INAOE at DEFT 2011: Using a Plagiarism Detection Method for Pairing Abstracts-Scientific Papers

Fernando Sánchez-Vega[1] Esaú Villatoro-Tello[1] Antonio Juárez-Gozález[1]
Luis Villaseñor-Pineda[1] Manuel Montes-y-Gómez[1,2] Luis Meneses-Lerín[3]
(1) Laboratory of Language Technologies, Department of Computational Sciences,
National Institute of Astrophysics, Optics and Electronics (INAOE), Mexico.
(2) Department of Computer and Information Sciences,
University of Alabama at Birmingham.
(3) LDI (CNRS-UMR) Université Paris 13
{fer.callotl, villatoroe, antjug, villasen, mmontesg}@inaoep.mx

Résumé. Cet article décrit la méthode développée par le Laboratoire de Technologies Langagières de l'INAOE pour la tâche d'appariement résumés/articles dans le cadre de DEFT 2011. Pour aborder cette tâche, on a présupposé qu'un auteur emploi les mêmes expressions contenues dans le corps d'un article pour construire le résumé respectif. En conséquence, notre méthode cherche à retrouver les parties de texte réutilisé dans les résumés et les articles afin de déterminer le degré de dérivation entre eux. Notre méthode suit une stratégie non-supervisée qui ne dépend d'aucune ressources linguistiques, ce qui permet à notre méthode d'être générale et indépendante de la langue. Les résultats obtenus indiquent que le calcul de degré de dérivation entre deux documents peut être utilisé pour ce type de tâches.

Abstract. This paper describes the method developed by the Laboratory of Language Technologies of INAOE for the task of pairing abstracts with their respective scientific articles at the DEFT 2011 edition. Our main hypothesis is that authors commonly employ the same expressions contained in the body of the paper for constructing its respective abstract. Accordingly, our method focuses on the problem of finding portions of reused text between abstracts and papers in order to determine the degree of derivation between them. The proposed method is a non-supervised strategy that does not depend on any external linguistic resource, which allows our method to be general and language independent. Obtained results indicate that using the proposed method for determining the level of derivation between two documents is appropriate for the task of pairing abstracts-papers.

Mots-clés : DEFT 2011, Réutilisation de texte, Document Plagiarism, Indice de réécriture.

Keywords: DEFT 2011, Text reuse, Document Plagiarism, Rewriting Index.

1 Introduction

In order to solve the problem of pairing papers with their respective abstracts, our method assumes that an abstract is in fact a *summary* of the content of some scientific paper. Within such *summary*, normally the main ideas are mentioned reusing the same expressions (or even shorter versions) of those originally exposed in the body of the paper. Therefore, to solve this task we want to find the abstract that better represents the *summary* of some scientific paper. Hence, if we consider that authors will employ very similar (or even the same) expressions to those contained in the original paper for the construction of the abstract, we can model the problem of abstracts-papers pairing as particular case of plagiarism detection.

From a general point of view, document plagiarism detection involves finding similarities between any two documents which are more than just a coincidence and more likely to be result of copying (Clough, 2003). This is a very complex task since reused text is commonly modified with the aim of hide or camouflage the reused text.

The proposed method, which we call the *Rewriting Index*, is able to discover portions of text that have suffered some modifications such as word elimination, different word's order, word insertion and word substitution, allowing to perform a partial matching between any two documents. Our goal was to show that using the *Rewriting Index* algorithm for computing documents' similarity, it is possible to achieve a better performance in the task of abstracts-papers pairing than only considering single words as the general degree of overlap. Is worth mentioning that the proposed method represents a non-supervised strategy (hence, it does not requires of a training phase) and it does not depend on any external linguistic resource, which allows our method to be language independent.

The rest of the paper is organized as follows. Section 2 presents some recent work on the task of plagiarism detection. Section 3 describes the proposed algorithm for finding portions of reused text. Section 4 presents the experimental configuration and results obtained. Finally, Section 5 depicts our conclusions and formulates some directions for future work.

2 Related Work

There are two major approaches for plagiarism detection (Ceska, 2007), which are : *simple* and *structural* approaches. The main variation among these two techniques consists in the strategy used to compute similarities between documents.

2.1 Simple approach

These techniques are called *simple* since they do not consider the structure of the text, and documents are commonly represented by means of their *bag of words* (BOW). Under this type of representation, documents' similarity is computed by some measure that only considers the words contained in both documents, hence, documents with high similarity degree are considered as plagiarized (Barron-Cedeno & Rosso, 2009; Hoad & Zobel, 2003; Zechner *et al.*, 2009). In (Metzler *et al.*, 2005) a similar approach is employed, where the BOW representation is constructed considering only the most frequent words.

Although the BOW strategies are very effective finding relevant documents to some particular text (*e.g.*, finding documents containing some text extracted from the suspicious document), they are not very effective finding documents where the reused text have suffered some modifications (*e.g.* when words are changed by others with similar meaning) which is a common practice of plagiarists. In addition, these strategies are affected by the thematic correspondence of the documents, which implies the existence of common domain-specific words, causing an overestimation of their overlap.

2.2 Structured approach

These type of approaches consider as key element for measuring similarities between documents the natural structure of language, such as the lexical similarity, the word's order and/or the word's *part-of-speech*. Structured techniques for detecting plagiarized documents can be divided into those that are dependent on some external resource and those that are totally independent.

2.2.1 Depending on external resources.

In (Si *et al.*, 1997) the entire structure of the document is considered for evaluation. One of the major drawbacks of this approach is that documents must be in a specific input format (*e.g.*, LaTeX) which includes some labels that facilitate the identification of certain sections of the document. Some other approaches that fit into this category are those that apply automatic translation tools to determine when two documents have high probabilities of been plagiarized (Chien-Ying *et al.*, 2010). Finally, in (Rehurek, 2008) syntactic trees are generated for each documents' sentence, and these trees are employed to determine how many sentences between two documents have the same structure or to identify those words that correspond to the same *part-of-speech*.

The main drawback of these approaches is the high dependency on external resources such as WordNet, or the existence of large and well balanced data sets for training the syntactic analysis and/or the automatic translation process, which drives to a language dependent approach.

2.2.2 Independent from external resources.

Methods employed within this category represent the language structure by means of the order and the adjacency of the words contained in the considered text. For this type of approaches there is no particular interest in knowing the words' *part-of-speech*, instead of that, these techniques try to capture the context of the words of interest. By doing this, it is possible to measure the quantity of reused text portions and also it is possible to know if it occurs in a similar context.

A common factor within these techniques is the text fragmentation, which consists on the generation of *chunks, shingles* or *n-grams* (Barron-Cedeno & Rosso, 2009; Basile *et al.*, 2009; Clough *et al.*, 2002). These parts of the text had some granularity which can be of different sizes (Hoad & Zobel, 2003), from a couple of characters, a few words, or even the entire document. The intuitive idea is to represent documents by using these text *chunks*. The main problem with these strategies is in deciding the size of the *chunk*, since small *chunks* can lead to a high number of repetitions even when there is no actual plagiarism, or if *chunks* are too large it will not be possible to identify those reused text that have suffered minor modifications, such as a different word order.

Our work differs from previous efforts in that our proposed approach, called the *Rewriting Index* (ReI), is able to discover text that have suffered some modifications such as word elimination, different word order, word insertion and word substitution, allowing to compute a partial matching between documents.

3 Plagiarism Detection Method

Generally, as stated above, common word sequences between the suspicious and source documents are considered the primary evidence of plagiarism. Nevertheless, using their presence as unique indicator of plagiarism is too risky, since thematic coincidences also tend to produce sequences of common text (*i.e.*, false positives). In addition, even a minor modification to hide the plagiarism will avoid the identification of the corresponding sequences, generating false negatives.

In order to handle these problems we propose a novel strategy for detecting plagiarised text, called the *Rewriting Index* method, which is able to identify portions of reused text even if the reused text have suffered some modifications.

In the following section we describe our algorithm for identifying and extracting the common text between the suspicious (D^S) and the source document (D^R). From here, the suspicious documents will be the *abstracts* and the source documents will be the *scientific papers*.

3.1 Identifying the reused text

Our proposed approach extracts strings (possible portions of reused text) considering a wide flexibility threshold and it is called the *Rewriting Index* method. This method allows to assign a weight value to each word contained in the suspicious document considering its degree of membership to a possible portion of plagiarized text. Hence, we are able to identify portions of text, that even if they do not represent an exact match, are in fact plagiarized strings. In other words, we are able to obtain non-consecutive portions of reused text ; therefore we are able to capture the common actions of plagiarist such as : word elimination, different word's order, word insertion and word interchange (*e.g.,* by synonyms).

In particular, the *Rewriting Index* method is an *ad-hoc* search algorithm that uses a vicinity (*i.e.,* context window) V that contains v words from the source document D^R (*i.e.*, V moves through the text of document D^R). The position of this vicinity window is defined by its central element which is called the *focus* element and we will refer to it as V^f. Accordingly, V^f will contain the word w_j^R, *i.e.,* the word w at the position j contained in the document D^R.

Additionally, the elements (words) at the right side from V^f will be defined as the V^+ elements, and in similar form, the elements at the left side from V^f will called as the V^- elements. Notice that the V^+ and V^- elements are within the context window. Finally, since our method considers the entire document D^R when searching for possible reused strings, we define as D_-^R to all the elements that appear at the left side of V, and as D_+^R to all the elements that appear at the right side of V.

According to these definitions, the *Rewriting Index* algorithm will assign a ReI value to each word w_i^S (*i.e.,* the word at position i within document D^S) depending on its position within D^R. The pseudocode for computing the ReI values for each word w_i^S is described in the Algorithm 1.

As it is possible to observe in the Algorithm 1, the ReI takes different values (c_i) depending on the position of the searched word w_i^S.That is, if the searched word appears at the V^f position the ReI is equal to 1 indicating a literal copying case ; if the word appears at the right of the*focus* it takes values c_2 or c_4 suggesting a moderate or a great number of deletion/insertion operations respectively ;if the word appears at le left of the*focus* it takes values c_3 or c_5 signifying a moderate or severe change on the word's order respectively ; finally, if the searched word does not appears in D^R, its ReI value is equal to 0. In general, the constants c_i fulfil the following condition :

$$1 > c_2 > c_3 > c_4 > c_5 > 0 \tag{1}$$

The *Rewriting Index* algorithm is able to provide the ReI value of each w_i^S (*i.e.,* to evaluate the complete document D^S) in a time proportional to $O(n)$ in the best case, considering n is the number of words contained in D^S, which means that the suspicious document represents an exact copy of D^R. In the other hand, the worst case will be when every w_i^S does not exists in D^R, which means there is no plagiarism, which leads to a time proportional to $O(nm)$ considering that m represents the number of words contained in D^R.

3.2 Computing the Rewriting Index measure

Previously we explained how to compute the ReI value for each word contained in D^S. However, the *Rewriting Index* measure f^{ReI} refers to a single value that represents how much the words from D^S are taken from D^R (*i.e.,* how much the words from the abstract are taken from the paper in evaluation). The definition of this similarity measure f^{ReI} is as follows :

$$f^{ReI} = \sum_{w_i^S \in D^S} \frac{ReI(w_i^S)}{\mid D^S \mid} \tag{2}$$

Notice that for computing the f^{ReI} measure we considered all the ReI values of every w_i^S.

4 Experiments and Results

4.1 DEFT 2011 Task 2 description

For the DEFT 2011 edition two main tasks were proposed, however we only participate in the Task 2, which main goal is to pair a scientific paper with an abstract. Two different modalities were proposed for this particular task :

1. **TRACK 1** - Pairing abstracts with full papers : For this track, participants were provided with abstracts and a complete version of several papers, *i.e.*, papers that contain all their original sections, such as introduction, related work, evaluation, discussion, conclusions, etc.).
2. **TRACK 2** - Pairing abstracts with incomplete papers : Contrary to the previous track, for this exercise papers do not contain the introduction and conclusion sections.

4.2 Corpus

The provided corpus is composed of scientific papers mainly published in journals from the humanities field (Anthropologie et Sociétés, Études internationales, Études littéraires, Meta, Revue des sciences de l'éducation), all of them in French.

The training corpus is composed of 300 abstract and 300 papers from 5 different reviews in the humanities (one abstract per file and one article per file). Notice that for TRACK 2 papers were reduced by eliminating the introduction and conclusions sections. And for the test corpus, 200 abstract and 200 papers were given, adding papers from a 6th review that was not part of the training set.

```
input  : word $w_i^S$, source document $D^R$ and the vicinity window $V$
output: the rewriting index score of $w_i$
//Enters if the searched word $w_i^S$ is equal to the one in $V^f$, and as a result moves $V^f$ one position and the $ReI$ gets the higher value
if ($w_i^S = V^f$) then
    $V^f \leftarrow V^f + 1$;
    $ReI(w_i^S) \leftarrow 1$;
end
//Enters if searched word $w_i^S$ appears at the right of $V^f$, and as a results moves $V^f$ to the next position where $w_i^S$ exists in $D^R$ and the $ReI$ value is assigned to constant $c_2$
else if ($w_i^S$ appears in $V^+$) then
    $V^f \leftarrow position_{D^R}(w_i^S) + 1$;
    $ReI(w_i^S) \leftarrow c_2$;
end
//Enters if searched word $w_i^S$ appears at the left of $V^f$, and as a result $V^f$ remains at the same position and $ReI$ is assigned to constant $c_3$
else if ($w_i^S$ appears in $V^-$) then
    $V^f \leftarrow V^f$;
    $ReI(w_i^S) \leftarrow c_3$;
end
//Enters if searched word $w_i^S$ appears at the right side of $V$, as a result $ReI$ is assigned to the constant $c_3$
else if ($w_i^S$ appears in $D_+^R$) then
    //Enters if there is more than one single word coincidence in the $D_+^R$ region, and as a result the $V^f$ element is moved to position where the coincidence was found, and the $V^-$ elements are updated for those that previously were in $V^+$
    if ($w_{i+1}^S = w_{position_{D^R}(w_i^S)+1}^R$) then
        $temp \leftarrow V^+$;
        $V^f \leftarrow position_{D^R}(w_i^S) + 1$;
        $V^- \leftarrow temp$;
    end
    $ReI(w_i^S) \leftarrow c_4$;
end
//Enters if searched word $w_i^S$ appears at the left side of $V$, as a result $V^f$ remains the same and $ReI$ is assigned to the constant $c_4$
else if ($w_i^S$ appears in $D_-^R$) then
    $V^f \leftarrow V^f$;
    $ReI(w_i^S) \leftarrow c_5$;
end
//Enters if the searched word $w_i^S$ does not exists in $D^R$, as a result $ReI$ is assigned to 0
else
    $ReI(w_i^S) \leftarrow 0$;
end
Exit;
```

Algorithm 1: The *Rewriting Index* evaluation algorithm

4.3 Proposed method configuration

The strategy followed for finding abstracts-papers pairs was : for each pair abstract-paper we computed the f^{ReI} measure (See expression 2). Afterwards, we performed a ranking process considering the f^{ReI} measure, resulting in an ordered list. Hence, the three best evaluated papers for each abstract are presented as the owners of the respective abstract.

Notice that in the proposed algorithm (Algorithm 1) there are a couple of parameters that have not been defined : the size of the window V and the values of the constants c_i. For the later one, these were defined as : $c_i = 1/i$. Notice that such definition do not violate the constraint defined in expression 1, resulting in the following values :

$$1 > \frac{1}{2} > \frac{1}{3} > \frac{1}{4} > \frac{1}{5} > 0 \tag{3}$$

And with respect to the size of V, we probed with 2 different sizes (5, and 11). Therefore, our experiments labelled as **RUN-1** refer to a context window V of size 5, meanwhile experiments labelled as **RUN-2** refer to a context window V of size 11.

4.4 Baseline method configuration

As our baseline method we employed the well known tool ROUGE (Chin-Yew, 2004) which is mainly oriented to the automatic evaluation of summaries. ROUGE is a tool that is able to measure word's co-occurrences between two documents[1] indicating somehow the degree of overlap between evaluated documents.

ROUGE is able to measure co-occurrences of single words (*i.e.,*$1 - grams$) up to $4 - grams$ (ROUGE-N), the intuition is that the greater the length of n, the better the estimation of the fluency between evaluated documents. Additionally, the ROUGE tool also proposes measuring the co-occurrences of *Longest Common Subsequences* (ROUGE-L), where the intuition is that the longer the LCS of two documents is, the more similar the two documents are.

For our experiments we assign a ROUGE score (R) to each pair of abstract-paper which is computed as follows :

$$R(a,p) = \frac{ROUGE - 1(a,p) + ROUGE - 2(a,p) + ROUGE - L(a - p)}{3} \tag{4}$$

where a refers to an abstract and p to some paper. Hence, once we have computed the R value for every pair of abstract-paper we kept the three papers that obtain a higher value of R for generating the proposed solution, *i.e.,* the three best evaluated papers for each abstract are presented as those with higher probabilities of being the owners of the respective abstract. Experiments performed using the baseline method are labelled as **RUN-3**.

4.5 Results

Since our proposed approach does not requires any training information, we only present obtained results over the test corpus. Table 1 shows the results from the proposed approach for both tracks. As it is possible to observe, the proposed method using a context window V of size 5 (RUN-1) allows to obtain a better accuracy performance for both tracks.

Notice that RUN-1 also outperforms the baseline method (RUN-3) which is a configuration based on word and common sequences co-occurrences. As expected, results from TRACK-2 are lower than those obtained in TRACK-1, which is a clear indicator of how common is the use of expressions contained in both the *Introduction* and *Conclusions* sections for the construction of the abstract.

[1] Ideally between summary pairs.

Track	Run ID	Accuracy
TRACK 1	RUN-1	0.970
	RUN-2	0.960
	RUN-3	0.949
TRACK 2	RUN-1	0.904
	RUN-2	0.848
	RUN-3	0.858

TAB. 1 – Results of the proposed approach

5 Conclusions

In this paper we propose to solve the problem of abstracts-papers pairing by means of a method design for detecting document plagiarism. Our method focuses on the detection of common (possible reused) strings between the source and suspicious documents.

The proposed algorithm, called the *Rewriting Index* method allows to assign a value to each word contained in the suspicious document considering its degree of membership to a possible portion of plagiarized text. By employing the *Rewriting Index* method we are able to identify portions of text, that even if they do not represent an exact match are in fact reused strings, therefore we are able to capture the common actions of plagiarist such as : word elimination, different word's order, word insertion and word interchange.

Experimental results on the DEFT corpus are encouraging ; they indicate that the proposed method for identifying portions of reused text, and measuring similarities between documents are appropriate for the abstracts-papers pairing task. Results also demonstrate that using word or even common sequences co-occurrences is insufficient for this task, conducting to several false positives cases.

Acknowledgments.
This work was done under partial support of CONACyT (Project grants 106013, 134186 and scholarship 224483 - 258345).

Références

BARRON-CEDENO A. & ROSSO P. (2009). On automatic plagiarism detection based on *n*-grams comparison. In *Proceedings of the SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 2009)*, Donostia-San Sebastian, Spain.

BASILE C., BENEDETTO D., CAGLIOTI E., CRISTADORO G. & ESPOSTI M. D. (2009). A plagiarism detection procedure in three steps : Selection, matches and "squares". In *Proceedings of the SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 2009)*, Donostia-San Sebastian, Spain.

CESKA Z. (2007). The feature of copy detection techniques. In *Proceedings of the 1st Young Researches Conference on Applied Sciences (YRCAS 2007)*, p. 5–10, Pilsen, Czech Republic.

CHIEN-YING C., JEN-YUAN Y. & HAO-REN K. (2010). Plagiarism detection using rouge and wordnet. *Journal of Computing.*, **2**(3), 34–44.

CHIN-YEW L. (2004). Rouge : a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, Barcelona, Spain.

CLOUGH P. (2003). Old a new challenges in automatic plagiarism detection. *National Plagiarism Advisory Service*, p. 391–407.

CLOUGH P., GAIZAUSKAS R., PIAO S. & WILKS Y. (2002). Meter : Measuring text reuse. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, USA.

HOAD T. C. & ZOBEL J. (2003). Methods for identifying versioned and plagiarized documents. *Journal of the American Society for Information Science and Technology*, **54**(3), 203–215.

METZLER D., BERNSTEIN Y., CROFT W., MOFFAT A. & ZOBEL J. (2005). Similarity measures for tracking information flow. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM'05)*, p. 517–524.

REHUREK R. (2008). Plagiarism detection through vector space models applied to a digital library. In *Proceedings of Recent Advances in Slavonic Natural Language Processing.*, p. 75–83.

SI A., LEONG H. V. & LAU R. W. H. (1997). Check : A document plagiarism detection system. In *Proceedings of the 1997 ACM Symposium an Applied Computing.*, p. 70–77, San Jose CA, USA.

ZECHNER M., MUHR M., KERN R. & GRANITZER M. (2009). External and intrinsic plagiarism detection using vector space models. In *Proceedings of the SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 2009)*, Donostia-San Sebastian, Spain.