<Integer, Double> StyleChange(String Global,int number)
    throws IOException {
  double result = 0.0;
  ArrayList<HashMap<String, Integer>> Tab=NewPro(Global,number);
  HashMap<String, Integer> Document=gramGlobal(Global,number);
   File f = new File ("G:\\Sample.txt");
     if (!f.exists()) {
         f.createNewFile();
     }
     FileWriter fw = new FileWriter(f.getAbsoluteFile());
     BufferedWriter bw = new BufferedWriter(fw);
    bw.write("x"+";"+"y"+ System.getProperty("line.separator"));


  //test the similarity between the profiles (segment /document)
  for(int i=0;i<Tab.size();i++) {
   result=similarity(Tab.get(i),Document);
   style.put(i,result);
   bw.write(i+";"+result + System.getProperty("line.separator"));

  }     bw.close();

     System.out.println("style"+style);
    return style;
    }
```

Listing 4.4: Style Change function

This function use gramGlobal(String path, int n) and NewPro(String D, int n) in a loop of the ArrayList of segment profile and inside the loop there is a call for similarity(HashMap<String, Integer> w,HashMap<String, Integer> B); the size of the ArrayList of segment profile is precise at

$$\lfloor 1 + \frac{D.length - 1000}{200} \rfloor \tag{4.2}$$

Chunk clustering: Comparing the chunk(segment) representation, attempting to cluster them into groups of similar styles *Identifying plagiarized passages:* Given StyleChange() function, the task of an intrinsic plagiarism detection is to detect peaks of that function corresponding to significantly different text section from the rest of the documents

HashMap<Integer, Double> style is a variable contains number of segment as a key and its similarity with the hole document as a $value$.

a value of 0.02 which determine the plagiarism-free Threshold.

`public double Mean(Collection<Double> collection)`

is the mean of StyleChange

```
1  public double Mean(Collection<Double> collection) {
2       double S=0.0;
3       for (Double s : collection) {
4       S+=s;
5     }
6     return S/collection.size();
7     }
```

Listing 4.5: Mean function

`public double standardDeviation(Collection<Double> collection)`

is the standard deviation

```
1  public double standardDeviation(Collection<Double> collection) {
2   //compute the standard deviation of the similarity
3       double std=0.0;
4       double mean=(double) Mean(collection);
5       for (Double var : collection) {
6           std+= Math.pow((double)var -(double) mean, 2);
7         }
8         double standDeviation= (double)std/((double)collection.size()-1);
9       return Math.sqrt(standDeviation) ;
10    }
```

Listing 4.6: Standard deviation function

For the detection in Document level:

```
double Deviation=standardDeviation(style.values());
  if(Deviation > Threshold)
     System.out.println("that Document is plagiarised");
  else System.out.println("that Document is plagiarism-free");
```

For Identifying plagiarized passages: we remove from Style Change all the text windows with value greater than Standard deviation + mean as

$$StyleChange(i', D) > mean + a^{*}standardDeviation' \tag{4.3}$$

where a is a constant determining the sensitivity of plagiarism detection method, empirically determined to 2.0.

1. `double equl=Deviation+Mean(style.values());`

2.
```
Iterator<Map.Entry<Integer, Double>> test =
                style.entrySet().iterator();
              while (test.hasNext()) {
                 Map.Entry<Integer, Double> entry =
                  test.next();
                 if(entry.getValue()>equl)
                  { //remove from style change all the text windows
              //with value greater than Standard deviation + mean.\\
                  test.remove();
                  }
                  }
```

```
1  public Map<Integer, Double> DetectingPlagiarism(String D) {
2      double Threshold=0.02;
3          double Deviation=standardDeviation(style.values());
4         System.out.println("Standard deviation"+Deviation);
5          if(Deviation > Threshold) {
6           System.out.println("that Document is plagiarised");
7           countPre++;
8          }
9          else System.out.println("that Document is plagiarism-free");
10             double equl=Deviation+Mean(style.values());
11             Iterator<Map.Entry<Integer, Double>> test = style.entrySet().
     iterator();
12             while (test.hasNext()) {
13                 Map.Entry<Integer, Double> entry = test.next();
14                 if(entry.getValue()>equl) { //remove from style change all
     the text windows
```

Document	Standard deviation
0005	0.070
00017	0.061
00034	0.062
00022	0.073

Table 4.2: The Standard Deviation of each document

```
15                                              //with value greater than
  Standard deviation + mean.
16                  test.remove();
17                }
18             }
19             System.out.println("that is hash map2 :"+style);
20
21            for (int val:style.keySet()) {
22             // pret(D,val);
23              //Plagiarized passage criterion:
24             if ( style.get(val) > Mean(style.values())+a*standardDeviation(
  style.values())) {
25                System.out.println("that is a Plagiarized passage:"+val);
26              }
27            }
28        return style;
29     }
```

Listing 4.7: DetectingPlagiarism function

Post-Processing

- Merging the overlapping and consecutive chunks that have been identified as outliers
- Rationale: to decrease detection granularity.

4.3 Discussion and Results

For document level the criterion we used is based on the variance of the style change function, if the document is written by one author, we expect the style change function to remain relatively stable. On the other hand, if there are plagiarized sections, the style change function will be characterized by peaks that significantly deviate from the average value.
However, the Nd in 4.1 measure is not independent of text length. Very short documents

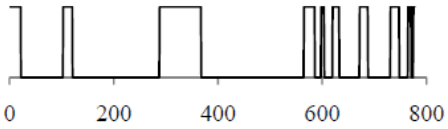

Figure 4.1: The real passage plagiarised of the document 0005 [64]

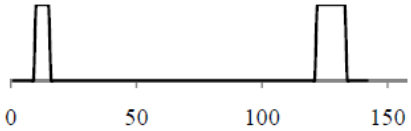

Figure 4.2: The real passage plagiarised of the document 00034 [64]

tend to have low style change function values. Moreover, very long texts are likely to contain stylistic changes made intentionally by the author. As the previous results we observe the style change function of documents 00017 showing in the page 56 and 00034 showing in page 58 fall under the plagiarism-free criterion. The former is a successful case where no plagiarism exists. On the other hand, in document 00034 actually it has two passage of plagiarism as shown in figure 4.2, but the style change function fails to produce significant peaks that would increase its standard deviation, also the average of standard deviation of that document is less than it in 00017 however that last has not any plagiarized passage. For the 00022 document of the page 57, the standard deviation of its style function is greater than the threshold as shown in the table 4.2 moreover this document is plagiarism-free.

In the document 0005 showing in the figure on page 55 show that the real passage plagiarism shown on the page 54 is too closely to the peaks these are in the representation of style change function of the document 0005.

In our work, we achieve a results which can precise if the input file is plagiarism-free or not, then extract the plagiarized sections; our result is very closely to the result presents in *Stamatatos* [64]. For test our work with the entire corpus and extract from it evaluation metrics that make to us a challenge, because that corpus big and need hardware that we can not establish it because of time shortage, maybe by more of time we can present the hopeful results.

One of the power point of that method is its simplicity and language independence resources. Moreover, it requires no text segmentation or preprocessing. But it need an optimization and tuning in the level of its parameter (Sliding window length, Sliding window step...).

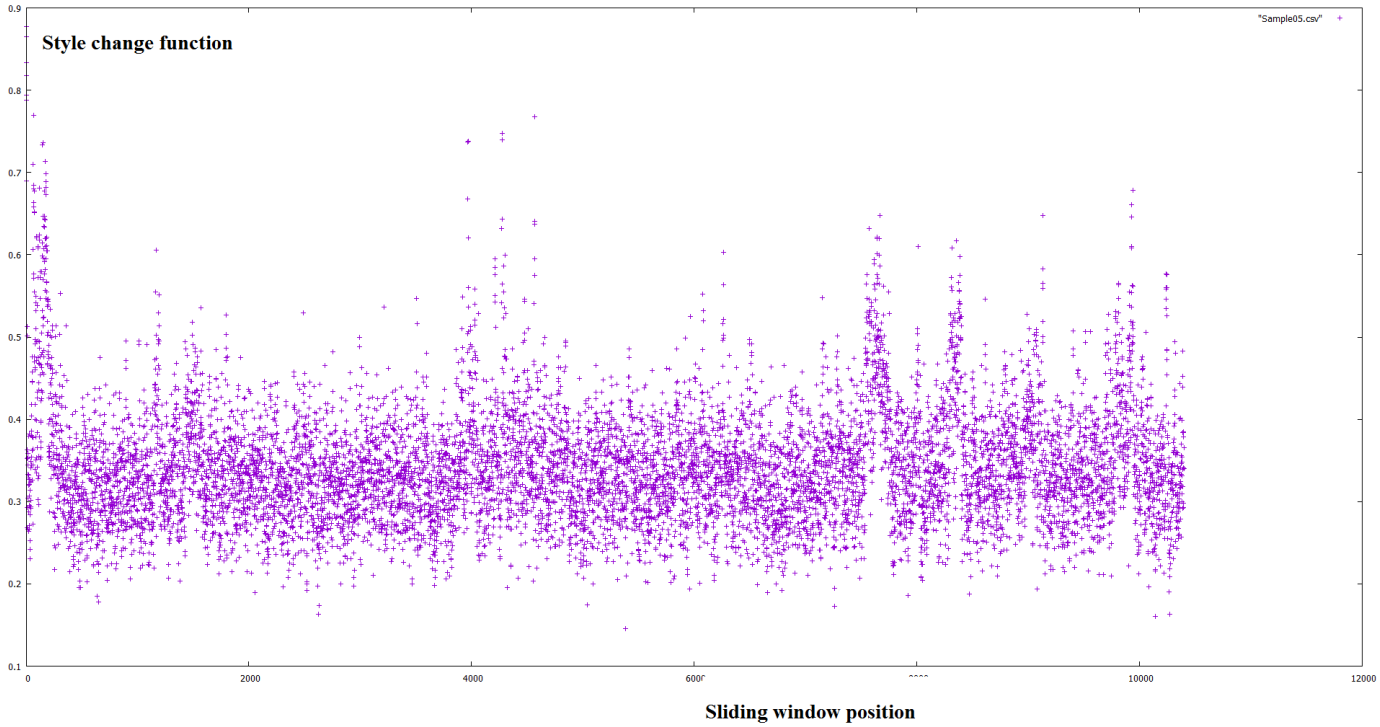

Figure 4.3: The style change function of document 00005.

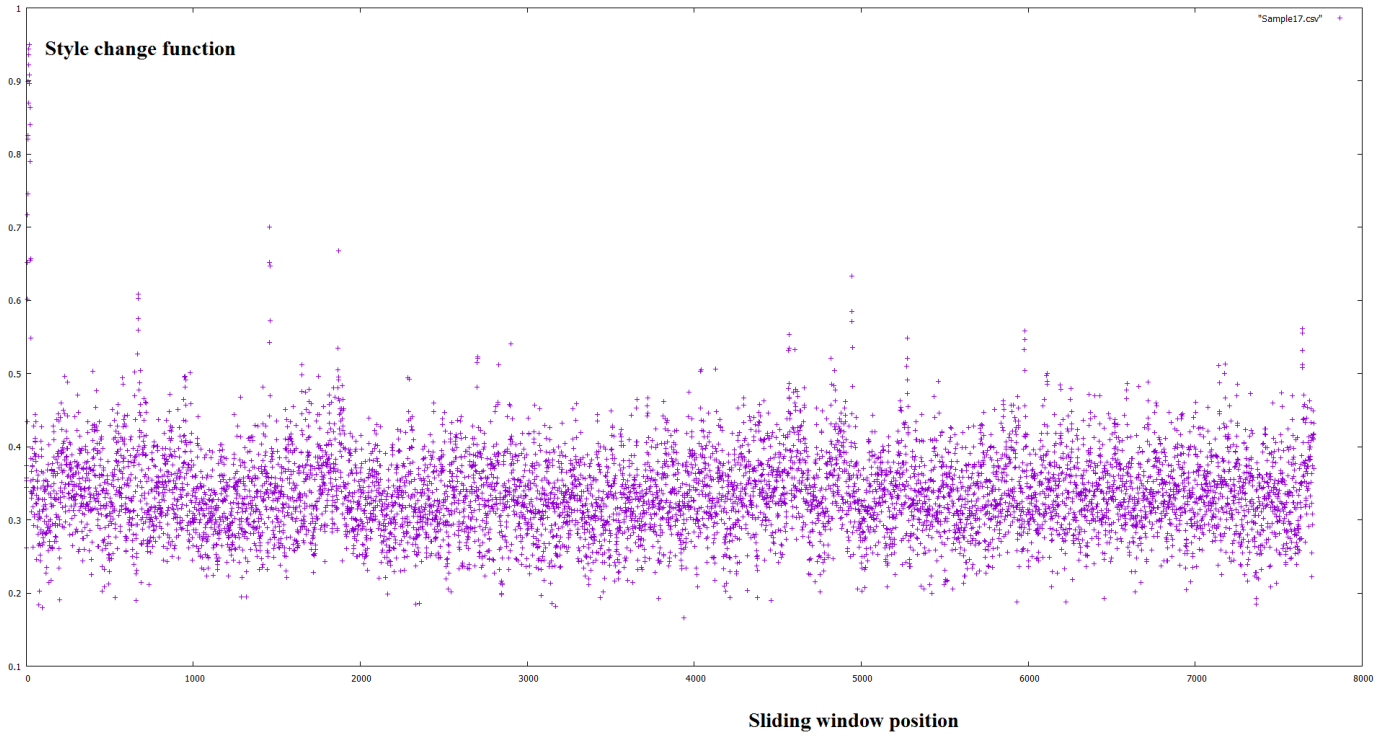

Figure 4.4: The style change function of document 00017

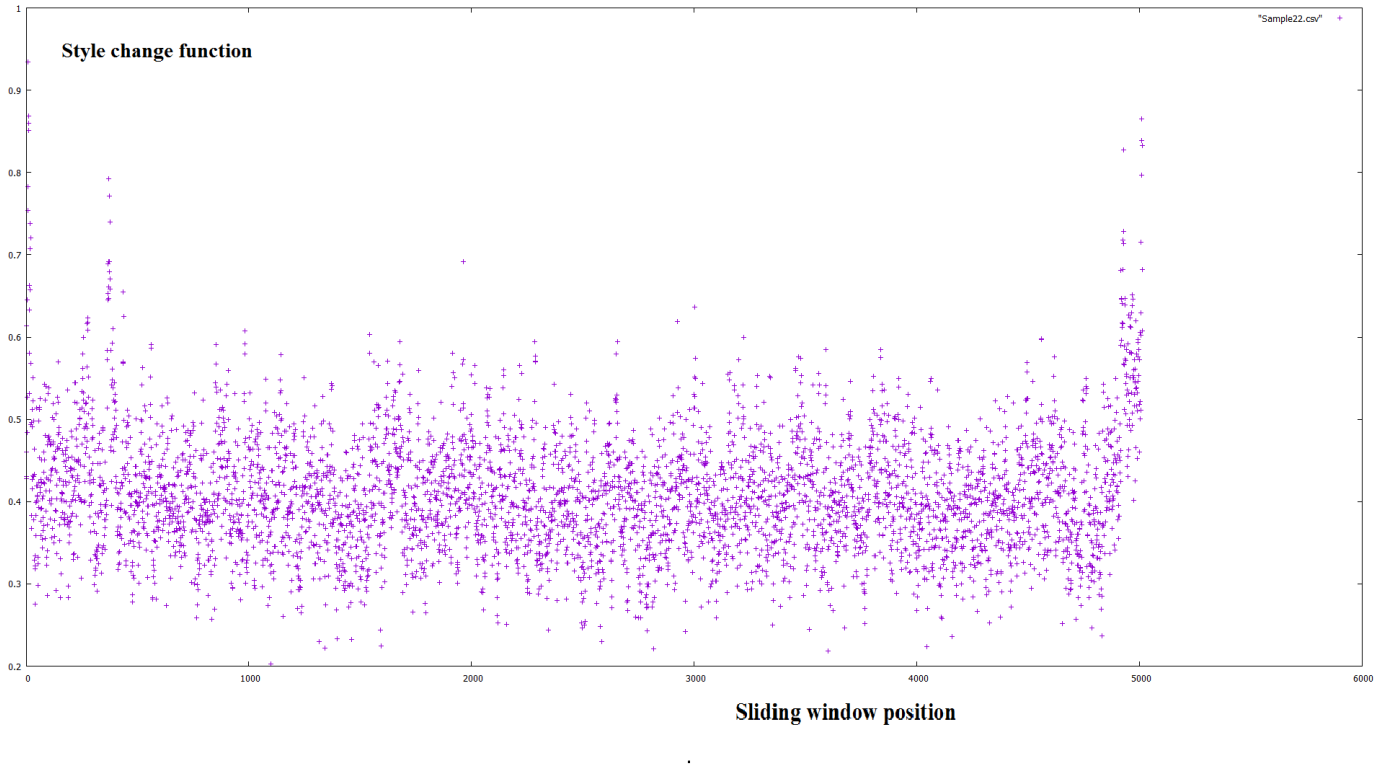

Figure 4.5: The style change function of the plagiarism-free document 00022

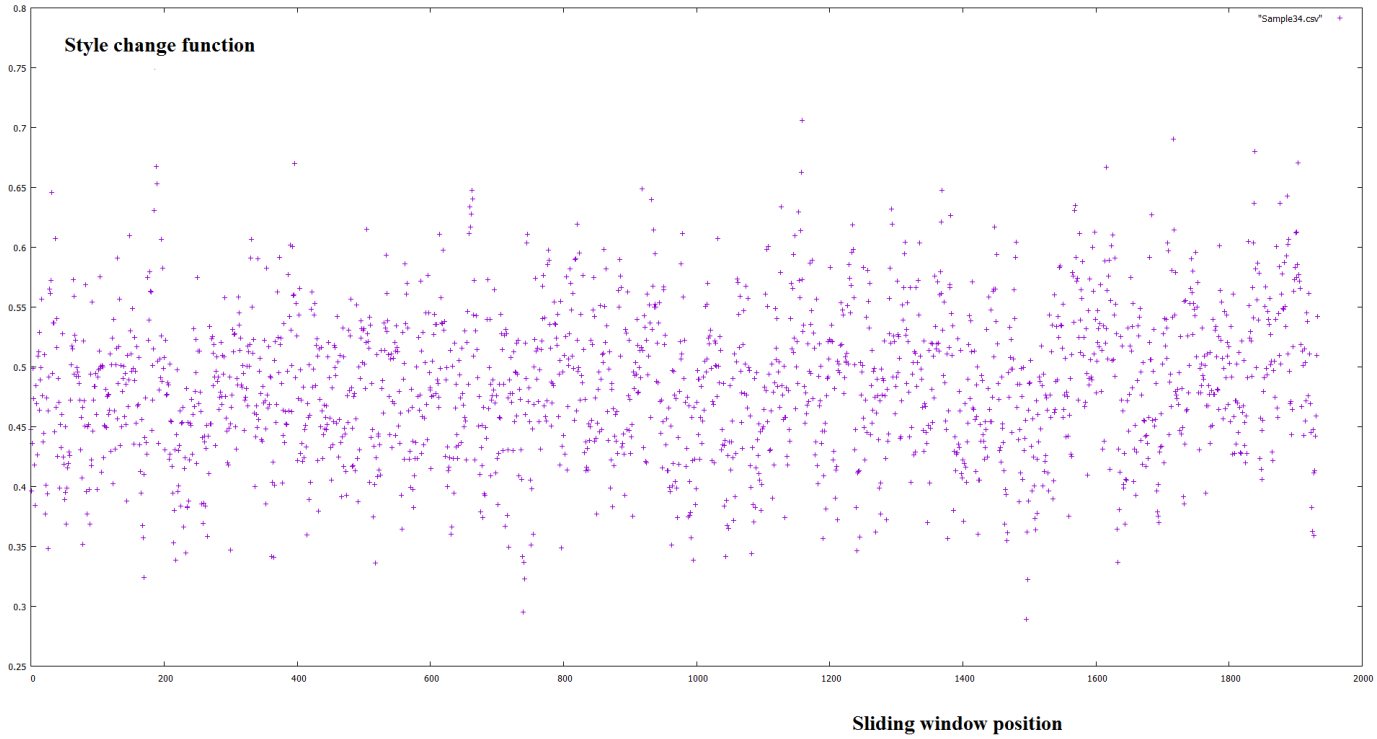

Figure 4.6: The style change function of document 00034

Conclusion

Due to the digital era, the volume of digital resources has been increasing in the World Wide Web enormously. Today, With the rapid access to these digital resources, the possibility of copyright violation and plagiarism has also been increasing simultaneously.

To address this problem, researchers started working on plagiarism detection since 1990.
In our work, we have tried to study in depth the problem of plagiarism, and as the detection of plagiarism is a field of data mining, we start with the definition of data mining, their tasks, techniques and applications. Then, we talked about plagiarism, its classification, detection, methods and its evaluation. Then, we spoke about different research works on extrinsic and intrinsic plagiarism detection methods, and finally, we complete of our study with an implementation of an intrinsic plagiarism detection methods which is based on 3-gram character. The obtained results which are represented on the graphs are almost similar to the results obtained in PAN09. The precision and recall of this system are not yet obtained because of the required efficiency in the material that we used in the calculations and the time shortage. In order to enhance we will explore the use another formula of similarity calculation which is based on sequences, or semantic attributes.

Bibliography

[1] Emmanuel M. Braverman Aizerman, Mark A. and Lev I. Rozonoér. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control 25:821–837*, 1964.

[2] Salha Alzahrani and Naomie Salim. Fuzzy semantic-based string similarity for extrinsic plagiarism detection lab report for pan at clef 2010.

[3] Salha Alzahrani, Naomie Salim, and Ajith Abraham. Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Trans. Systems, Man, and Cybernetics, Part C*, 42(2):133–149, 2012.

[4] Alberto Barrón-Cedeño. On the mono- and cross-language detection of text reuse and plagiarism. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 914–914, New York, NY, USA, 2010. ACM.

[5] Alberto Barrón-Cedeño, Chiara Basile, Mirko Degli Esposti, and Paolo Rosso. Word length n-grams for text re-use detection. In *Computational Linguistics and Intelligent Text Processing, 11th International Conference, CICLing 2010, Iasi, Romania, March 21-27, 2010. Proceedings*, pages 687–699, 2010.

[6] Alberto Barrón-Cedeño, Parth Gupta, and Paolo Rosso. Methods for cross-language plagiarism detection. *Knowl.-Based Syst.*, 50:211–217, 2013.

[7] Alberto Barrón-Cedeño and Paolo Rosso. On automatic plagiarism detection based on n-grams comparison. In *Advances in Information Retrieval, 31th European Conference on IR Research, ECIR 2009, Toulouse, France, April 6-9, 2009. Proceedings*, pages 696–700, 2009.

[8] Alberto Barrón-Cedeño, Paolo Rosso, David Pinto, and Alfons Juan. 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, July 22, 2008*, 2008.

[9] Chiara Basile, Dip Matematica, Dario Benedetto, Emanuele Caglioti, Giampaolo Cristadoro, and Mirko Degli Esposti. Caglioti e.: A plagiarism detection procedure in three steps: selection, matches and 'squares. In *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09*, pages 1–9, 2009.

[10] Philip Bille, Patrick Hagge Cording, Johannes Fischer, and Inge Li Gørtz. Lempel-ziv compression in a sliding window. In *28th Annual Symposium on Combinatorial Pattern Matching, CPM 2017, July 4-6, 2017, Warsaw, Poland*, pages 15:1–15:11, 2017.

[11] M. Popescu C. Grozea, C. Gehl. Encoplot: Pairwise sequence matching in linear time applied. In *3rd PAN WORKSHOP. UNCOVERING PLAGIARISM, AUTHORSHIP AND SOCIAL SOFTWARE MISUSE. 25th ANNUAL CONFERENCE OF THE SPANISH SOCIETY FOR NATURAL LANGUAGE PROCESSING, SEPLN 2009*.

[12] Tommy W. S. Chow and M. K. M. Rahman. Multilayer som with tree-structured data for efficient document retrieval and plagiarism detection. *Trans. Neur. Netw.*, 20(9):1385–1402, September 2009.

[13] Hussain A. Chowdhury and Dhruba K. Bhattacharyya. Plagiarism: Taxonomy, tools and detection techniques. *CoRR*, abs/1801.06323, 2018.

[14] Prabhakar Raghavan Christopher D. Manning and Hinrich Schütze. *An Introduction to Information Retrieval*. Cambridge University Press Cambridge, England, April 1, 2009.

[15] Fintan Culwin and Thomas Lancaster. Plagiarism, prevention, deterrence and detection, 2000.

[16] Kenneth D. Butterfield Donald L. McCabe, Linda Klebe Treviño. Cheating in academic institutions: A decade of research. Technical report, 2001.

[17] Margaret H. Dunham. *Data Mining- Introductory and Advanced Concepts*. Pearson Education, 2006.

[18] Mohamed Elhadi and Amjad Al-Tobi. Detection of duplication in documents and webpages based documents syntactical structures through an improved longest common subsequence. *IJIPM*, 1(1):138–147, 2010.

[19] EL Habib Benlahmar Faouzia Benabbou. Comparaison des techniques de détection du plagiat académique. In *4ème Journée sur les Technologies d'Information et de Modélisation TIM'16*, June 2016.

[20] Usama Fayyad, Gregory Piatetsky-shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

[21] Yongjian Fu. Data mining: Tasks, tchniques, and applications. *University of Missouri - Rolla*.

[22] Bela Gipp. *Citation-based Plagiarism Detection - Detecting Disguised and Cross-language Plagiarism using Citation Pattern Analysis*. Springer, 2014.

[23] Bela Gipp and Jöran Beel. Citation based plagiarism detection: a new approach to identify plagiarized work language independently. In *HT'10, Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, Toronto, Ontario, Canada, June 13-16, 2010*, pages 273–274, 2010.

[24] Bela Gipp and Norman Meuschke. Citation pattern matching algorithms for citation-based plagiarism detection: greedy citation tiling, citation chunking and longest common citation sequence. In *Proceedings of the 2011 ACM Symposium on Document Engineering, Mountain View, CA, USA, September 19-22, 2011*, pages 249–258, 2011.

[25] Bela Gipp, Norman Meuschke, and Joeran Beel. Comparative evaluation of text- and citation-based plagiarism detection approaches using guttenplag. In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, JCDL '11, pages 255–258, New York, NY, USA, 2011. ACM.

[26] Michael Goebel and Le Gruenwald. A survey of data mining and knowledge discovery software tools. *SIGKDD Explorations*, 1(1):20–33, 1999.

[27] O. Halvani. Register genre seminar: Towards intrinsic plagiarism detection. Technical report, February 2015.

[28] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques, 3rd edition*. Morgan Kaufmann, 2011.

[29] D. M. Hawkins. Identification of outliers. *Chapman and Hall.*, London, 1980.

[30] A. Honore. Some simple measures of richness of vocabulary. association for literary and linguistic computing bulletin,. 2005.

[31] John Houvardas and Efstathios Stamatatos. N-gram feature selection for authorship identification. In *Artificial Intelligence: Methodology, Systems, and Applications, 12th International Conference, AIMSA 2006, Varna, Bulgaria, September 12-15, 2006, Proceedings*, pages 77–86, 2006.

[32] Ludek Hrebícek. Zipf's law and text. *Glottometrics*, 3:27–38, 2002.

[33] Sangeetha Jamal. Seminar report on plagiarism detection techniques. Technical report, 2010.

[34] Nik'e van Vugt Jurriaan Hage, Peter Rademaker. A comparison of plagiarism detection tools. Technical report, 2010.

[35] M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14:10–25, 1963.

[36] Mike Kestemont, Kim Luyckx, and Walter Daelemans. Intrinsic plagiarism detection using character trigram distance scores - notebook for PAN at CLEF 2011. 2011.

[37] Moshe Koppel and Jonathan Schler. Authorship verification as a one-class classification problem. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, 2004.

[38] T. Lancaster. *Effective and Efficient Plagiarism Detection*. London South Bank University, 2003.

[39] Yuhua Li, David McLean, Zuhair Bandar, James O'Shea, and Keeley A. Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE Trans. Knowl. Data Eng.*, 18(8):1138–1150, 2006.

[40] Caroline Lyon, James A. Malcolm, and Bob Dickerson. Detecting short passages of similar text in large document collections. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2001, Pittsburgh, PA USA, June 3-4, 2001*, 2001.

[41] Vítor T. Martins, Daniela Fonte, Pedro Rangel Henriques, and Daniela da Cruz. Plagiarism Detection: A Tool Survey and Comparison. In Maria João Varanda Pereira, José Paulo Leal, and Alberto Simões, editors, *3rd Symposium on Languages, Applications and Technologies*, volume 38 of *OpenAccess Series in Informatics (OASIcs)*, pages 143–158, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[42] Hermann A. Maurer, Frank Kappe, and Bilal Zaka. Plagiarism - A survey. *J. UCS*, 12(8):1050–1084, 2006.

[43] Paul McNamee and James Mayfield. Character n-gram tokenization for european language text retrieval. *Information Retrieval*, 7(1-2):73–97, 2004.

[44] N. Meuschke and B. Gipp. *State-of-the-art in Detecting Academic Plagiarism*. 2015.

[45] Daniel Micol, Óscar Ferrández, Fernando Llopis, and Rafael Muñoz. A textual-based similarity approach for efficient and scalable external plagiarism analysis - lab report for PAN at CLEF 2010. In *CLEF 2010 LABs and Workshops, Notebook Papers, 22-23 September 2010, Padua, Italy*, 2010.

[46] Dr.S.B.Kishor Nilesh Channawar. Implicit ascertain for plagiarism detection and text classification. *Indian J.Sci.Res. 17(2): 163-167, 2018*, 2018.

[47] Gabriel Oberreuter and Juan D. Velásquez. Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style. *Expert Syst. Appl.*, 40(9):3756–3763, 2013.

[48] Josephson Institute of Ethics. Report card on the ethics of american youth. Technical report, 2012.

[49] Ahmed Hamza Osman, Naomie Salim, and Mohammed Salem Binwahlan. Plagiarism detection using graph-based representation. *CoRR*, abs/1004.4449, 2010.

[50] Slimane Oulad-Naoui. *Fouille de motifs : formalisation et unification. (Pattern Mining: Formalisation and Unification)*. PhD thesis, University of Laghouat, Algeria, 2018.

[51] Fuchun Peng, Dale Schuurmans, Vlado Keselj, and Shaojun Wang. Language independent authorship attribution with character level n-grams. In *EACL 2003, 10th Conference of the European Chapter of the Association for Computational Linguistics, April 12-17, 2003, Agro Hotel, Budapest, Hungary*, pages 267–274, 2003.

[52] Rafael Corezola Pereira, Viviane Pereira Moreira, and Renata Galante. A new approach for cross-language plagiarism analysis. In *Multilingual and Multimodal Information Access Evaluation, International Conference of the Cross-Language Evaluation Forum, CLEF 2010, Padua, Italy, September 20-23, 2010. Proceedings*, pages 15–26, 2010.

[53] Martin Potthast, Alberto Barrón-Cedeño, Benno Stein, and Paolo Rosso. Cross-language plagiarism detection. *Lang. Resour. Eval.*, 45(1):45–62, March 2011.

[54] Martin Potthast, Andreas Eiselt, Alberto Barrón-Cedeño, Benno Stein, and Paolo Rosso. Overview of the 3rd international competition on plagiarism detection. 2011.

[55] Martin Potthast, Benno Stein, and Maik Anderka. A wikipedia-based multilingual retrieval model. In *Advances in Information Retrieval , 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings*, pages 522–530, 2008.

[56] Martin Potthast, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. An evaluation framework for plagiarism detection. In *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 997–1005, 2010.

[57] Martin Potthast, Benno Stein, Andreas Eiselt, Bauhaus universität Weimar, Alberto Barrón-cedeño, and Paolo Rosso. P.: Overview of the 1st international competition on plagiarism detection. In *In: SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09), CEUR-WS.org*, pages 1–9, 2009.

[58] M. K. M. Rahman and Tommy W. S. Chow. Content-based hierarchical document organization using multi-layer hybrid network and tree-structured features. *Expert Syst. Appl.*, 37(4):2874–2881, 2010.

[59] Sameer Rao, Parth Gupta, Khushboo Singhal, and Prasenjit Majumder. External & intrinsic plagiarism detection: VSM & discourse markers based approach - notebook for PAN at CLEF 2011. 2011.

[60] Saul Schleimer, Daniel S. Wilkerson, and Alex Aiken. Winnowing: Local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, pages 76–85, New York, NY, USA, 2003. ACM.

[61] Leanne Seaward and Stan Matwin. Intrinsic plagiarism detection using complexity analysis. *University of Ottawa 2096 Madrid Avenue, Ottawa, ON,K2J 0K4*, 2009.

[62] Vladislav Shcherbinin and Sergey Butakov. Using microsoft sql server platform for plagiarism detection.

[63] Jason Sorensen. A competitive analysis of automated authorship attribution techniques. 2005.

[64] Efstathios Stamatatos. Intrinsic plagiarism detection using character n-gram profiles. *University of the Aegean 83200 - Karlovassi, Samos, Greece*, 2009.

[65] Benno Stein and Sven Eissen. Near similarity search and plagiarism analysis. *From Data and Information Analysis to Knowledge Engineering*, pages 430–437, 2006.

[66] Benno Stein, Sven Meyer zu Eissen, and Martin Potthast. Strategies for retrieving plagiarized documents. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 825–826, 2007.

[67] Benno Stein, Sven Meyer zu Eissen, and Martin Potthast. Strategies for retrieving plagiarized documents. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 825–826, 2007.

[68] Victor Thompson. Methods for detecting paraphrase plagiarism. *CoRR*, abs/1712.10309, 2017.

[69] Özlem Uzuner, Boris Katz, and Thade Nahnsen. Using syntactic information to identify plagiarism. In *Proceedings of the Second Workshop on Building Educational Applications Using NLP*, EdAppsNLP 05, pages 37–44, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[70] Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998.

[71] Rui Xu and D. Wunsch, II. Survey of clustering algorithms. *Trans. Neur. Netw.*, 16(3):645–678, May 2005.

[72] Rajiv Yerra and Yiu-Kai Ng. A sentence-based copy detection approach for web documents. In *Fuzzy Systems and Knowledge Discovery, Second International Conference, FSKD 2005, Changsha, China, August 27-29, 2005, Proceedings, Part I*, pages 557–570, 2005.

[73] G.U. Yule. *The Statistical Study of Literary Vocabulary*. Archon Books, 1968.

[74] Haijun Zhang, Gang Liu, Tommy W. S. Chow, and Wenyin Liu. Textual and visual content-based anti-phishing: A bayesian approach. *IEEE Trans. Neural Networks*, 22(10):1532–1546, 2011.

[75] Sven Meyer zu Eissen and Benno Stein. Intrinsic plagiarism detection. In *Advances in Information Retrieval, 28th European Conference on IR Research, ECIR 2006, London, UK, April 10-12, 2006, Proceedings*, pages 565–569, 2006.

[76] Sven Meyer zu Eissen, Benno Stein, and Marion Kulig. Plagiarism detection without reference collections. In *Advances in Data Analysis, Proceedings of the 30th Annual Conference of the Gesellschaft für Klassifikation e.V., Freie Universität Berlin, March 8-10, 2006*, pages 359–366, 2006.