Automatic evaluation of existing plagiarism detection tools

Siwar Nadhri, Maryam Elamine and Lamia Hadrich Belguith

University of Sfax, MIRACL Research Laboratory, Sfax, Tunisia

Abstract

The vast expansion of data over the Internet, as well as the ease with which people may access it, has resulted in several issues, including authorship attribution, copyrights, plagiarism, etc. Indeed, plagiarism is an increasing problem among various domains mainly in journalism, politics, academia, etc. Plagiarism is the act of attributing to oneself the work of another without citing the original source. Consequently, plagiarism detection tools are emerging. Nevertheless, the choice of the most effective tool remains a serious matter for the users. Thus, in this paper, we present our proposed method for automatically evaluating plagiarism detection systems. Hence, we tested three existing tools: WCopyfind, Compare It and Compare Suite and observed their behavior on a French and English corpora. The preliminary results indicate the superiority of Compare Suite, accuracy wise, and of Compare It in execution time. We also remarked the wide difference in the comportment of the tools using French and English corpora.

Keywords

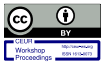
Extrinsic plagiarism detection, plagiarism detection tools, automatic evaluation, software testing.

1. Introduction

The growth of the media has made it possible to obtain a large amount of data [8, 4]. In fact, information technology has evolved rapidly in the last decade [3]. The easy availability of the Internet poses a danger to information integrity and every data can be plagiarized [7]. Plagiarism is unethical and a serious offence [1]. It comprises a danger to the instructive cycle since understudies might get acknowledgment for another person's work or complete courses without really accomplishing the ideal learning results [11]. Actually, plagiarists use different methods to shroud their illegal activities, like revising parts of the copied text, changing a few words with their equivalents, and so forth [13]. Recognizing plagiarism is an everlasting concern inside universities, and recent years have witnessed surprising advances in plagiarism detection tools [5]. Anti-plagiarism tools, also known as text-matching tools, are expected to use state-of-the-art methods to detect plagiarism. Current systems are rather great at identifying copy/paste cases. Nevertheless, with the variety of forms of plagiarism ranging from a simple reformulation to a complex level of obfuscation including translation, the capability of these tools is always put to question [6, 11]. The incapacity of anti-plagiarism tools can be worrying, especially, since a modern research reported that 70% of students have confessed to plagiarizing, with about half being guilty of an earnest cheating offence on a written assignment [2].

Our aim in this paper is to test the capability of three free existing tools, namely: WCopyfind[2] , Compare It [3]and Compare Suite[4]. The choice of these three systems is because they are open access, free and function on an offline database unlike other free tools that only search the Internet for possible plagiarism cases. We tested them on a French corpus that we created from simple copy/paste cases and

Tunisian Algerian Conference on Applied Computing (TACC 2021), December 18–20, 2021, Tabarka, Tunisia
EMAIL : siwar.nadhri1@gmail.com (S. Nadhri); mary.elamine@gmail.com (M. Elamine); lamia.belguith@fsegs.usf.tn (L. H. Belguith)



[2] https://plagiarism.bloomfieldmedia.com/software/wcopyfind/
[3] https://www.grigsoft.com/wincmp3.htms
[4] https://comparesuite.com/

an English corpus provided by the PAN@CLEF[5] competition containing cases of obfuscation ranging from simple to complex levels. Our experiments show that Compare It is the fastest of the three systems. Compare Suite is the slowest, but it is the most efficient. All three programs have a serious encoding problem with French language. This inspired us to experiment another language (English). Despite the fact that the English corpus contains various levels of obfuscation, preliminary results prove that the three plagiarism detection tools functioned better using the before mentioned corpora, rather than the French corpus, which was mostly comprised of copy/paste cases.

The remainder of this paper is organized as follows. Section 2 presents a literature review. Section 3 describes our proposed method followed by the presentation of our experiments and results in Section 4. Finally, Section 5 gives some concluding remarks followed by future work directions.

2. Related Works

Since the turn of the century, not only has the subject of plagiarism been highly appreciated, but also has text-matching software, which is utilized to discover suspected plagiarized passages in manuscripts. Many scientific publications have discussed text matching software solutions in terms of classification, comparative research, overview, and comparison.

In their work, [11] have defined two comparison criteria for the evaluation of 15 tools tested using documents in eight languages. The first criterion is the coverage of the tool, that ranges on a scale of 0 (worst) to 5 (best). It comprises four main requirements, which are: language comparison (i.e., the languages covered by the tools), types of plagiarism sources (Wikipedia extracts, open-access papers, student theses, and online documents), plagiarism forms (copy-paste, synonym replacement, manual paraphrase, translation) and plagiarism detection based on a single-source or multi-source documents. The second criterion is the usability of the tool, i.e., collecting the viewpoint of end-users; it is based on a sum of points (0, 0.5 or 1 point). After many experiments, the authors concluded that the tools' usability performance is superior to their coverage performance because they do not detect all text similarity and suffer from false positives.

Shkodkina and Pacauskas [12] compared three plagiarism detection systems. They tested the tools based on a set of criteria and features, in the academic context in Ukraine. The compared tools are Unicheck[6] , eTXT[7] and Turnitin[8] . The authors chose these systems since they are available in Ukraine. The authors suggested four criteria, where each of them has a set of features: (1) affordability, (2) material support, (3) functionality, and (4) showcasing. The authors enumerated some assets and handicaps of each program and concluded that eTXT is more appropriate for personal use, while either Turnitin or Unicheck are more suitable for institutional use. Specifically, Unicheck appears to be one of the most appropriate and efficient systems for Ukrainian universities.

In their investigation, [10] completed a comparative analysis of five systems using the same eight articles in two test series. The first test comprised articles that had not been altered; the second test included articles that had been manually modified by rearranging terms in the text. The percentage of plagiarism discovered and the time spent by the systems checking the articles were the main focus of their investigation. Then, the authors employed a multi-criteria decision-making for choosing the best system. However, they did not give a clear indication of the comparison purpose or how much plagiarism was discovered by the systems. They also looked at usability through the lens of a criterion called "additional support", which included the ability to alter content directly on the website and multilingual checking.

In his study, [9] tested a variety of plagiarism detection tools. The author categorized them into free and non-free systems. He compared the most popular tools from each category. For the first category, the author concluded that the features of free plagiarism detection tools range from one another, therefore, it is best to try them all in order to choose the ideal one for individual necessities. As for the

[5] https://pan.webis.de/
[6] https://unicheck.com/
[7] https://www.etxt.biz/?lang=en
[8] https://www.turnitin.com/

second category, according to the author, the best thorough plagiarism detection tool is iThenticate[9] . In any case, it is likewise the most costly for individual users. Nevertheless, universities, research focuses and associations can manage the cost of the significant expense of the program.

In addition to scientific research, some anti-plagiarism tools conduct, in a quest for self-evaluation, an investigation on the performance of the existing plagiarism detection systems. In 2019, Scribbr[10] , a paid anti-plagiarism software, conducted a study to compare its performance against that of other systems. The comparative analysis involved 10 plagiarism detection tools (paid and free). The study included two forms of plagiarism: direct (by using a 100% plagiarized document with extracts from magazines, books and Internet sites) and dispersed (by using a real document with original paragraphs and 50% plagiarized segments). Table 1 presents the best anti-plagiarism software for 2019 according to the evaluation conducted by Scribbr[11] .

Table 1
The top 10 anti-plagiarism software of 2019

Plagiarism detection tool	Identified plagiarism for a 50% plagiarized document	Identified plagiarism for a 100% plagiarized document	Overall Accuracy
Scribbr	44 %	75 %	★★★★★
Ephorus[12]	23 %	61 %	★★★★★
Quetext[13]	29 %	53 %	★★★★½
Compilatio[14]	28 %	51 %	★★½☆☆
BibMe[15]	19 %	57 %	★★★☆☆
Plagscan[16]	17 %	58 %	★★★☆☆
Plagramme[17]	16 %	61 %	★★★☆☆
Grammarly[18]	0 %	24 %	★★★☆☆
Smallseotools[19]	5 %	28 %	★☆☆☆☆
SE Reports[20]	4 %	34 %	★☆☆☆☆

3. Proposed Method

In this section, we present our proposed method for evaluating plagiarism detection tools. Our proposed method comprises five main steps namely: Document analysis, output unification, output tagging, post-processing and tools evaluation process (Figure 1).

[9] https://www.ithenticate.com/
[10] https://www.scribbr.fr/logiciel-anti-plagiat/
[11] https://www.scribbr.fr/le-plagiat/meilleur-logiciel-anti-plagiat/
[12] https://www.ephorus.com/
[13] https://www.quetext.com/
[14] https://www.compilatio.net/
[15] https://www.easybib.com/grammar-and-plagiarism/
[16] https://www.plagscan.com/fr/
[17] https://www.plagramme.com/
[18] https://www.grammarly.com/plagiarism-checker
[19] https://smallseotools.com/plagiarism-checker/
[20] https://searchenginereports.net/plagiarism-checker

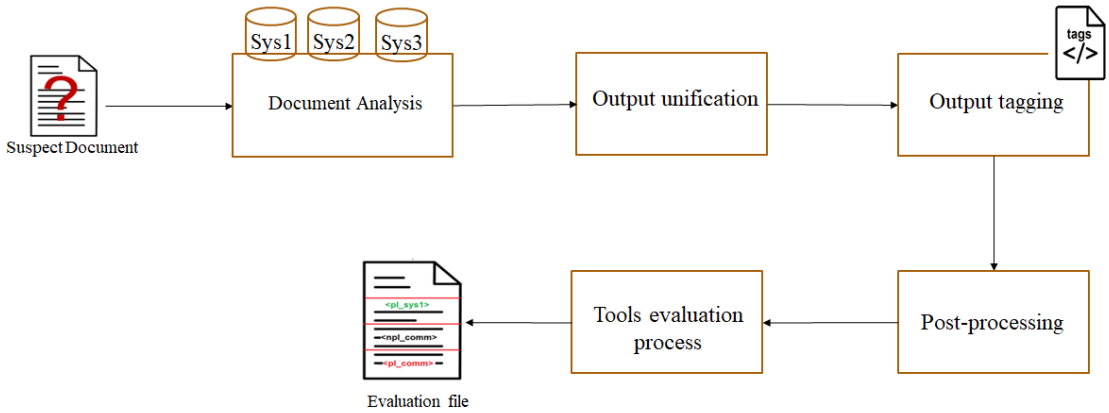


Figure 1: Main steps of our proposed method

3.1. Corpus creation

In our work, we aspire to, automatically, be able to distinguish the best tool to identify plagiarism cases in academia. Indeed, we decided to work in French because it is the language in which the majority of student reports are written at our universities. Thus, we created a French corpus inspired by PAN-PC-09 corpora[21]. We gathered a set of 200 documents from the site "Thèses.fr[22]". Although the documents are all in French, they are multi-genres (Economics, Computer Science, Physics, etc.). The collected documents are all in PDF, since we cannot work directly with this file format, we converted all documents into TXT format. Then, after close inspection, we removed a set of encrypted documents (Figure 2). Consequently, in final, we have 140 documents in our corpus. In fact, for the purpose of our experiments, we created a set of 30 plagiarized documents from the collection of 140 documents. These documents were the fruit of copy/paste cases. No reformulation or obfuscation was done. In fact, our corpus comprises an average of 75608 words per document. As for the created Fake documents, they contain, approximately, between 10000 and 20000 words each. We also created, for each Fake document, an XML file comprising a thorough description of the source of plagiarism as well as the start and the end of plagiarism. Figure 3 presents an example of an XML description of a plagiarized document from our corpus.

(cid:17)(cid:19)(cid:18)(cid:20)(cid:17)

(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)(cid:26)(cid:18)

(cid:17)(cid:19)(cid:18)(cid:30)(cid:29)

(cid:18)(cid:27)(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)

(cid:17)(cid:19)(cid:18)(cid:30)(cid:29)(cid:31)(cid:18) (cid:17)

(cid:18)(cid:28)(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)(cid:26)(cid:18)

(cid:17)(cid:19)(cid:18)(cid:30)(cid:29)(cid:31)(cid:18)#(cid:29)

(cid:18)(cid:27)(cid:18)(cid:27)(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)(cid:28)(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)(cid:26)(cid:18)(cid:27)(cid:18)

Figure 2: Example of an encrypted document

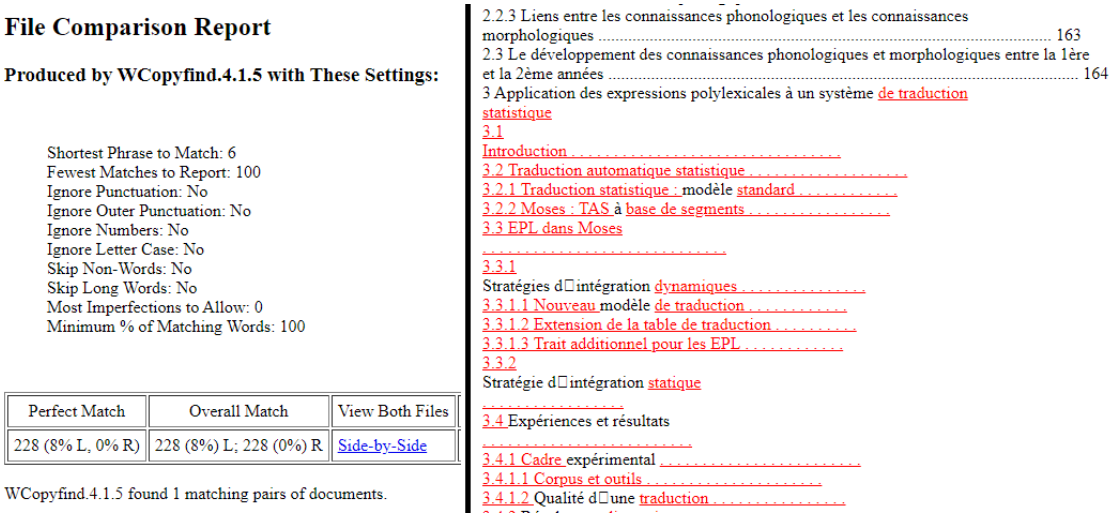
[21] https://webis.de/data/pan-pc-09.html
[22] http://www.theses.fr/

```
<? xml vesion="1.0" encoding="UTF-8" ?>
<Doc_pl_desc name="Plagiat-1.txt">
    <offset_Pl src_doc name="manuscrit0090.txt" start_src="1" finish_src="106" start_pl="1" finish_pl="106"/>
    <offset_Pl src_doc name="manuscrit0080.txt" start_src="193" finish_src="280" start_pl="108" finish_pl="195"/>
    <offset_Pl src_doc name="manuscrit0090.txt" start_src="177" finish_src="251" start_pl="201" finish_pl="278"/>
    <offset_Pl src_doc name="manuscrit00123.txt" start_src="603" finish_src="697" start_pl="282" finish_pl="376"/>
    <offset_Pl src_doc name="manuscrit00100.txt" start_src="283" finish_src="362" start_pl="379" finish_pl="458"/>
    <offset_Pl src_doc name="manuscrit00123.txt" start_src="1167" finish_src="1375" start_pl="461" finish_pl="670"/>
    <offset_Pl src_doc name="manuscrit0080.txt" start_src="2059" finish_src="2558" start_pl="673" finish_pl="1172"/>
    <offset_Pl src_doc name="manuscrit0089.txt" start_src="3615" finish_src="3935" start_pl="1175" finish_pl="1199"/>
    <offset_Pl src_doc name="manuscrit00100.txt" start_src="1879" finish_src="1884" start_pl="1201" finish_pl="1204"/>
</Doc_pl_desc>
```

Figure 3: Example of an XML description for a plagiarized document from our corpus

3.2. Document Analysis

In this step, we analyzed the suspect document with the tools we aim to compare: WCopyfind, Compare It and Compare Suite. Each system generated an HTML output that we will be using in further steps. For the first tool, the output consists of an HTML file comprising the chosen settings, the source(s) of plagiarism and the similarity rate. Any plagiarism is colored in red or black, otherwise (Figure 4).



File Comparison Report

Produced by WCopyfind.4.1.5 with These Settings:

Shortest Phrase to Match: 6
Fewest Matches to Report: 100
Ignore Punctuation: No
Ignore Outer Punctuation: No
Ignore Numbers: No
Ignore Letter Case: No
Skip Non-Words: No
Skip Long Words: No
Most Imperfections to Allow: 0
Minimum % of Matching Words: 100

Perfect Match	Overall Match	View Both Files
228 (8% L, 0% R)	228 (8%) L; 228 (0%) R	Side-by-Side

WCopyfind.4.1.5 found 1 matching pairs of documents.

2.2.3 Liens entre les connaissances phonologiques et les connaissances morphologiques 163
2.3 Le développement des connaissances phonologiques et morphologiques entre la 1ère et la 2ème années 164
3 Application des expressions polylexicales à un système de traduction statistique
3.1
Introduction . . . . . .
3.2 Traduction automatique statistique . . . . . .
3.2.1 Traduction statistique : modèle standard . . . . . .
3.2.2 Moses : TAS à base de segments . . . . . .
3.3 EPL dans Moses
. . . . . .
3.3.1
Stratégies d□intégration dynamiques . . . . . .
3.3.1.1 Nouveau modèle de traduction . . . . . .
3.3.1.2 Extension de la table de traduction . . . . . .
3.3.1.3 Trait additionnel pour les EPL . . . . . .
3.3.2
Stratégie d□intégration statique
. . . . . .
3.4 Expériences et résultats
. . . . . .
3.4.1 Cadre expérimental . . . . . .
3.4.1.1 Corpus et outils . . . . . .
3.4.1.2 Qualité d□une traduction . . . . . .

Figure 4: Output of WCopyfind

For the second tool, the HTML report contains the analysis' statistic as well as a colored Side-by-Side comparison: the left side refers to the source document, whereas the right is for the suspect document. The black-colored segments point the plagiarized parts (i.e., these parts are for sure plagiarized). Each color, otherwise, is considered non-plagiarism. The red color indicates that these parts are unique. Green is for the parts belonging only to the source. Blue implies that some minor changes to the text have been identified (Figure 5).

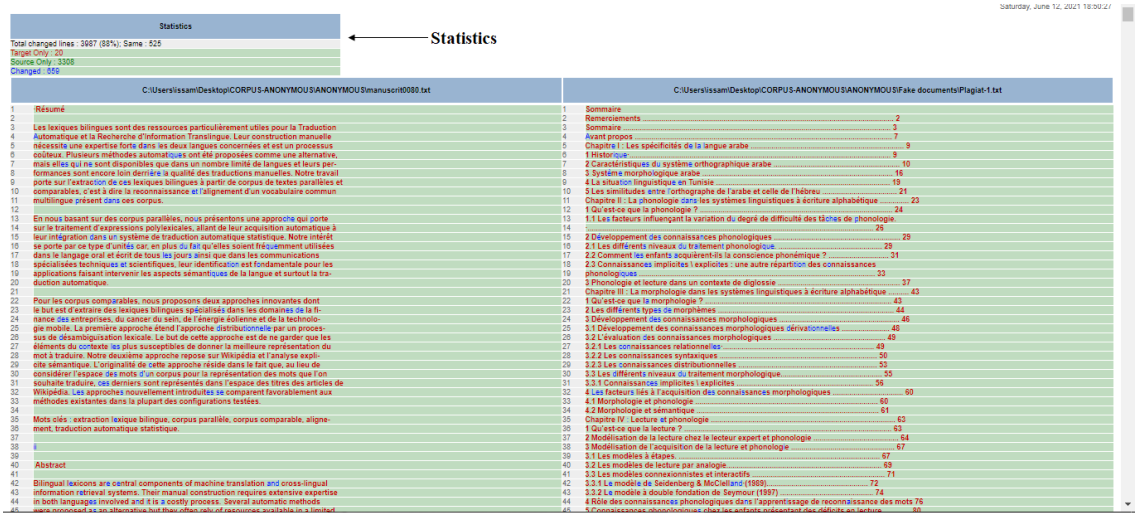


Figure 5: Output of Compare It

Finally, for the third tool, the HTML file comprises the analysis' statistics, the chosen options for the comparison and the colored plagiarism report. The white background designates the presence of plagiarism, a colored one indicates otherwise. The blue background implies that the designated parts are not plagiarized, red stands for parts that belong only to the source document, whereas the green background indicates that a few changes to the text have been detected (Figure6).

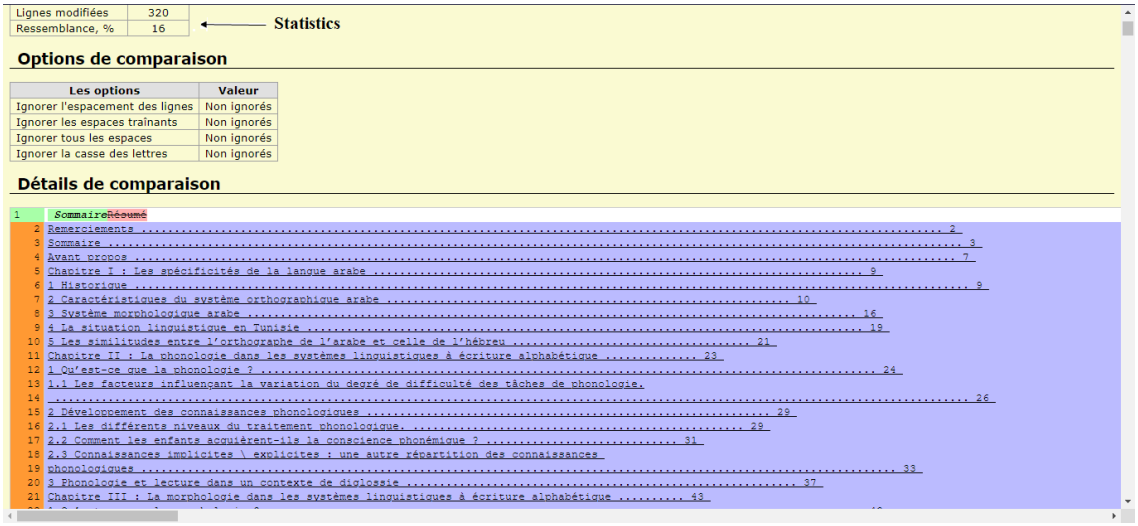


Figure 6: Output of Compare Suite

3.3. Output unification

As previously mentioned, each system has its own output for the same suspect document. In order to proceed our evaluation, we need to clean the output of each tool. For WCopyfind, the HTML code of the file is compressed in one line. Using the
 tag, we split the lines to obtain a structured document. For Compare It, the output contains a Side-by-Side structure: one part indicating the plagiarized document and another for the source. We eliminated the parts referring to the source document, thus keeping only the description of the plagiarized one. As for Compare Suite, we discovered that some of the lines were divided (i.e., a tagged line is written on two or more lines before the end of the tag), therefore, we reorganized the document in such a way that, each line is enclosed in its appropriate opening and closing tag. Then, we removed the parts of the file referring to the source document, in this case the parts with the red background.

3.4. Output Tagging

Following the obtained documents from the unification step, we tagged each output produced from the plagiarism detection algorithms. For the parts identified as plagiarized by the tools, we added the tag <tag_pl>, otherwise we add the tag <tag_npl>. For WCopyfind, we searched for the part in the HTML code indicating the presence of a red font and encased it in the tags " <tag_pl> ... <\tag_pl>" to indicate that this part is plagiarized. We enclosed the remaining parts in the tags " <tag_npl> ... <\tag_npl>". For Compare It, in accordance to the «*classes*» present in the code, we tagged the document correspondingly. Finally, for Compare Suite, following the HTML tags indicating the background of the text, we added our tags (either plagiarized or not).

3.5. Post-processing

We had a lot of encoding issues because we were working with a French corpus. Actually, each of the tools examined produced a lot of noise, including several unusual and encrypted characters such as: "ST, PU2, x92, etc.". Given the nature of the obtained outputs, we cleaned the noise as much as possible. Some of the characters, however, persisted in the documents, requiring a manual intervention; these cases were kept, since our aim is to implement an automatic evaluation. In addition, as part of the post-processing, we also eliminated some of the unnecessary tags such as : <html>, <head>, <body>, etc. As a result, we obtained for each system, a file containing only parts labeled with either <tag_pl> (for the plagiarized parts) or <tag_npl> otherwise. Hereafter, we present in Figure 7 some examples of the encrypted characters. Figures 8 and 9 illustrate an example of system output before and after post-processing, respectively.

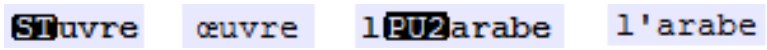


Figure 7: Examples of some encrypted characters

```
<BODY>
<table><tr><td align=right>Saturday, June 12, 2021 18:50:27</td></tr></table>
<table  style="width:30%" cellspacing=1 cellpadding=0>
<thead><tr style="height:20px"><td>Statistics</td></tr></thead>
</table>
<table>
<thead><tr><td width=50%>C:\Users\issam\Desktop\CORPUS-ANONYMOUS\ANONYMOUS\manuscrit0080.txt</td><td width=50%>C:\

<table  cellspacing=1 cellpadding=0>
<tag_npl><tr><td class="LineNum">1</td><td class="InSame"><span class="InDiff"> Sommaire</span></td></tr></ta
<tag_npl><tr><td class="LineNum">2</td><td class="Changed"><span class="InDiff">Remerciements .....
<tag_npl><tr><td class="LineNum">3</td><td class="InSame"><span class="InDiff">Sommaire .....
```

Figure 8: Example of output before post-processing

```
<tag_npl>  Sommaire </tag_npl>
<tag_npl>  Remerciements .....
<tag_npl>  Sommaire .....
<tag_npl> A vant   propos .....
<tag_npl>  Chapitr e  I : Les   spécificités   d e  l a l angue   arabe .....
<tag_npl>  1   Histor ique .....
<tag_npl>  2   Caractéristiqu es  d u  systèm e  orthographique   arabe .....
```

Figure 9: Example of output after post-processing

3.6. Tools evaluation process

In this step, we created an evaluation file tagged accordingly to the plagiarism identified with each of the tools. Thus, we obtained a single document cluttered with tags. Therefore, we performed a reduction to the added tags, i.e., instead of having a collection of consecutive tags, we reduced them in accordance of whether there exists plagiarism cases or not. In what follows, we give an illustrative example: if in a given line, we have <pl_sys3><pl_sys2><pl_sys1> (this indicates that this part is identified as plagiarized by the three tools), we reduced them in one tag <pl_comm>. In case we have

these tags <pl_sys2><pl_sys1>, we reduced them to <pl_sys2_sys1>; this indicates that the first tool ”sys1” and second tool ”sys2” identified this part as plagiarized, whereas the third system did not identify it. Figures 10 and 11 show an example of a file before and after tag reduction.

```
<npl_sys2><npl_sys1>morphologiques ..................... 163</npl_sys1></npl_sys2>
<npl_sys1>2.3 Le développement des connaissances phonologiques et morphologiques </npl_sys1>
<npl_sys1>et la 2ème années ........................... 164</npl_sys1>
<pl_sys2><pl_sys1>statistique</pl_sys1></pl_sys2>
<npl_sys2><pl_sys1>Introduction . . . . . .  . . .</pl_sys1></npl_sys2>
<pl_sys2><pl_sys1>4 Contexte et Matériel</pl_sys1></pl_sys2>
<npl_sys3><pl_sys1>de ressources lexicales. En revanche</pl_sys1></npl_sys3>
<pl_sys3><pl_sys1>e qui soit également bien formée. Le modèle de probabilité </pl_sys1></pl_sys3>
<npl_sys3><pl_sys2><pl_sys1>4.7 Conclusion . . . . . .  . .</pl_sys1></pl_sys2></npl_sys3>
<pl_sys3><pl_sys2><pl_sys1>3.4 Expériences et résultats</pl_sys1></pl_sys2></pl_sys3>
```

Figure 10: Example of an evaluation file after adding the tags in accordance to each tool

```
<pl_sys2_sys1>3 Application des expressions polylexicales à un système de traduction</pl_sys2_sys1>
<pl_sys2_sys1>statistique</pl_sys2_sys1>
<pl_sys2_sys1>3.1</pl_sys2_sys1>
<pl_comm>3.4 Expériences et résultats</pl_comm>
<pl_sys1>Comme il a été mentionné dans le chapitre , les lexiques bilingues constituent un</pl_sys1>
<pl_sys2>système statistique montre une amélioration qui reste petite, dont on ne sait pas si</pl_sys2>
```

Figure 11: Example of an evaluation file after tag reduction

4. Experiments and Results

In this section, we present our experiments and we give our remarks considering the evaluation of the tools. The purpose of our work was to compare existing plagiarism detection tools, and to be able to observe the behavior of each tool. In fact, in our assessment, we created an evaluation document displaying the agreement and disagreement of the experimented tools. Figure 12 gives an example of an evaluation document. In fact, after tagging our documents and creating the evaluation file, we colored each area of the document based on the new tags, using the original document as a reference (the given suspect document). For instance, if all tools have identified the same part as plagiarized, the latter will be colored in Red. If only the first and second tool have identified a part as plagiarized, it will be colored in Pink, etc.

Pl_CompareIt	Pl_CompareSuite	Pl_WCopyfind	Pl_comm	NPl_comm	Pl_compsuit_compit	Pl_wcopy_compsuit	Pl_wcopy_compit
Green	Light Blue	Yellow	Red	Black	Pink	Dark Blue	Light Green

Accuracy of CompareIt	Accuracy of CompareSuite	Accuracy of WCopyfind
84.493 %	96.356 %	10.862 %

3 Application des expressions polylexicales à un système de traduction statistique
3.1
Introduction . . . . . .
L'avantage principal des corpus comparables est leur disponibilité dans di?érentes langues et di?érents domaines de spécialité. C'est pour ces raisons que l'extraction de lexiques bilingues à partir de corpus comparables a attiré notre attention. Cette tâche fera l'objet de la troisième partie de ce manuscrit, où nous présenterons de nouvelles méthodes de création de lexiques bilingues.
Cette partie est organisée comme suit : dans le chapitre 4 nous présentons le contexte global de la tâche d'extraction lexicale à partir de corpus comparables et introduisons les ressources linguistiques utilisées, notamment les corpus comparables sur lesquels nous avons réalisé nos expériences. Le chapitre 5 est centré sur l'approche d'extraction de lexiques bilingues spécialisés à partir de corpus comparables étendant l'approche distributionnelle par l'utilisation de la désambiguisation sémantique. En?n, nous décrivons dans le chapitre 6 l'approche utilisant l'analyse sémantique explicite pour extraire des lexiques bilingues.
Chapitre 4
4.1. Introduction
des points de différences en le comparant avec l'arabe standard moderne.
En effet, dans ce chapitre nous présentons un aperçu sur le dialecte tunisien, son historique ainsi que les principales différences et similitudes avec l'arabe standard. En?n, nous mettons l'accent sur les dif?cultés de l'analyse syntaxique du DT.

Figure 12: Final output for the evaluation of a suspect document with the plagiarism detection tools

As we previously mentioned, we faced many obstacles with the French corpus. This encouraged us to test the plagiarism detection tools on a different language. Consequently, we chose to work with the

English language and we experimented the PAN-PC-09 corpus[23] (PAN Plagiarism Corpus 2009), which is a collection of more than 28000 documents: 14429 source documents (source of plagiarism), collected from Project Gutenberg[24]. A set of 14428 suspicious documents in which artificial plagiarism has been automatically inserted. The plagiarism cases have been constructed using a so-called random plagiarist, i.e., a computer program which constructs plagiarism according to a number of random variables. The variables include the percentage of plagiarism in the whole corpus, the percentage of plagiarism per document, the length of a single plagiarized section and the degree of obfuscation per section. In our work, we experimented 110 suspicious documents, 55 of which included instances of plagiarism. The others are designated as plagiarism suspects; however, they do not contain any plagiarism.

As we carried out our experiments, we discovered that, although the English corpus comprises different levels of obfuscation, the three plagiarism detection tools performed better on this corpus rather than with the French corpus. Nevertheless, all three systems have issues with paraphrased segments. Table 2 presents the evaluation of a plagiarized document with the compared tools.

Table 2
Evaluation of the tested tools

	WCopyfind	Compare It	Compare Suite
License	Free	Free	Free for 30 days
Supported Languages	English, French, Italian, Dutch, German, etc.	No information	No information
Execution time	~ 53.48 seconds	~ 50.4 seconds	~ 548.8 seconds
Accuracy	10.862 %	84.493 %	96.356 %

All three tools are free, with the exception of Compare Suite, which is only free for 30 days. WCopyfind is capable of analyzing documents from different languages: English, Italian, French, Dutch, German, etc. As for the other tools, we have no clear information on the supported languages. The fastest tool is Compare It with a response time, approximately equal to, 50.4 seconds. It is worth noting that each tool generates a similarity rate for the identified plagiarism. Since we have the statistics of the plagiarized documents in our XML description, we are able to compute the accuracy of each tool. It is true that Compare Suite is the slowest of the three tools. However, it is the most efficient in identifying plagiarism cases with an accuracy of 96.356%.

5. Conclusion

In this paper, we presented our method for the automatic evaluation of existing plagiarism detection tools. In our present work, we focused mainly on French and English documents. With French, we faced an encoding problem, which affected the accuracy of the tools. Indeed, although our French corpus comprises mainly cases of word-for-word (copy/paste) plagiarism, the tools were not able to identify some of the plagiarized parts. Despite the fact that with English, the documents contain cases of obfuscation, the tools performed better while using it. However, we noticed that they are feeble in identifying paraphrases. It is worth noting that, for the majority of research works, the tools are tested by a multitude of users. The final decision is influenced by the reports given by them. However, in our work, given that we have the output of the anti-plagiarism system, we are able to automatically compare it and evaluate its performance. As future works, we aim to consider other languages (different corpora), and more tools, mainly free, the focus of our evaluation will be systems that are, if possible, both online (searches the Web for plagiarism cases) and offline (searches a set of given documents).

[23] https://zenodo.org/record/3250083#.YUrXIrhKjIW
[24] https://www.gutenberg.org/

6. References

[1] A. Nair, A. Nair, G. Nair, P. Prabhu and S. Kulkarni: Semantic Plagiarism Detection System for English Texts, International Research Journal of Engineering and Technology (IRJET), 7,5, e-ISSN: 2395-0056, p-ISSN: 2395-0072, 2020.

[2] A. Patil and N. Bomanwar: Survey on Different Plagiarism Detection Tools and Software's, International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 7 (5), pp. 2191-2193, 2016.

[3] A. Pratomo, A. Irawan and M. Risa: Similarity detection design using Winnowing Algorithm as an effort to apply green computing, Journal of Physics: Conference Series, doi:10.1088/1742-6596/1450/1/012065, 2020.

[4] D. Sakamoto and K. Tsuda: A Detection Method for Plagiarism Reports of Students, in: 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science, 159 (2019), 1329–1338.

[5] I. Ben Salem, P. Rosso and S. Chikhi: On the use of character N-grams as the only intrinsic evidence of plagiarism, Language Resources and Evaluation 53(3), pp. 363–396, 2019.

[6] K. Vani and D. Gupta: Study on Extrinsic Text Plagiarism Detection Techniques and Tools. Journal of Engineering Science and Technology, 9(4), 150–164. doi:10.25103/jestr.094.23, 2016.

[7] M. Elamine, F. Bougares, S. Mechti and L. Belguith: Extrinsic plagiarism detection for French language with word embeddings, in: 19th International Conference on Intelligent Systems Design and Applications (ISDA), 2019.

[8] M. Elamine, S. Mechti and L. Hadrich Belguith: Hybrid plagiarism detection method for French language, International Journal of Hybrid Intelligent Systems, vol. 16, no. 3, pp. 163-175, September 2020.

[9] M. N. Nahas: Survey and Comparison between Plagiarism Detection Tools, American Journal of Data Mining and Knowledge Discovery, vol. 2(2), pp. 50-53, doi: 10.11648/j.ajdmkd.20170202.12, 2017.

[10] Š. Křížková, H. Tomášková and M. Gavalec: Preference comparison for plagiarism detection systems. In O. Cordón (Ed.). Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, Canada. 1760–1767. doi: 10.1109/FUZZ-IEEE.2016.7737903, 2016.

[11] T. Foltynek, D. Dlabolova, J. Mudra, D. Weber-Wulff, A. Anohina-Naumeca, L. Kamzola, S. Kleanthous, S. Razi, J. Kravjar and J. G. Dib: Testing of support tools for plagiarism detection, In proceedings of 5th international conference, Plagiarism across europe and beyond, 2020.

[12] Y. Shkodkina and D. Pacauskas: Comparative Analysis of Plagiarism Detection Systems. Business Ethics and Leadership, 1(3), 27–35. doi: 10.21272/bel.1(3), pp. 27-35, 2017.

[13] Z. Iqbal, S. Murtaza and H. Ayub: Handling Illusive Text In Document To Improve Accuracy Of Plagiarism Detection Algorithm, Preprint DOI: 10.31219/osf.io/hq2j8, 2020.