Reducing the Plagiarism Detection Search Space on the Basis of the Kullback-Leibler Distance

Alberto Barrón-Cedeño, Paolo Rosso, and José-Miguel Benedí

Department of Information Systems and Computation,
Universidad Politécnica de Valencia,
Valencia 46022, Camino de Vera s/n, Spain
{lbarron, prosso, jbenedi}@dsic.upv.es
http://www.dsic.upv.es/grupos/nle/

Abstract. Automatic plagiarism detection considering a reference corpus compares a suspicious text to a set of original documents in order to relate the plagiarised fragments to their potential source. Publications on this task often assume that the search space (the set of reference documents) is a narrow set where any search strategy will produce a good output in a short time. However, this is not always true. Reference corpora are often composed of a big set of original documents where a simple exhaustive search strategy becomes practically impossible.
Before carrying out an exhaustive search, it is necessary to reduce the search space, represented by the documents in the reference corpus, as much as possible. Our experiments with the METER corpus show that a previous search space reduction stage, based on the Kullback-Leibler symmetric distance, reduces the search process time dramatically. Additionally, it improves the Precision and Recall obtained by a search strategy based on the exhaustive comparison of word n-grams.

1 Introduction

The easy access to a wide range of information via electronic resources such as the Web, has favoured the increase of plagiarism cases. When talking about text, plagiarism means to use text written by other people (even adapting it by rewording, insertion or deletion), without any credit or citation. In fact, the reuse of self-written text is often considered as self-plagiarism.

Plagiarism detection with reference tries to find the source of the potentially plagiarised fragments from a suspicious document in a set of reference documents. Some techniques based on the exhaustive comparison of suspicious and original documents have already been developed. These techniques apply comparison of sentences [7], structure of documents [15] and entire documents [10]. Examples of the used comparison strategies are dot plot [15] and n-grams [10].

One of the main difficulties in this task is the great size of the search space, i.e., the reference documents. To our knowledge, this problem has not been studied deeply enough, neither there are published papers on this issue. Given a suspicious document, our current research is precisely oriented to the reduction

A. Gelbukh (Ed.): CICLing 2009, LNCS 5449, pp. 523-534, 2009.

of the search space. The proposed approach is based on the Kullback-Leibler distance, which has been previously applied to many applications, ranging from image retrieval [5] to document clustering [13]. The reduction of search space for plagiarism detection is a more specific case of clustering: instead of grouping a set of related documents, the task is to define a reduced set of reference documents containing texts with a high probability of being the source of the potentially plagiarised fragments in a suspicious document. The final objective is to relate potentially plagiarised sentences to their source. Our experiments show that a reduction of the search space based on the Kullback-Leibler distance improves processing time as well as the quality of the final results.

The rest of the paper is structured as follows. Section 2 gives an overview of plagiarism detection including some state-of-the-art approaches. Section 3 defines the proposed method for reducing the search space as well as it describes the exhaustive search strategy we have opted for. Section 4 gives a description of the corpus we have used for our experiments. Section 5 describes the experiments and the obtained results. Finally, Section 6 gives some conclusions and draws future work.

2 Plagiarism Detection Overview

In automatic plagiarism detection, a correct selection of text features in order to discriminate plagiarised from non-plagiarised documents is a key aspect. Clough [3] has delimited a set of features which can be used in order to find plagiarism cases such as changes in the vocabulary, amount of similarity among texts or frequency of words. This kind of features has produced different approaches to this task.

Intrinsic plagiarism analysis [11] is a different task from plagiarism detection with reference. It captures the style across a suspicious document in order to find fragments that are plagiarism candidates. This approach saves the cost of the search process, but it does not give any hint about the possible source of the potentially plagiarised text fragments.

In those cases where a reference corpus is considered, the search process has been based on different features. *Ferret* [10] considers text comparison based on word n-grams. The reference, as well as the suspicious text, is split into trigrams, composing two sets which are compared. The amount of common trigrams is considered in order to detect potential plagiarism cases. *PPChecker* [7] considers the sentence as the comparison unit in order to compare local similarity. It differentiates among exact copy of sentences, word insertion, word removal and rewording on the basis of a Wordnet-based word expansion process.

A major difficulty in this task is the dimension of the reference corpus D. Even assuming that D contains the source document of the plagiarised fragments in a suspicious text s, the search strategy must be efficient enough to accurately find it in a reasonable time. An exhaustive comparison of sentences, paragraphs or any other text chunk s_i in order to answer the question: *is there a chunk* $s_i \in s$ *included in a document of* D*?* could be impossible if D is very large. The complexity of the comparison process is $O(n \cdot m)$, where n and m are the lengths of s and D in fragments. Some efforts have already spend to improve the search speed, such as fingerprinting [16]. In this case a numerical value (fingerprint), which becomes the comparison unit, is assigned to each text chunk of the reference as well as the suspicious text. However, in this case each suspicious document is still compared to the entire reference corpus. In [15] a structural comparison of documents in order to reduce the search space is performed. Unfortunately, this method requires reference and suspicious documents written in LaTeX.

3 Method Definition

Given a reference corpus of original documents D and a plagiarism suspicious document s, our efforts are oriented to efficiently localise the subset of documents D' such that $|D'| \ll |D|$. The subset D' is supposed to contain those documents d with the highest probabilities of including the source of the plagiarised text fragments in s. After obtaining this subset, an exhaustive search of the suspicious sentences in s over D' can be performed. Our search space reduction method, the main contribution of this work, is based on the Kullback-Leibler symmetric distance.

3.1 The Kullback-Leibler Distance

The proposed search space reduction process is based on the Kullback-Leibler distance, which has shown good results in text clustering [2, 13]. In 1951 Kullback and Leibler proposed the after known as *Kullback-Leibler divergence* (KL_d) [8], also known as cross-entropy. Given an event space, KL_d is defined as in Eq. 1. Over a feature vector $\mathcal{X}$, KL_d measures the difference (or equality) of two probability distributions P and Q.

$$KL_d(P \mid\mid Q) = \sum_{x \in \mathcal{X}} P(x) log \frac{P(x)}{Q(x)} \quad . \tag{1}$$

KL_d is not a symmetric measure, i.e., $KL_d(P \mid\mid Q) \neq KL_d(Q \mid\mid P)$. Due to this fact, Kullback and Leibler (and also some other authors) have proposed symmetric versions of KL_d, known as Kullback-Leibler symmetric distance (KL_δ). Among the different versions of this measure, we can include:

$$KL_\delta(P \mid\mid Q) = KL_d(P \mid\mid Q) + KL_d(Q \mid\mid P) \ , \tag{2}$$

$$KL_\delta(P \mid\mid Q) = \sum_{x \in \mathcal{X}} (P(x) - Q(x)) log \frac{P(x)}{Q(x)} \ , \tag{3}$$

$$KL_\delta(P \mid\mid Q) = \frac{1}{2} \left[KL_d \left(P \mid\mid \frac{P+Q}{2} \right) + KL_d \left(Q \mid\mid \frac{P+Q}{2} \right) \right] \ , \tag{4}$$

$$KL_\delta(P \mid\mid Q) = \max(KL_d(P \mid\mid Q), KL_d(Q \mid\mid P)) \ . \tag{5}$$

The equations correspond respectively to the versions of Kullback and Leibler [8], Bigi [2], Jensen [6] and Bennet [1]. A comparison of these versions showed that there is no significant difference in the obtained results [13]. We use Eq. 3 due to the fact that it only implies an adaptation of Eq. 1 with an additional subtraction. The other three options perform a double calculation of KL_d, which is computationally more expensive.

Given a reference corpus D and a suspicious document s we calculate the KL_δ of the probability distribution P_d with respect to Q_s (one distance for each document $d \in D$), in order to define a reduced set of reference documents D'. These probability distributions are composed of a set of features characterising d and s (Subsections 3.2 and 3.3). An exhaustive search process (Subsection 3.4) can then be applied on the reduced set D' instead of the entire corpus D.

3.2 Features Selection

A feature selection must be made in order to define the probability distributions P_d. We have considered the following alternative techniques:

1. *tf* (term frequency). The relevance of the i-th term t_i in the j-th document d_j is proportional to the frequency of t_i in d_j. It is defined as:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \ , \tag{6}$$

   where $n_{i,j}$ is the frequency of the term t_i in d_j and is normalised by the frequency of all the terms t_k in d_j.

2. *tfidf* (term frequency-inverse document frequency). The weight *tf* of a term t_i is limited by its frequency in the entire corpus. It is calculated as:

$$tfidf_{i,j} = tf_{i,j} \cdot idf_i = tf_{i,j} \cdot log \frac{|D|}{|\{d_j \mid t_i \in d_j\}|} \ , \tag{7}$$

   where $|D|$ is the number of documents in the reference corpus and $|\{d_j \mid t_i \in d_j\}|$ is the number of documents in D containing t_i.

3. *tp* (transition point). The transition point tp^* is obtained by the next equation:

$$tp^* = \frac{\sqrt{8 \cdot I_1 + 1} - 1}{2} \ . \tag{8}$$

   I_1 is the number of terms t_k appearing once in d_j [12]. In order to give more relevance to those terms around tp^*, the final term weights are calculated as:

$$tp_{i,j} = (\langle tp^* - f(t_i, d_j) \rangle + 1)^{-1} \ , \tag{9}$$

   where, in order to guarantee positive values, $\langle \cdot \rangle$ is the absolute value function.

The aim of the feature selection process is to create a ranked list of terms. Each probability distribution P_d is composed of the top terms in the obtained list, which are supposed to better characterise the document d. We have experimented with $[10, \ldots, 90]\%$ of the terms with the highest $\{$*tf,tfidf,tp*$\}$ value in d (Section 5).

3.3 Term Weighting

The probability (weight) of each term included in P_d is simply calculated by Eq. 6, i.e., $P(t_i, d) = tf_{i,d}$. These probability distributions are independent of any other reference or suspicious document and must be calculated only once.

Given a suspicious document s, a preliminary probability distribution Q'_s is obtained by the same weighting schema, i.e., $Q'(t_i, s) = tf_{i,s}$. However, when comparing s to each $d \in D$, in order to determine if d is a source candidate of the potentially plagiarised sections in s, Q'_s must be adapted.

The reason is that the vocabulary in both documents will be different in most cases. Calculating the KL_δ of this kind of distributions could result in an infinite distance ($KL_\delta(P_d \mid\mid Q'_s) = \infty$), when a t_i exists such that $t_i \in d$ and $t_i \notin s$. The probability distribution Q_s does depend on each P_d. In order to avoid infinite distances, Q_s and P_d must be composed of the same terms. If $t_i \in P_d \cap Q'_s$, $Q(t_i, s)$ is smoothed from $Q'(t_i, s)$; if $t_i \in P_d \setminus Q'_s$, $Q(t_i, s) = \epsilon$. This is simply a back-off smoothing of Q. In agreement with [2], the probability $Q(t_i, s)$ will be:

$$Q(t_i, s) = \begin{cases} \gamma \cdot Q'(t_i \mid s) & \text{if } t_i \text{ occurs in } P_d \cap Q'_s \\ \epsilon & \text{if } t_i \text{ occurs in } P_d \setminus Q'_s \end{cases} . \tag{10}$$

Note that terms occurring in s but not in d are not relevant. γ is a normalisation coefficient estimated by:

$$\gamma = 1 - \sum_{t_i \in d, t_i \notin s} \epsilon \ , \tag{11}$$

respecting the condition:

$$\sum_{t_i \in s} \gamma \cdot Q'(t_i, s) + \sum_{t_i \in d, t_i \notin s} \epsilon = 1 \ . \tag{12}$$

ϵ is smaller than the minimum probability of a term in a document d. After calculating $KL_\delta(P_d \mid\mid Q_s)$ for all $d \in D$, it is possible to define a subset of source documents D' of the potentially plagiarised fragments in s. We define D' as the ten reference documents d with the lowest KL_δ with respect to s.

3.4 Exhaustive Search Process

Once the reference subcorpus D' has been obtained, the aim is to answer the question "*Is a sentence $s_i \in s$ plagiarised from a document $d \in D'$?*". Plagiarised text fragments use to appear mixed and modified. Moreover, a plagiarised sentence could be a combination of various source sentences. Due to these facts, comparing entire documents (and even entire sentences) could not give satisfactory results.

In order to have a flexible search strategy, we codify suspicious sentences and reference documents as word n-grams (reference documents are not split into sentences). It has been shown previously that two independent texts have a low level of matching n-grams (considering $n > 1$). Additionally, codifying texts in this way does not decrease the representation level of the documents. In the particular case of [10], this has been shown for trigrams. In order to determine if the i-th sentence s_i is plagiarised from d, we compare the corresponding sets of n-grams. Due to the difference in the size of these sets, we carry out an asymmetric comparison on the basis of the *containment* measure [9], a value in the range of $[0,1]$:

$$C(s_i \mid d) = \frac{|N(s_i) \cap N(d)|}{|N(s_i)|} \ , \tag{13}$$

where $N(\cdot)$ is the set of n-grams in $\cdot$.

Once every document d has been considered, s_i becomes a candidate of being plagiarised from d if the maximum $C(s_i \mid d)$ is greater than a given threshold.

4 The Corpus

In our experiments, we have used the *XML* version of the *METER corpus* [4]. This corpus was created as part of the METER (MEasuring TExt Reuse) Project.[1] The METER corpus is composed of a set of news reported by the Press Association (PA). These news were distributed to nine British newspapers (The Times, The Guardian, Independent and The Telegraph, among others), which can use them as a source for their own publications.

For experimental purposes, we have considered 771 PA notes as the original documents, which is the entire set of PA notes in this corpus version. The corpus of suspicious documents is composed of 444 newspaper notes. We selected this subset due to the fact that the fragments in their sentences are identified as *verbatim*, *rewrite* or *new*. These labels mean that the fragment is copied, rewritten or completely independent from the PA note, respectively.

Verbatim and rewritten fragments are triggers of a plagiarised sentence s_i. s_i is considered plagiarised if it fulfils the inequality $|s_{i_v} \cup s_{i_r}| > 0.4|s_i|$, where $|\cdot|$ is the length of $\cdot$ in words whereas s_{i_v} and s_{i_r} are the words in verbatim and rewritten fragments, respectively. This condition avoids erroneously to consider sentences with named entities and other common chunks as plagiarised. Some statistics about the reference and suspicious corpora are included in Table 1. The pre-processing for both reference and suspicious documents consists of word-punctuation splitting ($w, \rightarrow [w][,]$) and a Porter stemming process [14].[2]

5 Experiments

As we have pointed out, the aim of the proposed method is to reduce the search space before carrying out an exhaustive search of suspicious sentences across the reference documents.

[1] http://www.dcs.shef.ac.uk/nlp/meter/

[2] We have used the Vivake Gupta implementation of the Porter stemmer, which is available at http://tartarus.org/~martin/PorterStemmer/

Table 1. Statistics of the corpus used in our experiments

Feature	**Value**
Reference corpus size (kb)	1,311
Number of PA notes	771
Tokens / Types	226k / 25k
Suspicious corpus size (kb)	828
Number of newspapers notes	444
Tokens / Types	139k / 19k
Entire corpus tokens	366k
Entire corpus types	33k

Once P_d is obtained for every document $d \in D$, the entire search process is the one described in Fig. 1.

```
Algorithm 1: Given the reference corpus $D$ and a suspicious document $s$:

// Distance calculations
Calculate $Q'_s(t_k) = tf_{k,s}$ for all $t_k \in s$
For each document $d$ in the reference corpus $D$
    Define the probability distribution $Q_s$ given $P_d$
    Calculate $KL_\delta(P_d \mid\mid Q_s)$
// Definition of the reduced reference corpus
$D' = \{d\}$ such that $KL_\delta(P_d \mid\mid Q_s)$ is one of the 10 lowest distance measures
$n_{s_i} = [n\text{-grams in } s_i]$ for all $s_i \in s$
// Exhaustive search
For each document $d$ in the reduced reference corpus $D'$
    $n_d = [n\text{-grams in } d]$
    For each sentence $s_i$ in $s$
        Calculate $C(n_{s_i} \mid n_d)$
    If $\max_{d \in D'}(C(n_{s_i} \mid n_d)) \geq Threshold$
        $s_i$ is a candidate of being plagiarised from $\arg\max_{d \in D'}(C(n_{s_i} \mid n_d))$
```

Fig. 1. Plagiarism detection search process.

We have carried out three experiments in order to compare the speed (in terms of seconds), and quality of the results (in terms of Precision, Recall and F-measure), of the plagiarism detection process with and without search space reduction. The experiments explore four main parameters:

1. Length of the terms composing the probability distributions: $l = \{1, 2, 3\}$
2. Feature selection technique: tf, $tfidf$ and tp
3. Percentage of terms in d considered in order to define P_d: $[10, \cdots, 90]\%$
4. Length of the n-grams for the exhaustive search process: $n = \{1, 2, \ldots, 5\}$

In the first and second experiments, we carried out a 5-fold cross validation. The objective of our first experiment was to define the best values for the first three parameters of the search space reduction process. Given a suspicious document (newspaper) s we consider that D' has been correctly retrieved if it includes the source document (PA) of s. Figure 2 contains the percentage of sets correctly retrieved in the experiments carried out over the different development sets. In the five cases the results were practically the same.

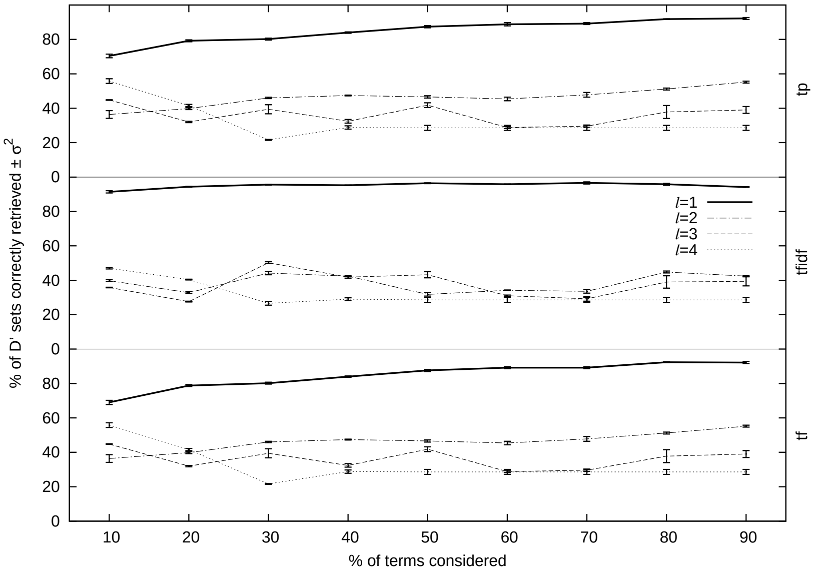

Fig. 2. Evaluation of the search space reduction process. Percentage of sets correctly retrieved ($\{tf, tfidf, tp\}$ = feature extraction techniques, l = term length)

The best results for the three feature selection techniques are obtained when unigrams are used. Higher n-gram levels produce probability distributions where a good part of the terms has a weight near to 1. These distributions (where almost all the terms have the same probability) do not allow KL_δ to determine how close are two documents. Regarding the feature selection techniques, considering tf does not give good results. In this case a good number of functional words (prepositions and articles, for example), which are unable to characterise a document, are considered in the corresponding probability distributions. The results obtained by considering tp are close to those of tf. Considering mid-terms (which tries to discard functional words), seems not to characterise either this kind of documents because they are too noisy. The results with this technique could be better with longer documents. The best results in this case are obtained with $tfidf$. Functional and other kinds of words that do not characterise the document are eliminated from the considered terms and the probability distributions characterise correctly the reference (and after the suspicious) document.

Regarding the length of the probability distributions, the quality of the retrieval is practically constant when considering $tfidf$ with unigrams. The only real improvement is achieved when considering 20% of the document vocabulary; the percentage of correctly retrieved documents increases from 91% to 94% when 10% and 20% of the vocabulary is considered. The best option is to consider the 20% of the vocabulary in order to compose the probability distributions of the reference documents. In this way we obtain a good percentage of correctly retrieved reference documents with a sufficiently low dimension for the probability distributions. When applying the best parameters over the corresponding test sets, the obtained results did not show significant variations.

The second experiment aims to explore the fourth parameter (on the exhaustive search process). The containment threshold was varied in order to decide whether a suspicious sentence was plagiarised or not. Precision, Recall and F-measure were estimated by considering the five development sets of suspicious documents. Figure 3 shows the results obtained with n in the range $[1, 5]$ over the entire reference corpus D.

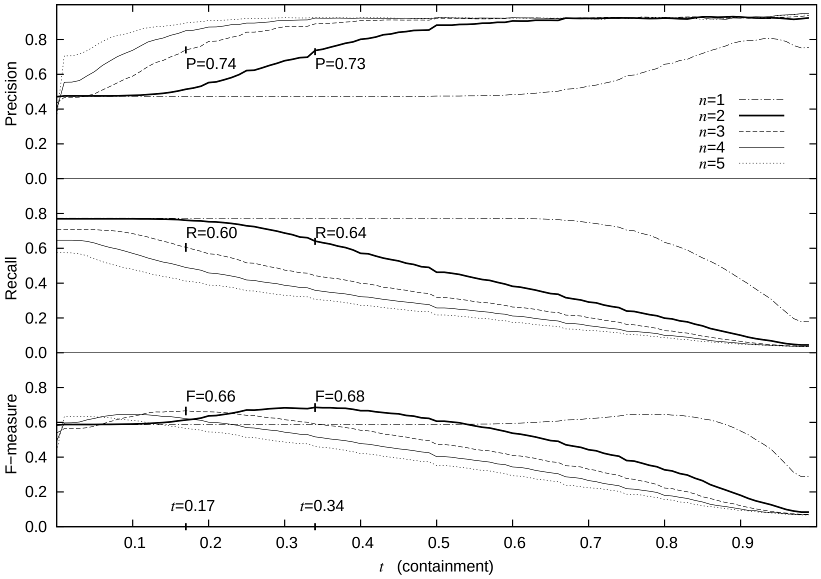

Fig. 3. Exhaustive search process evaluation. (n = n-gram level, t =threshold)

The text codification, based on a simple bag of words ($n = 1$), does not consider context information and style. This results in a good Recall (practically constant until $threshold = 0.7$). However, the probability of a reference document of containing the entire vocabulary of a suspicious sentence s_i is too high. Due to this reason, Precision is the lowest among all the experiments. On the other side, considering n-grams of level 4 (and higher) produces a rigid search strategy. Minor changes in the plagiarised sentences avoid their detection, resulting in the lowest Recall values.

The best results are obtained when considering bigrams and trigrams (best F-measures are 0.68 and 0.66, respectively). In both cases, the word n-grams are short enough to handle modifications in the plagiarised fragments as well as long enough to compose strings with a low probability of appearing in any (but the plagiarism source) text. Trigram-based search is more rigid, resulting in better Precision. Bigram-based search is more flexible, allowing a better Recall. The difference is reflected in the threshold in which the best F-measure values are obtained for both cases: 0.34 for bigrams versus 0.17 for trigrams. The threshold with the best F-measure t^* was after applied to the corresponding test set. The obtained results were exactly the same ones obtained during the estimation, confirming that $t^* = \{0.34, 0.17\}$ is a good threshold value for bigrams and trigrams, respectively.

The third experiment shows the improvement obtained by carrying out the reduction process in terms of speed and quality of the output. Table 2 shows the results obtained by bigrams when s is searched over D as well as D', i.e., the original and the reduced reference corpora. In the first case, we calculate the containment of $s_i \in s$ over the documents of the entire reference corpus D. Although this technique by itself obtains good results, considering too many reference documents that are unrelated to the suspicious one, produces noise in the output, affecting Precision and Recall. An important improvement is obtained when $s_i \in s$ is searched over D', after the search space reduction process.

With respect to the processing time, the average time needed by the method to analyse a suspicious document s over the entire reference corpus D is about 2.32 seconds, whereas the entire process of search space reduction and the analysis of the document s over the reduced subset D' needs only 0.19 seconds[3]. This big time difference is due to three main factors: (1) P_d is pre-calculated for every reference document, (2) $Q'(s)$ is calculated once and simply adapted to define each Q_s given P_d, and (3) instead of searching the sentences of s in D, they are searched in D', which only contains 10 documents.

With respect to the output quality, Precision and Recall become higher when the search space reduction is carried out. Moreover, this result is obtained considering a lower containment threshold. The reason for this behaviour is simple: when we compare s to the entire corpus, each s_i is compared to many documents that are not even related to the topic of s, but contain common n-grams. Note that deleting those n-grams composed of "common words" is not a solution due to the fact that they contain relevant information about the writing style. The reduction of the threshold level is due to the same reason: less noisy compar-

[3] Our implementation in Python has been executed on a Linux PC with 3.8GB of RAM and a 1600 MHz processor.

isons are made and plagiarised sentences that were not considered before are now taken into account.

Table 2. Results comparison: exhaustive search versus space reduction + exhaustive search. (P =Precision, R =Recall, F = F-measure, t = avg. processing time (secs.))

Experiment	**threshold**	**P**	**R**	**F**	**t**
Without space reduction	0.34	0.73	0.63	0.68	2.32
With space reduction	**0.25**	**0.77**	**0.74**	**0.75**	**0.19**

6 Conclusions

In this paper we have investigated the impact of the application of a search space reduction process as the first stage of plagiarism detection with reference. We have additionally explored different n-grams levels for the exhaustive search subprocess, which is based on the search of suspicious sentences codified as n-grams over entire documents of the reference corpus. The obtained results have shown that bigrams as well as trigrams are the best comparison units. Bigrams are good to enhance Recall whereas trigrams are better to enhance Precision, obtaining an F-measure of 0.68 and 0.66 over the entire reference corpus, respectively.

The search space reduction method is the main contribution of this work. It is based on the Kullback-Leibler symmetric distance, which measures how closed two probability distributions are. The probability distributions contain a set of terms from the reference and suspicious documents. In order to compose them, term frequency, term frequency-inverse document frequency and transition point (tf, $tfidf$ and *tp*, respectively) have been used as feature selection techniques. The best results were obtained when the probability distributions were composed of word unigrams selected by $tfidf$.

In the experiments a comparison of the obtained results was made (also in terms of time performance) by carrying out the exhaustive search of n-grams over the entire as well as the reduced reference corpora. When the search space reduction was applied, the entire reference corpus (700 documents approximately) was reduced to only 10 reference documents. In this optimised condition, the plagiarism detection process needs on average only 0.19 seconds instead of 2.32. Moreover, the F-measure was improved (from 0.68 to 0.75 when using bigrams).

As future work we would like to consider a different measure from the Kullback-Leibler distance for the search space reduction process. Moreover, it would be interesting to carry out an exhaustive search process based on the fingerprinting technique (after the reduction process). Additionally, we would like to validate the obtained results in a bigger corpus composed of larger documents. Unfortunately we do not have knowledge about the existence of a corpus matching the required characteristics and creating one is by itself a hard task.

Acknowledgements. We would like to thank Paul Clough for providing us the METER corpus. This work was partially funded by the MCyT TIN2006-15265-C06-04 research project and the CONACyT-MEXICO 192021/302009 grant.

References

1. Bennett, C.H., Gács, P., Li, M., Vitányi, P.M., Zurek, W.H.: Information Distance. IEEE Transactions on Information Theory 44(4) 1407–1423 (1998)
2. Bigi, B.: Using Kullback-Leibler distance for text categorization. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 305–319. Springer, Heidelberg (2003)
3. Clough. P.: 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)
4. Clough, P., Gaizauskas, R., Piao, S.: Building and Annotating a Corpus for the Study of Journalistic Text Reuse. In: 3rd International Conference on Language Resources and Evaluation (LREC 2002), Las Palmas, Spain, vol. V, pp. 1678–1691 (2002)
5. Do, M.N., Vetterli, M.: Texture Similarity Measurement Using Kullback-Leibler Distance on Wavelet Subbands. In: International Conference on Image Processing, vol. 3, pp. 730–433 (2000)
6. Fuglede, B., Topse, F.: Jensen-Shannon Divergence and Hilbert Space Embedding. In: IEEE International Symposium on Information Theory (2004)
7. Kang, N., Gelbukh, A., Han, S.-Y.: PPChecker: Plagiarism pattern checker in document copy detection. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 661–667. Springer, Heidelberg (2006)
8. Kullback, S., Leibler, R.A.: On Information and Sufficiency. Annals of Mathematical Statistics 22(1), 79–86 (1951)
9. Lyon, C., Malcolm, J., Dickerson, B.: Detecting Short Passages of Similar Text in Large Document Collections. In: Conference on Empirical Methods in Natural Language Processing, Pennsylvania, pp. 118–125 (2001)
10. Lyon, C., Barrett, R., Malcolm, J.: A Theoretical Basis to the Automated Detection of Copying Between Texts, and its Practical Implementation in the Ferret Plagiarism and Collusion Detector. In: Plagiarism: Prevention, Practice and Policies Conference, Newcastle, UK (2004)
11. Meyer zu Eissen, S., Stein, B.: Intrinsic plagiarism detection. In: Lalmas, M., MacFarlane, A., Rüger, S.M., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds.) ECIR 2006. LNCS, vol. 3936, pp. 565–569. Springer, Heidelberg (2006)
12. Pinto, D., Jiménez-Salazar, H., Rosso, P.: Clustering abstracts of scientific texts using the transition point technique. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 536–546. Springer, Heidelberg (2006)
13. Pinto, D., Benedí, J.-M., Rosso, P.: Clustering narrow-domain short texts by using the Kullback-Leibler distance. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp 611–622. Springer, Heidelberg (2007)
14. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
15. Si, A., Leong, H.V., Lau, R.W.H.: CHECK: A Document Plagiarism Detection System. In: ACM Symposium on Applied Computing, CA, pp. 70-77 (1997)
16. Stein, B.: Principles of Hash-Based Text Retrieval. In: Clarke, Fuhr, Kando, Kraaij, and de Vries (eds.) 30th Annual International ACM SIGIR Conference, pp. 527–534 (2007)