2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)

Tools for external plagiarism detection in DOCODE

Juan D. Velásquez
Department of Industrial Engineering
University of Chile
República 701, Santiago, Chile
Email: jvelasqu@dii.uchile.cl

Edison Marrese-Taylor
Department of Industrial Engineering
University of Chile
República 701, Santiago, Chile
Email: emarrese@wi.dii.uchile.cl

***Abstract*—In this paper we describe the algorithms and tools offered by DOCODE, a system for plagiarism detection in educational institutions, with a special focus on the task of external plagiarism detection using the Web as a source of information. In that context, although DOCODE is a full-featured system based on several algorithms, our main contribution is an algorithm that given a document is capable of retrieving similar or related documents from the Web, tackling the problem of external plagiarism detection. However, all our algorithms work together to provide high-level plagiarism detection functionalities to our users. Therefore, here we also give details about how these functionalities are bundled and presented in ad-hoc Web-based user interfaces for different kinds of clients, supporting the decision-making process regarding possible plagiarism cases.**

I. Introduction

Today's scenario shows a significant change in the way of accessing information, emphasizing the use of the Internet as one of the main sources of knowledge. However, access to the Web has been cited as one of the main reasons for the perceived decline in academic integrity, particularly in relation to taking others' work and labeling it as one's own, a phenomenon also known as plagiarism [1]. Concretely, because there is a vast amount of easy to access information, the text plagiarism phenomenon, defined as the action of literally copying and pasting someone else's work without the proper citation, has been becoming more popular and easier to resort to.

The act of plagiarism — and particularly text plagiarism — is an important issue for educational purposes at every level, because it could affect a student's learning process [2]. In this context, the term student plagiarism is often used to refer to the incidences of plagiarism committed by students who attend educational institutions [3]. International studies demonstrate the magnitude of this behavior, in both establishments of secondary and higher education, in which a high percentage of students reported using the Web as a major source of plagiarism [4]. In [5], Posner recently estimated that one-third of all high-school and college students have committed some kind of plagiarism. However, the results obtained from several other surveys of plagiarism among students make this estimation seem overly optimistic [3]. The situation in Chile is not different. A 2010 survey carried out by the Department of Industrial Engineering of the University of Chile showed that about 55% of middle school students and 42% of higher education students declared having copied information without citing the source. Moreover, the vast majority of students recognized that the information needed for doing research is obtainable on the Web, mainly through search engines [6].

As a result of this growing phenomenon, there has been a desire on the part of teachers to attack the problem by developing different measures to detect the originality of the work submitted by the students [1]. However, one of the main challenges regarding originality examination and plagiarism detection is that, given the large volume of documents and information sources that exist today, they are becoming increasingly more complex tasks. On the one hand, manual examination appears as an extremely time-consuming process and as a virtually impossible task because teachers often do not have the necessary time for exhaustive reviews. On the other hand, even though Internet search engines, such as Google, can be used to detect Internet plagiarism, the detection process is also, by any standards, both tedious and labor-intensive [3]. As a response to this phenomenon, plagiarism detection engines are becoming more and more important in educational institutions. These are pieces of software that compare documents with possible sources in order to identify similarity and so discover submissions that might be plagiarized [7], making it easier for teachers to analyze a vast number of documents.

In this context, this work is intended to give a description of the Web-based tools that are implemented by our plagiarism detection system, DOCODE 2.0. Our system is a full-featured tool that offers many services regarding plagiarism detection. However, in this paper we will be focusing on the most important and novel approaches in the detection of possible plagiarism cases using the Web as a source of information, an approach also called external plagiarism detection. Our proposals include the combination of several state-of-the-art algorithms that have proven to be very effective for plagiarism detection. These algorithms are embedded in a scalable and robust architecture that allows them to, given a source document, bring suspicious documents from the Web and find the plagiarized sections among them, if any. We bundle these low-level functionalities according to the most common users' needs in educational institutions and offer them to clients through intuitive and user-friendly Web-based interfaces that help our users in the process of dealing with possible plagiarism cases.

The rest of this paper is structured as follows. Below, in Section II we review related work regarding plagiarism and external plagiarism detection. Then, in Section III, we explain how DOCODE is structured and how it works, giving details about what kind of plagiarism detection services it provides based on information available on the Web. Later, Section IV introduces our Web-based user interfaces that allow our users to do requests to the system and explore its results. Finally, Section V presents conclusions and proposed future work.

978-1-4799-4143-8/14 $31.00 © 2014 IEEE
DOI 10.1109/WI-IAT.2014.111

II. Related Work

In this section, we give a short review about plagiarism, including some of the most important definitions stated by the scientific community regarding external plagiarism detection.

A. The problem of plagiarism

Plagiarism or the act of plagiarizing, means to appropriate ideas, passages, etc., from another work or author. However, many other alternative definitions can be found in literature. For example, in [8], the term plagiarism is usually used to refer to the theft of words or ideas, beyond what would normally be regarded as general knowledge.

One of the first attempts to define plagiarism types was proposed in the early 90s, in [9]. According to the author, plagiarism must be considered a serious breach of scholarly ethics, being a theft of credit for ideas in a competitive intellectual marketplace. The paper actually proposes that plagiarism can take six distinct forms, which were also discussed later in [10]. The same paper also says that in education, students may plagiarize to gain a grade, while in academics, the reason may often be to gain popularity and status. However, in both cases, if a plagiarism relationship exists between two texts, it suggests that the texts exhibit some degree of intertextuality, which would not appear between them if independently written. Concretely, in [11], authors state that students may use various techniques for disguising plagiarism in their submitted work, regardless of the type of cheating behavior. In this context, [12] proposes five different plagiarism levels, which were developed further in [3], in 2010.

B. External Plagiarism Detection

A lot of research has been conducted on detecting plagiarism automatically. However, in most of the recent work, plagiarism is simply considered as the reuse of someone else's work while pretending it to be one's own. In this context, [13] declares that literature on the subject often puts plagiarism detection on a level with the identification of highly similar sections in texts or other objects. Thus, [14] gives a formal definition of a plagiarism case $s = \langle s_{plg}, d_{plg}, s_{src}, d_{src} \rangle$ as a 4-tuple which consists of a passage s_{plg} in a document d_{plg} that is the plagiarized version of some source passage s_{src} in d_{src}. When given d_{plg}, the task of a plagiarism detector is to detect s, say, by reporting a plagiarism detection $r = \langle r_{plg}, d_{plg}, r_{src}, d'_{src} \rangle$ which consists of an allegedly plagiarized passage r_{plg} in d_{plg} and its source r_{src} in d'_{src}, and which approximates s as closely as possible.

The same authors also considered that the existing view on plagiarism did not show the whole picture and decided to divide plagiarism detection into two major problem classes, namely external plagiarism detection and intrinsic plagiarism detection. On the one hand, we found the intrinsic approach, in which the plagiarism detector attempts to detect plagiarized passages solely based on information extracted from d_{plg} [15]. On the other hand, in external plagiarism detection, it is assumed that the source document d_{src} for a given plagiarized document d_{plg} can be found in a document collection D, such as the Web.

Regarding external plagiarism detection, the process is typically divided into three steps. In the first place, there is candidate retrieval, which identifies a small set of candidate documents D_{src} that are likely to be sources for plagiarism regarding d_{plg}. The second phase is a detailed comparison, where each candidate document d_{src} in D_{src} is compared to d_{plg}, extracting all passages of text that are highly similar. Finally, the third phase is a knowledge-based post-processing, where the extracted passage pairs are cleaned, filtered, and possibly visualized for later presentation [14].

In order to develop new insights on the topic, annual competitions have been organized under the name of PAN since 2009. For the purpose of these competitions, organizers also developed the first corpora explicitly comprising plagiarized text and a set of performance measures for plagiarism detection in [16]. Later, the same authors presented an evaluation framework for plagiarism detection in [17], which included a revised version of their corpora and the formal definition of metrics to evaluate the performance of an automatic plagiarism detector. Again, in 2012, the same authors decided to construct a new corpus comprising long, manually written documents, for the first time emulating the entire process of plagiarizing [18].

As a result of the PAN competitions and because of the interest of the scientific community in the problem of plagiarism, several techniques to detect different plagiarism cases exist today. Depending on the kind of plagiarism that is being employed, an important set of approaches, based on different characteristics of the text, can be used to proceed with a detection method [7]. Existing literature on each topic is vast, so some authors have already surveyed approaches in automatic plagiarism detection. Here, we will merely refer to the most important studies, including [10], [19] and [12].

III. Proposed System

In this section, we present a description of our system, specially focusing on the functionalities that are based on documents extracted from the Web. We also briefly discuss some details about the architecture that supports all our system.

As we said before, DOCODE is aimed to deliver very different services to a wide number of institutions and individuals. Because of this reason, we offer DOCODE as a Web service and we have already developed clients for some of the most important Virtual Learning Environments (VLE). Because we have chosen the service-oriented paradigm for offering DOCODE, our system is supported by an architecture designed with Java Enterprise Edition or JEE and based on multiple layers. Figure 1 shows the layers that we have defined. Under this paradigm each layer corresponds to the implementation of a different logical functionality for the system. Below, we briefly explain the main role of each layer.

The first layer is the Client Layer, where the users access the application and consume its services. Two types of clients are possible to be used: Web service clients and simple Web-based clients. For the Web service clients, each request is sent to our service with a given URL, using the SOAP protocol over HTTP. Our service receives the request, processes it and returns a response. As of now, DOCODE is only working in the asynchronous mode. Therefore, after each request is received, the requirement is queued until the system is free to process it. After the answer is ready, it is sent or served to the user. The

second layer is related to the second type of client, as it is the Web Layer. This basically corresponds to the implementation of the different Web interfaces to consume the services using browsers.

Next is the Service layer, in charge of making the Web services available to the external net. The service layer represents an interface between clients (people and systems) and is where the business rules are implemented. To implement the service, we chose Java API for XML Web Services (JAX-WS), which is included in the JEE plaftorm we selected, GlassFish Server Open Source Edition. We used XSD (XML Schema Definition) files to define the structures that are sent within the SOAP message and a WSDL (Web Service Description Language) file to declare all the structures defined in the XSD file, as well as the methods that are available for execution.

After, we find the the Business Logic Layer, which saves the actions and processes that contain the business rules to DOCODE's correct operation. The business logic rules captures business requirements, processes data and prepares a display of the results. Once the requirements have been captured and documented, it is necessary to involve the people who are in charge to analyze the requirements and check whether plagiarism has been detected or not.

Then, we have the Persistence and Data Layers. The former is implemented using an object-oriented database, based on Entity Beans, in charge of representing the relational model as a model of objects, using ad-hoc libraries. Finally, the Data layer appears as one of the critical parts of our solution, as it is in charge of managing issues regarding data redundancy, data reuse, access control and frequent backups. In our case, this layer is supported by the file system and a relational database. Both structures work together in order to allow the managing and processing of large sets of documents in parallel. Tables disposed in the database are mostly intended to save information about the registered users who are allowed to access DOCODE's services, about the jobs that have been executed and about the documents that have been processed.

The main functionalities of our system are provided by the application and combination of seven algorithms that tackle different perspectives of the problem of plagiarism. These algorithms comprise four general tasks, which are briefly described below.

1) *Cross-document originality analysis*: Given two or more documents, we want to analyze them and discover if they have passages that may have been plagiarized or if they present any other hint of plagiarism.
2) *Retrieval of similar documents*: Given a source document, the goal is to search for other similar documents from which the author may have plagiarized. For doing so, as proposed by literature, two alternative sources of documents can be used, the Web and internal private databases.
3) *Quotation Detection*: References may have been deliberate or inaccurate been used in a document. In this sense, our intention is to extract quotations to help detecting fake references (those that do not really exist), false references (they exist but do not match text being referenced) and the use of "forgotten" or expired links to sources [3].
4) *Detection of suspicious passages*: Using only one single documents, the goal is to find possible plagiarized passages. With this task, we take the perspective of intrinsic plagiarism detection.

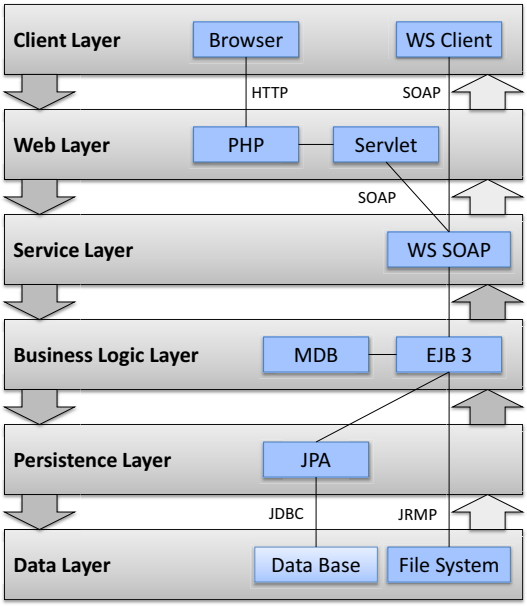

Figure 1. DOCODE system's layers diagram.

In the following sub-sections we will discuss our algorithms for each one of these four basic tasks, with a special focus on the task of gathering similar documents from the Web.

A. Cross-document originality analyzers

The first algorithm inside DOCODE is our cross-document copy-detector, called FAST DOCODE [20]. FAST DOCODE is based on two main phases, which are applied after an initial preprocessing step in each case. In general terms, the algorithm first reduces the search space by using an approximated search of segments of n-grams and then, using an exhaustive search algorithm within selected pairs of documents, finds the offset and its length for both exact and obfuscated copy.

FAST DOCODE was tested during the PAN2010 and PAN2011 competitions, and obtained the 5th and 3rd place respectively. Using the PAN2011 corpus, the algorithm obtained 22.58% for recall and 91.17% for precision. More details of these results can be found in the publications that have been released regarding our algorithm so far, particularly, in [21].

Our second approach focuses on analyzing the originality of a large set of documents using a thematic analyzer, which helps to determine whether a large number of documents have similar overall content. In this case, our approach is based on LSA (Latent Semantic Analysis,) which presents a technique to analyze relationships between a set of documents and the

terms they contain by producing a set of new concepts related to the documents and terms. In order to do this, the algorithm assumes that words that are close in meaning occur in similar pieces of text. Our approach is inspired by the work of [22], where the authors propose a plagiarism detection technique based on LSA. More details can be found in our related paper [23].

B. Suspicious passages detectors

Our second functionality is intended to find suspicious passages in a document using only the same document as a source of plagiarism, by using the intrinsic plagiarism detection strategy. In this case, we propose two main algorithms to detect possible suspicious passages.

In the first place, we implement a change-of-writing-style detector. The intuition behind our proposal is that if some of the words used in the document are author specific, one can think that those words could be concentrated in the segments that the mentioned author wrote [24]. In this manner, the algorithm proposes a model for writing style quantification, aimed at finding significant deviations in a document's writing style; these differing segments could have been plagiarized and are probably useful as a starting point to search for possible source candidates [25]. Our algorithm was tested during the PAN competitions in 2010 and 2011 using the corresponding corpora. Compared to state-of-the-art approaches, the algorithm achieved remarkable results, managing to obtain the first place at the PAN2011 competition, obtaining a precision and recall of 33.98% and 31.23%. More details about these results can be found in our related papers, [24] and [25].

On the other hand, we also propose an approach to detect the use of techniques or tricks that try to exploit various weaknesses of existing plagiarism detection systems [3]. Based on our experience in the Chilean case, we tackle the insertion of invisible white-colored letters as a replacement for spaces, a behavior that seems to be one of the most common strategies employed by students when trying to disguise plagiarism. Our approach in this context is an algorithm that checks if the average word length of each sentence is in a range of what is considered normal or acceptable. We have already calibrated our system setting some threshold parameters using linguistic-based rules and experiments.

C. Similar Document Retrievers

Another basic feature that is provided by DOCODE is a similar Web document retriever, which is in charge of obtaining suspicious documents from the Web. Given an initial suspicious document d and a collection D of documents from which d's author may have plagiarized, the first step (so-called heuristic retrieval step) proposes to retrieve a small number of candidate documents $D_x \in D$, which are likely to be sources for plagiarism. This usually considers that D is very large [26].

Based on the suspicious document our algorithm tackles the problem of obtaining a set of similar documents from the Web using search engines. We therefore see this problem from the perspective of Information Retrieval. Although literature usually proposes that search queries can be grouped into three categories, namely informational, navigational and transactional queries, in [27], the authors introduce a new information requirement — retrieving similar documents from the Web. In [28], the authors call this problem the Web document similarity retrieval problem (WDSRP). The key difference between classic query categories and the WDSRP is that, as input, it considers a given document instead of a text-based query.

As stated by [29], given a document D, from which a vocabulary V can be extracted, a language model MD from D is a function that maps a probability measure over strings drawn from V. Language models are used as ranking functions in information retrieval, estimating the probability of generating a query q given a document language model MD, i.e., $P(q|MD)$. In our query generation task, the probabilistic distribution from the language model is used as a randomized term extraction procedure from D. In this context, our algorithm uses the Hypergeometric Language Model (HLM), an extension of language models inspired by the multivalued hypergeometric distribution [30], thus proposing that when obtaining terms from D to create a query, terms should be extracted one by one without replacement. Here, the premise is that new terms give more information to a search engine than repeated ones in the generated query, considering that search engines allow a maximum length of input queries. The process then starts with the extraction of the vocabulary from D and the assignment of term extraction probabilities, calculated using customizable weighting approaches like tf, tf-idf, among others [31]. Afterwards, we proceed to construct our queries by the concatenation of successive randomized term extractions, without replacing the extracted terms. The length of queries is customizable. In addition to this algorithm, we also proposed another query system aimed to extract a sample of proportionally distributed n-grams ensuring that the terms of a query belong to the same topic [32]. We call this algorithm the random n-gram sample (RNS) fingerprint approach.

Finally, we propose an algorithm to estimate the similarity between document D and each document retrieved from the Web. Our approach combines two main features. The first feature is based on the Zipf-like distribution function over the content of the retrieved documents, modeling the relevance of a given Web search engine answer of a query as a Zipf-like distribution. In this sense, the relevance of the results presented in a Web search engine is inversely related to their rankings. In this manner, we are estimating the relevance of the answer of a query by fusing its ranking and the reliability of search engine results [28]. Our second feature is the result of combining the title and the summary (a.k.a. snippet) of the search engine results into a vector space model, aiming to build an approximated representation of the document's content. Then, we use our two features together to predict similarity assuming that they are strongly related with the similarity between D and the Web document for each result. Our model can be fitted using methods such as Artificial Neural Networks (ANNs) or other related regression techniques [32].

Our strategy has been tested experimentally to measure the effectiveness of the model at satisfying user information needs, which are related to the WDSRP. We first generated a manually-elaborated corpus[1] of 160 paragraphs, selected from

[1]Our corpus is freely available in http://dcc.uchile.cl/~fbravo/docode/corpus.xml

different Web sites in Spanish. Next, the paragraphs were sent to the system as input and the top 15 answers from each paragraph were manually reviewed and classified as relevant or non-relevant results (2,400 Web documents). The criteria to label an answer as relevant was that the retrieved document must contain the given paragraph exactly and the selected evaluation measure was the precision at k, the number of relevant documents retrieved in the top k results divided by the number of documents retrieved in the top k results. Table I shows the obtained performance.

k	1	2	3	4	5
Precision	86.9%	70.9%	60.6%	53.0%	46.9%

Table I. PRECISION FOR RELEVANT DOCUMENTS RETRIEVED IN THE TOP k RESULTS.

As seen, results prove that our proposal is able to satisfy the document similarity retrieval problem. Likewise, they show that the developed meta-search model significantly improved the retrieval capacity over the results of a single search engine [31]. Our algorithm is currently able to connect with the three most important search engines, Google, Yahoo! Boss and Bing.

On the other hand, since our system stores all the documents that have been processed in the past and all the Web documents that have been downloaded by the Web retriever in a historic internal database, we also implement a document retriever for this special database. Our algorithm is based on Apache Lucene[2] and it is basically able to collect a set of similar documents given an initial document as a query. We use Lucene because it provides a ranking model based on the classic vector space model and the Boolean model from Information Retrieval [3]. We take advantage of these optimized models to help the algorithm select the most similar documents regarding the query.

D. Quotation detector

The last algorithm that provides a basic functionality is a quotation detector for the Spanish language. In this case, we take the work of [33] as a basis, where guidelines of different writing styles are presented for writers, editors and publishers. Based on these guidelines, we built several strategies, based on regular expressions an additional context information, to detect the different quotation styles in the most accurate manner.

Since we wanted to test the effectiveness of our approach, we manually elaborated a corpus of bibliographic quotations using 20 undergraduate bachelor theses of agronomy students from the University of Chile. We carefully revised the documents and extracted all the quotations. Altogether, the documents had 15,169 sentences, from which only 536 were tagged as quotations — only 3.53% of the sentences. We then extracted all the quotations using our algorithm and compared the results with our manually elaborated gold standards, seeing the problem as in the case of Information Retrieval. To evaluate if each extracted quotation was in fact inside our annotated set, we combined different string matching algorithms, including Levenshtein Distance, Needleman Wunsch and Smith Waterman to generate new a index that was best suitable for our case. Finally, using this index, we set a similarity threshold of 90% of similarity to declare two quotations as the same.

Results showed that in average we achieved a precision of 24.45% and a recall of 76.91% in the task of extracting quotations from real text. Considering that when searching for quotations recall seems to be more critical, our results seem to be effective and very promising. Also, precision can be improved using additional algorithms to filter our extracted results, for instance, using search engines. We will give more insight about this component in future work.

IV. USER INTERFACE

In this section, we show how DOCODE's user interfaces are designed. As we said in previous sections, we have developed several clients for our service. However, it is possible to divide our interfaces regardless of these clients, but based on the kind of service that we aim to deliver. In general terms, the services provided by the interfaces on each category are the same. The first category includes all the clients that we have already developed, with support for Moodle[4], Sakai[5] and also for U-Cursos[6], an academic platform used by more than 30 academic institutions in Chile. This category also includes DOCODE ASP, a piece of software created by us based on Moodle that is targeted toward those users that are interested in using our services but do not yet possess any of the supported clients. Because of this, DOCODE ASP is offered via the Web and therefore can be accessed using any Web browser[7]. Since we implement all the basic features of Moodle as a VLE, we also offer DOCODE ASP to independent teachers or to institutions that do not currently use any system for supporting their activities. Finally, the second and third categories correspond to two special services that we have also developed, named DOCODE Thesis and DOCODE Lite. We will give more insights about each category in the following subsections.

A. VLE clients and DOCODE ASP

Before going further into introducing our user interface specifications for this category, we first need to define some concepts related to student plagiarism that will help in the explanations. These concepts will be concerned with the fact that most of the educational institutions are currently using VLEs or other learning platforms based on the Web that provide access to classes, homework, grades and so on.

Consider an educational institution that actively uses a VLE. It is possible to recognize certain similarities in the way students are organized and evaluated. In our notation, a written assignment or task given to a student or group of students will be called *homework*. The file containing the answers submitted by one student will be called *submitted document* or simply document. On the other hand, the person (or group of persons) in charge of assigning a particular *homework* will be called *teacher*. Then, we define a *course* as a set of students that have been assigned to the same *homework*. Finally, we define a *corpus* as a collection of submitted documents for

[2]http://lucene.apache.org/

[3]https://lucene.apache.org/core/3_6_2/scoring.html

[4]https://moodle.org/

[5]https://sakaiproject.org/

[6]https://www.u-cursos.cl/

[7]http://asp.docode.cl/

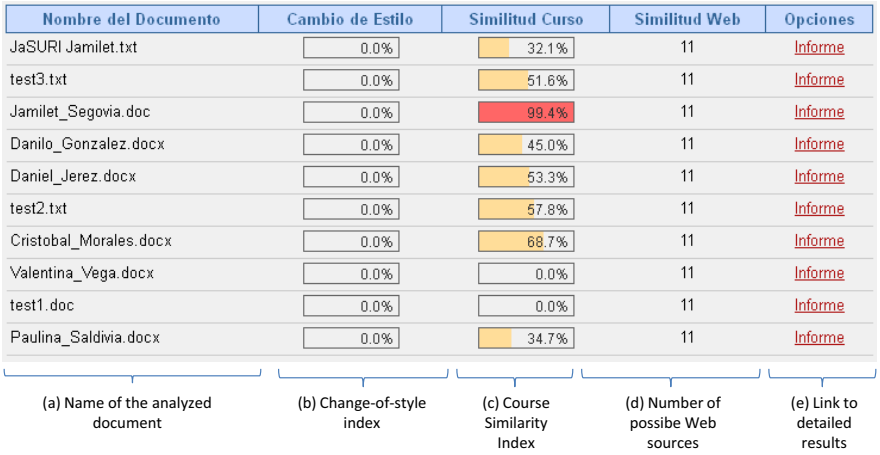

Figure 2. Page showing the general results for one processed *corpus*. The system presents a set of measures regarding each document inside the corpus, comprising different types of analyses. In the picture, column (b) presents a measure that evaluates the relative amount of passages that present writing-style changes, as the percentage of the text in the document. We also show a hidden-text indicator, which only lights up in case we detect an unusual and possibly deceptive behavior in a document. In column (d), we show the number of possible Web sources collected using our similar Web document retriever algorithm. Finally, in column (c) we present a measure of how close each document is to the rest of the documents in the *corpus*, based on our cross-document copy detector algorithm. To build this measure, we compare all the documents inside a corpus each other and show the higher similarity value obtained. In the image, we see a possible case of deceptive behavior in row 3, showing a document that presents a high value of similarity to another document in the *corpus*.

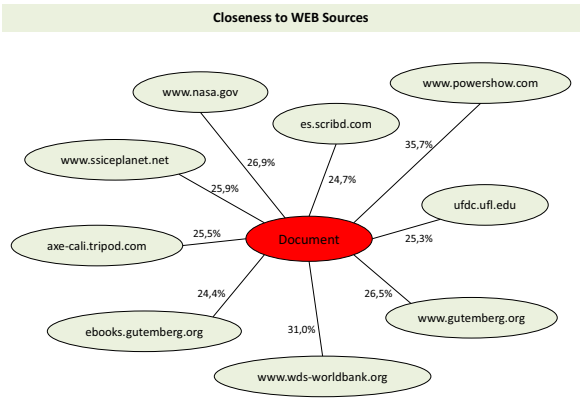

Figure 3. Chart showing the summary of Web sources detected for a single document. For each detected source, the chart also shows the degree of similarity between the analyzed document and the source Web page using percentage. For each case, the measure is based on the total length of the similar passages that are detected in the source, as percentage of the length of the analyzed document.

one *homework*. DOCODE operates on the basis of *courses*, *homework* and the corresponding *corpora*, which may contain one or more documents. Results offered by DOCODE for *corpus* can be grouped in two categories, namely, for single documents inside one *corpus*, or for all the documents inside the *corpus* simultaneously.

Our system offers two main screens to explore the results, each one of which is mainly focused on one of the categories explained before. In this manner, results for each document inside a *corpus* are first presented in a general screen, which is intended to show the big picture of all the submitted documents. By clicking on each row on the screen (see column (e) in Figure 2), users can access the detailed results of that document, which are displayed on the page for single documents. The basic setting for this new page is an interactive on-line version of the document that will be used to show the suspicious passages of the text.

In the case of the general results page, our results include several numerical measures for each document inside a *corpus*, which are disposed in the upper part of the page in a table-like manner, as well as three charts that summarize some of these values and present them in a more intuitive manner. Figure 2 shows an example of this table. On the other hand, our first chart, named *Course Similarity Analysis Chart*, is based on FAST DOCODE and is intended to show the degree of similarity between all the documents inside a corpus. The similarity degree between each pair of documents is based on the length of all passages that are similar as a percentage of the total length of the documents. Our second chart, named *Thematic Analysis Chart*, is based on our multi-document thematic analyzer algorithm and is intended to show the similarity in topics between documents inside a corpus. Our last chart is called *Web Sources Analysis Chart*, and is based on the results of our similar Web document retriever of all the documents inside the *corpus*. Basically, we select the more frequent obtained sources for the *corpus* and generate a chart to visualize them.

On the detailed page for a particular document, we offer lists with different items regarding possible plagiarism cases. In the first place, we offer a list of all the documents in the corpus that may be part of an intra-corporal plagiarism case. For each one of these documents, we present the list of passages that are similar. Below this list, we show another list with all the passages that present style deviations according to our algorithms. Below, we also offer a list with all the

extracted quotations. By clicking on any item on any list, the system highlights the selected passage in the interactive document mentioned above.

Finally, we offer the list of the URLs of each detected Web source for the document. By clicking each URL, the corresponding list of suspicious passages are shown. By clicking again on each URL, the original Web page is opened on a new window, while if any item is clicked, the corresponding passage is highlighted in the interactive document. In addition, we offer a chart showing all the Web sources and the degree of similarity with the document (see Figure 3 for more details).

B. DOCODE Thesis

Another special tool that we offer is DOCODE Thesis, a service specially targeted to analyzing the originality of single long files, including theses and other related documents. In this case, like for DOCODE ASP, our tool is based on Sakai and is offered via the Web, so it can be accessed using any Web browser.

DOCODE Thesis basically offers the same features than DOCODE ASP, but it has a different design and is mainly supported by our similar Web document retriever. Since in this case each *corpus* is intended to be composed of only single documents, the discovery of external sources of plagiarism is a crucial task. Likewise, the quotation analysis is also important because it helps professors and educators check the way students are writing references. Therefore, in this case, our interface is focused in handling and presenting the results of these algorithms to the users in the best possible manner. The design that is currently on development is very similar to DOCODE ASP, including our interactive version of the document. However, for DOCODE Thesis, the results are focused on the topics mentioned above.

C. DOCODE Lite

Last but not least, we also offer DOCODE Lite, a free service where users can analyze a single document by comparing its content with different Web sources. This special service is offered via the Web and can be accessed by any user upon registration. This piece of software is intended mostly for users interested in trying some of the functionalities provided by DOCODE.

The design of DOCODE Lite follows the same ideas that we have already discussed, with the difference that it does not need any VLE client or DOCODE ASP to operate. Basically, after creating one free account, users can upload a file less than 3 MB in size and send it using DOCODE Lite's Web platform. After the file is processed and the results are ready, the user receives an e-mail with the results of the analysis based on Web sources, showing all the detected sources and the corresponding suspicious passages. As Figure 4 shows, for each result we also provide its probability of being a real source of possible plagiarism for the analyzed document, based on the prediction given by our approach, as explained in Section III-C.

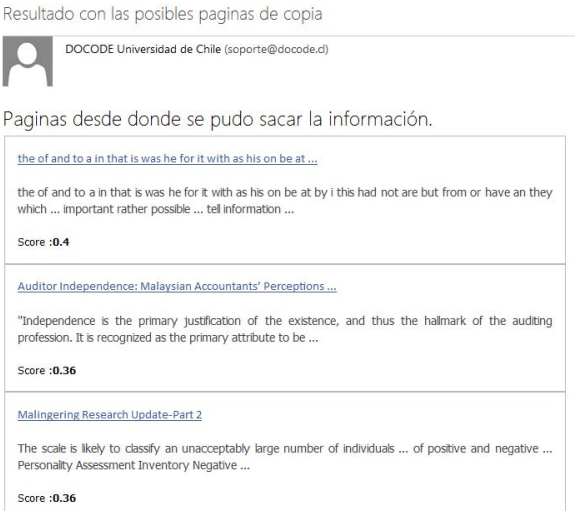

Figure 4. Example of an automatically generated e-mail with the results of DOCODE Lite. Since our similar Web Retriever algorithm is not language dependent, in the picture we see results for the analysis of a document written in English.

V. Conclusions

In this paper, we have described the Web-based algorithms and tools offered by DOCODE, a full-featured plagiarism detection engine that is intended to offer professors and educators a set of tools to gain insights about their students' work, making it easier to analyze a vast number of documents.

We have shown that DOCODE implements state-of-the art algorithms for external plagiarism detection and that it also has several Web-based user interfaces to display the results of the analysis in a simple and intuitive manner for different kinds of clients with different needs. This is possible since we developed our system as a service-oriented platform. Our system's architecture allows us to offer high availability and total transparency for the clients that consume the services.

Despite the fact that DOCODE offers several algorithms to detect multiple kinds of plagiarism cases using different strategies, regarding possible extra-corporal plagiarism [34] we have shown that DOCODE successfully implements state-of-the-art algorithms to perform external plagiarism detection using the Web as a source of information. This feature, primarily based on the combination of our cross-document copy-detector and our similar Web document retriever algorithms, allowed us to design ad-hoc tools to offer our clients. These tools are presented to our clients through intuitive and user-friendly Web-based interfaces, always offering the highest standards in our results, as they are supported by our past publications and awards in the PAN competitions.

Some authors, including [3] propose that an effective automatic plagiarism detector has to be able to identify the plagiarism involved in copying from another student's work and also the plagiarism involved in copying without acknowledgment from reference materials (such as those in textbooks and the Internet). For that matter, we see that DOCODE covers both cases, embracing the problems of intra-corporal plagiarism, using the documents inside a *corpus*, and also extra-corporal plagiarism, based on the results of our similar Web document retriever algorithm.

For future work, we plan to continue working DOCODE Thesis, which is still on development stage. On the other hand,

we are also already working on a set of new visualization tools based on RIA (Rich Internet Application) for DOCODE ASP. However, we first want to know which of the current features the users liked or disliked most so far in order to make our new interface better. At the same time, we intend to evaluate the efficiency of DOCODE in order to prepare our physical systems for larger scales. Finally, we also want to generalize our quotations detector algorithm in order to make it work for different languages.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the continuous support of the Chilean Millennium Institute of Complex Engineering Systems (ICM: P-05-004-F, CONICYT: FBO16); the INNOVA CORFO project (11DL2-10399) entitled, DOCODE: DOcument COpy DEtector (www.docode.cl).

REFERENCES

[1] P. M. Scanlon and D. R. Neumann, "Internet plagiarism among college students," *Journal of College Student Development*, vol. 43, no. 3, pp. 374–385, 2002.

[2] H. Maurer and N. Kulathuramaiyer, "Coping with the copy-paste-syndrome," in *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2007*. AACE, 2007, pp. 1071–1079.

[3] T. Kakkonen and M. Mozgovoy, "Hermetic and web plagiarism detection systems for student essays—an evaluation of the state-of-the-art," *Journal of Educational Computing Research*, vol. 42, no. 2, pp. 135–159, 2010.

[4] D. McCabe, "Cheating: Why students do it and how we can help them stop," in *Guiding Students from Cheating And Plagiarism to Honesty And Integrity: Strategies for Change*. Libraries Unlimited, 2005, pp. 237–246.

[5] R. A. Posner, *The little book of plagiarism*. Random House Digital, Inc., 2009.

[6] F. Molina, J. D. Velásquez, S. Ríos, P. A. Calfucoy, and M. Cocina, "El fenómeno del plagio en documentos digitales: Un análisis de la situación actual en el sistema educacional chileno," *Revista Ingeniería de Sistemas Volumen XXV*, 2011.

[7] P. Clough, "Plagiarism in natural and programming languages: an overview of current tools and technologies," *Research Memoranda: CS-00-05, Department of Computer Science. University of Sheffield, UK*, 2000.

[8] C. Park, "In other (people's) words: Plagiarism by university students–literature and lessons," *Assessment & Evaluation in Higher Education*, vol. 28, no. 5, pp. 471–488, 2003.

[9] B. Martin, "Plagiarism: a misplaced emphasis," *Journal of Information Ethics*, vol. 3, no. 2, pp. 36–47, 1994.

[10] P. Clough and D. O. I. Studies, "Old and new challenges in automatic plagiarism detection," in *National Plagiarism Advisory Service, 2003*, 2003, pp. 391–407.

[11] M. Mozgovoy, T. Kakkonen, and G. Cosma, "Automatic student plagiarism detection: future perspectives," *Journal of Educational Computing Research*, vol. 43, no. 4, pp. 511–531, 2010.

[12] H. A. Maurer, F. Kappe, and B. Zaka, "Plagiarism - a survey." *J. UCS*, vol. 12, no. 8, pp. 1050–1084, 2006.

[13] A. Eiselt, M. Potthast, B. Stein, P. Rosso, and A. Barrón-Cedeño, "Overview of the 1st international competition on plagiarism detection," in *Proceedings of the 3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software misuse and 1st International Competition on Plagiarism Detection*, 2009, pp. 1–9.

[14] M. Potthast, A. Barrón-Cedeño, A. Eiselt, B. Stein, and P. Rosso, "Overview of the 2nd international competition on plagiarism detection," in *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[15] S. M. Zu Eissen and B. Stein, "Intrinsic plagiarism detection," in *Advances in Information Retrieval*. Springer, 2006, pp. 565–569.

[16] A. Barrón-Cedeño, M. Potthast, P. Rosso, B. Stein, and A. Eiselt, "Corpus and evaluation measures for automatic plagiarism detection," in *LREC*, 2010.

[17] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso, "An evaluation framework for plagiarism detection," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 2010, pp. 997–1005.

[18] M. Potthast, T. Gollub, M. Hagen, J. Kiesel, M. Michel, A. Oberländer, M. Tippmann, A. Barrón-Cedeño, P. Gupta, P. Rosso *et al.*, "Overview of the 4th international competition on plagiarism detection." in *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[19] A. Barrón-Cedeño, P. Gupta, and P. Rosso, "Methods for cross-language plagiarism detection," *Knowledge-Based Systems*, vol. 50, pp. 211–217, 2013.

[20] G. Oberreuter, S. A. Ríos, and J. D. Velásquez, "Fastdocode: Finding approximated segments of n-grams for document copy detection - lab report for pan at clef 2010," in *Notebook Papers/LABs/Workshops of the 2010 PAN at CLEF Uncovering Plagiarism, Authorship, and Social Software Misuse*, 2010.

[21] G. Oberreuter, G. L'Huillier, S. A. Ríos, and J. D. Velásquez, "Outlier-based approaches for intrinsic and external plagiarism detection," in *Knowlege-Based and Intelligent Information and Engineering Systems*. Springer, 2011, pp. 11–20.

[22] Z. Ceska, "Plagiarism detection based on singular value decomposition," in *Advances in Natural Language Processing*, ser. Lecture Notes in Computer Science, B. Nordström and A. Ranta, Eds. Springer Berlin Heidelberg, 2008, vol. 5221, pp. 108–119.

[23] F. Bravo-Marquez, G. L'Huillier, P. Moya, S. A. Ríos, and J. D. Velásquez, "An automatic text comprehension classifier based on mental models and latent semantic features," in *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, ser. i-KNOW '11. ACM, 2011, pp. 23:1–23:7.

[24] M. Potthast, A. Eiselt, A. Barrón-Cedeño, B. Stein, and P. Rosso, "Overview of the 3rd international competition on plagiarism detection," in *CLEF (Notebook Papers/Labs/Workshop)*, 2011.

[25] G. Oberreuter and J. D. Velásquez, "Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style," *Expert Systems with Applications*, vol. 40, no. 9, pp. 3756–3763, 2013.

[26] B. Stein, S. M. Zu Eissen, and M. Potthast, "Strategies for retrieving plagiarized documents," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 825–826.

[27] A. R. Pereira and N. Ziviani, "Retrieving similar documents from the web," *Journal of Web Engineering*, vol. 2, no. 4, pp. 247–261, 2003.

[28] F. Bravo-Marquez, G. L'Huillier, S. A. Ríos, and J. D. Velásquez, "Hypergeometric language model and zipf-like scoring function for web document similarity retrieval," in *String Processing and Information Retrieval*, 2010, pp. 303–308.

[29] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 1.

[30] W. L. Harkness, "Properties of the extended hypergeometric distribution," *The Annals of Mathematical Statistics*, vol. 36, no. 3, pp. 938–945, 1965.

[31] F. Bravo-Marquez, G. L'Huillier, S. A. Ríos, J. D. Velásquez, and L. A. Guerrero, "Docode-lite: A meta-search engine for document similarity retrieval," in *Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2010, pp. 93–102.

[32] F. Bravo-Marquez, G. L'Huillier, S. A. Ríos, and J. D. Velásquez, "A text similarity meta-search engine based on document fingerprints and search results records," in *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, 2011, pp. 146–153.

[33] U. of Chicago. Press, *The Chicago manual of style*. University of Chicago Press, 1982.

[34] T. Lancaster and F. Culwin, "Classifications of plagiarism detection engines," *Innovation in Teaching and Learning in Information and Computer Sciences*, vol. 4, no. 2, 2005.