

Word Length n-Grams for Text Re-use Detection

Alberto Barrón-Cedeño¹, Chiara Basile²,
Mirko Degli Esposti², and Paolo Rosso¹

¹ NLEL-ELiRF, Department of Information Systems and Computation,
Universidad Politécnica de Valencia, Spain

{lbarron,prossso}@dsic.upv.es

<http://www.dsic.upv.es/grupos/nle/>

² Dipartimento di Matematica,

Università di Bologna, Italy

{basile,desposti}@dm.unibo.it

Abstract. The automatic detection of shared content in written documents –which includes text reuse and its unacknowledged commitment, plagiarism– has become an important problem in Information Retrieval. This task requires exhaustive comparison of texts in order to determine how similar they are. However, such comparison is impossible in those cases where the amount of documents is too high. Therefore, we have designed a model for the proper pre-selection of closely related documents in order to perform the exhaustive comparison afterwards. We use a similarity measure based on word-level n-grams, which proved to be quite effective in many applications. As this approach becomes normally impracticable for real-world large datasets, we propose a method based on a preliminary word-length encoding of texts, substituting a word by its length, providing three important advantages: (i) being the alphabet of the documents reduced to nine symbols, the space needed to store *n*-gram lists is reduced; (ii) computation times are decreased; and (iii) length *n*-grams can be represented in a trie, allowing a more flexible and fast comparison. We experimentally show, on the basis of the perplexity measure, that the noise introduced by the length encoding does not decrease importantly the expressiveness of the text. The method is then tested on two large datasets of co-derivatives and simulated plagiarism.

Keywords: word length encoding; text similarity analysis; text reuse analysis; plagiarism detection; information retrieval.

1 Introduction

Similarity between documents is a key factor in diverse Natural Language Processing and Information Retrieval (IR) tasks such as documents clustering and categorization [5]. Problems that require a deeper analysis of similarity between texts are text-reuse analysis [7], co-derivatives analysis [4], information flow tracking [14], and plagiarism detection [13]. In these tasks, we are not only interested in looking up how many keywords a pair of documents have in

common, but in how related their contents are. While this could be considered as a semantic problem, different methods based on chunks comparison, a purely syntactic approach, have shown competitive results [13].

The exhaustive comparison of entire documents is a hard task; comparing strings is computationally complex and defining the best chunks to be compared is not straightforward. On the one hand, comparison techniques have been designed on the basis of fingerprint models, such as Wining [16]. Fingerprinting is often based on the sub-sampling of text chunks and an information loss must be assumed when opting for these methods. On the other hand, when a comparison of the entire content of the document is required, character and word-level n -grams have shown to be a good option [6].

Detection of text reuse, co-derivatives, and plagiarism can be divided into three steps (cf. [17]): (*i*) heuristic retrieval of potential source documents—given a document, retrieving a proper amount of its potential source documents—; (*ii*) exhaustive comparison of texts—comparing the texts in order to identify those fragments which could be re-used and their potential sources—; and (*iii*) knowledge-based post-processing (only for plagiarism detection)—proper citations are eliminated from the plagiarism candidate fragments—. Nevertheless, research on these tasks often approaches step (*ii*) only, assuming that the rest are solved [12,10,6]. However, this is not true. Note that step (*i*) is a more specific case of clustering and IR: instead of grouping/retrieving a set of related documents, the task is to define a reduced set of potential source documents containing texts with a high probability of being the source of the text fragments in the analysed document.

Hereinafter we propose a method to approach step (*i*). We make it by estimating how close two documents are on the basis of the so named *length encoding*. The method encodes every word in the implied texts by its length in characters and splits the resulting text into n -grams. The comparison between documents can be then performed on the basis of standard measures such as the cosine distance or the Jaccard coefficient [8]. The method is tested on two corpora of simulated plagiarism and text co-derivatives showing promising results.

The remainder of the paper is laid out as follows. Section 2 gives a description of the two corpora we have used in our experiments. Section 3 describes the length encoding method, including an empirical analysis of its validity based on language models and perplexity. Section 4 includes the experiments we have carried on in order to compare how well the model works with respect to a “traditional” word-level n -gram comparison model. Finally, Section 5 draws conclusions and outlines future work.

2 Corpora

In order to perform our experiments we used two datasets: the PAN-PC-09 corpus and the Wikipedia co-derivatives corpus.¹

¹ Both corpora are available at <http://www.dsic.upv.es/grupos/nle/downloads.html>

2.1 PAN-PC-09 Corpus

The PAN-PC-09 [15] corpus was created in the framework of the 1st International Competition on Plagiarism Detection². This freely available resource for the evaluation of plagiarism detection methods is divided into development and test sections. The former can be used in order to tune up and test models as it includes annotations on the plagiarism cases as well as their sources. This is the section we used for our experiments. It contains 7214 source documents and 7214 suspicious documents. Further descriptions on this corpus are available in [2].

2.2 Co-derivatives Corpus

This corpus was generated for the analysis of co-derivatives, text reuse and (simulated) plagiarism. It is composed of more than 20,000 documents from Wikipedia in four different languages: English, German, Spanish and Hindi. It contains around 5,000 documents for each language, including some of the most frequently accessed articles in Wikipedia. For each article ten revisions were downloaded, composing the set of co-derivatives. The corpus pre-processing includes whitespace normalization, sentence detection, tokenization and case folding. An extensive description of the corpus construction can be found in [2].

3 Method Definition

3.1 Notation

The notation used throughout the rest of the paper is the following. Let D be the set of all reference documents and $\{d_q\}_{q \in Q}$ the set of query documents; these will be either the texts which are suspected of containing plagiarism (PAN-PC-09 corpus) or the most recent revision of each Wikipedia article (co-derivatives corpus). The query documents can be contained in D or not, depending on the experiment. For a document $d \in D$, let $V_n(d)$ be the set of all n -grams in d (the *n-gram vocabulary*). Let $D_q \subseteq D$ be the set of the first k neighbours of d_q according to some similarity measure. Let $L_q \subseteq D$ be the set of source documents of the re-used text in d_q : L_q contains, in the first case, all the sources for the plagiarism in d_q , as described in the development section of the PAN-PC-09 corpus, and in the second case it is composed of the 10 revisions of the Wikipedia article, the last of which is precisely d_q .

The goal of the method is to maximize the intersection between L_q and D_q , without increasing too much the number of retrieved texts k .

3.2 Length Encoding

The length encoding model was formerly introduced in [3], where it was used to reduce the search space for the PAN-PC-09 competition dataset. It takes the

² <http://www.webis.de/pan-09/competition.php>

idea of word-level n -grams comparison but, instead of comparing word strings, it compares length strings, that in fact become integer numbers. Let w be a word in a given text and let $|w|$ be its length in characters. The steps of the length encoding, including a brief example to illustrate, are the following:

Input:	This UFO related place is the so-called 'area 51'.
Pre-processing: substitute any non-letter symbol with a blank space.	This UFO related place is the so called area
Encoding: replace each word w with $\min(w , 9)$	4 3 7 5 2 3 2 6 4

After length encoding the document, n -grams can be obtained to characterise it. For instance, by considering $n = 5$, the resulting n -grams are: {43752, 37523, 75232, 52326, 23264}. Such n -grams can be handled as integers instead of strings, causing a saving of memory space (integers, indeed, occupy less space than strings, as discussed afterwards) and accelerating the comparison process. Note that if the “traditional” word n -gram schema is followed, the 5-grams for the example sentence above are: {this ufo related place is, ufo related place is the, related place is the so, place is the so called, is the so called area}. Clearly, adopting the usual approach also requires a lower-casing process, which is unnecessary with our length encoding.

Nevertheless, information is still redundant in the length n -gram list. In order to reduce redundancy, it is possible to profit from the limited vocabulary these n -grams are composed of. As the vocabulary is $\alpha = \{1, 2, 3, \dots, 9\}$, indeed, it is straightforward to compose a *trie* (also known as prefix tree) to represent the entire document. This is very difficult when considering the actual document vocabulary. Figure 1 contains the trie characterization of the sample text given before. For instance, in the example the 1-gram 4 (corresponding to both **this** and **area**) appears twice in the text, while the 4-gram 4375, corresponding to **This UFO related place** (third branch counting from the left and going down to the fourth generation from the root), appears once. The advantages of the proposed method are the following:

1. Computational time is significantly reduced.
2. The space occupied by the encoded documents is reduced with respect to the one needed to encode word n -grams or the list of length n -grams.
3. All the n -grams for $n \in \{1, \dots, N\}$, N being the depth of the trie, are available in the trie itself.

Regarding points 1 and 2, consider that, as beforementioned, instead of strings (be of characters or numbers), integers can be used. Integers can be handled on 32 or 64 bits, whereas strings are composed of chains of 16-bit characters (an average word of 4 characters occupies 64 bits). Additionally, comparing integers is much faster than comparing strings. It is true that other techniques, such

With respect to point 3, one single data structure includes the n -grams of all levels (up to a given threshold) in the document. As a result, the comparison can be carried on by considering any value of n without further processing. This makes the comparison strategy much more flexible, which is not possible, for instance, when considering fingerprinting models as Winnowing or SPEX).

The length encoding model certainly adds noise to the texts, since at low levels of n a lot of different text strings are translated into the same code. However, when increasing n , the noise decreases, becoming at last irrelevant. In order to show that, we exploit the concept of perplexity, an entropic measure which estimates the uncertainty of a language model (cf. [9]). Table 2 shows the values of perplexity, divided by the cardinality of the corresponding n -gram dictionary, for both word and length n -grams and with $n = 1, \dots, 6$. A convergence of the perplexity of the length n -gram model to that of the word n -gram model is evident, even if it was not possible to calculate the value for larger levels of n , because perplexity is a sentence-level measure, and sentence-end effects could affect the calculation for larger n .

Table 2. Perplexity for the different level language models, divided by the cardinality of the corresponding n -gram dictionary. *wng* = word n -grams; *lng* = length n -grams

encoding	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
<i>wng</i>	9.2×10^{-3}	1.4×10^{-4}	3.9×10^{-5}	2.7×10^{-5}	2.4×10^{-5}	2.3×10^{-5}
<i>lng</i>	9.6×10^{-1}	1.0×10^{-1}	1.1×10^{-2}	1.2×10^{-3}	1.4×10^{-4}	1.6×10^{-5}

In this context, it is also interesting to observe the distribution of n -gram frequencies in a large dataset. In Fig. 2 the data are reported for the distributions in a set composed of 500 documents extracted from the PAN-PC-09 corpus, with both length and word n -grams. Note that, as long as n grows, the two distributions tend to coincide, and a large superimposition is reached already for $n = 12$. This observation supports empirically the intuitive idea that, for a large enough n , the length encoding is “almost injective”, i.e., very few word n -grams are mapped to the same length n -gram.

3.3 Similarity Estimation

The similarity measure we opted for is the Jaccard coefficient [8], a very standard indicator based only on a comparison between the n -gram vocabularies of the two texts into consideration, totally disregarding n -gram statistics. This is a good measure in such cases where the value of n is large enough to make very unlikely that an n -gram repeats more than a few times in a text: when considering word n -grams, this happens already with $n = 3, 4$ (see for example the case of $n = 5$ in Fig. 2), which are certainly appropriate values in this case where we are considering text re-use cases. The Jaccard coefficient between texts d_q and $d \in D$ is defined as follows ($|\cdot|$ indicating here the cardinality of a set):

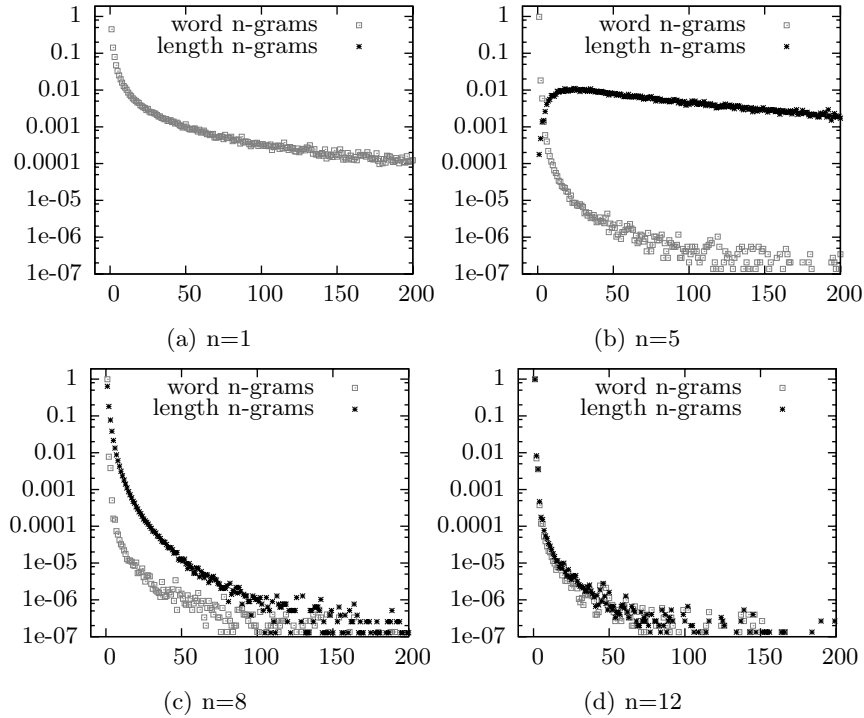


Fig. 2. Frequency distributions for length n -grams (*black stars*) and for word n -grams (*gray squares*), for some values of n . The number of occurrences lies on the x -axis, with the corresponding percentage of n -grams on the y -axis. The length n -gram distribution converges to the one of word n -grams as n grows. No stars appear in the first plot because we show up to 200 occurrences only, which is lower than the frequency of any possible 1-gram of length encoded text in a representative corpus.

$$J_n(d_q, d) := \frac{|V_n(d_q) \cap V_n(d)|}{|V_n(d_q) \cup V_n(d)|}. \tag{1}$$

J_n takes values in the interval $[0, 1]$. It is closer to 1 as long as the superimposition between the vocabularies of d_q and of d is larger.

4 Experiments

We performed experiments in order to compare a common word n -gram encoding to the proposed length n -gram encoding. Evaluation in terms of Recall was carried out by considering two corpora including cases of simulated plagiarism and text co-derivatives (cf. Sections 2.1 and 2.2, respectively). Results are shown in Sections 4.1 and 4.2. The methods were compared in terms of time-performance also, with results shown in Section 4.3.

4.1 Experiments on the PAN-PC-09 Corpus

Experiment Outline. The first experiment on this corpus (**exp1** hereinafter) has the aim of verifying the appropriateness of the length encoding for the recognition of relevant documents for plagiarism cases. In order to identify the appropriate value of n for such task, we first used a repeated sampling technique: for every run, we selected a small random subset $\{d_q\}_{q \in \tilde{Q}}$ of query documents and an appropriate subset \tilde{D} of reference documents, and evaluated the performance. The value of n that performed the best was then used to apply the length encoding method to the whole PAN-PC-09 development corpus.

We have already justified the use of word length encoding in Section 3 however, we also wanted to compare our method with the one based on “traditional” word-level n -grams. Therefore, we performed a second experiment (**exp2** hereinafter) where we selected a subset of the corpus (the same used in [3], composed of 160 query texts and 300 reference documents), and calculated the Jaccard coefficient for both length and word n -grams, with $n = 2, 4, \dots, 20$.

For both experiments we fixed k , the number of retrieved documents, to the value of 10, in agreement with the co-derivatives experiment (Section 4.2).

Measures of Performance. There are various possible definitions of the *recall* for this problem; the choice of the right one depends on what we want to measure precisely. First of all, we have to choose between a single query text average or a global average. To avoid problems of divisions by zero for those query documents not containing plagiarism, we decided to use a global measure, following the approach of [15].

Another choice is whether we want to measure the fraction of recalled *source texts* from which the plagiarism comes or that of the *plagiarised characters* contained in the selected source texts. In the PAN-PC-09 corpus every query document has an associated XML file with detailed annotation about the copied sections, with character-level precision. In order to take advantage from this annotation, we used for both **exp1** and **exp2** the following character-level measure:

$$R_c@k := \frac{\sum_q \sum_{s \in \Delta_q} |s|}{\sum_q \sum_{s \in \Lambda_q} |s|}, \quad (2)$$

where Λ_q is the set of all plagiarised sections in d_q , $\Delta_q \subseteq \Lambda_q$ is the set of plagiarised passages in d_q that come from its first k neighbours according to the n -gram distance into consideration (i.e., from the selected texts in L_q), and $|s|$ expresses here the length of passage s , measured in characters. This measure gives a larger weight to longer copied passages, in the same spirit of the measure used for the Competition on Plagiarism Detection [15].

Since half of the query documents in the PAN-PC-09 are entirely original, containing no plagiarised passages, we did not calculate any *precision* measure.

Results. The results of **exp1** are given in Fig. 3(a). The recall $R_c@10$ is greater than 0.8 for all values of n larger than 8, and it reaches its maximum for $n = 12$.

An important observation is that identifying the relevant texts in such small samples of the corpus is much simpler, from a purely statistical viewpoint, than the “real” task of detecting few relevant texts for each suspicious document in the whole dataset of 7,214 sources. At this point, thus, having identified 12 as a proper value for n , we calculated the Jaccard coefficient J_{12} on the whole PAN-PC-09 development corpus and obtained a recall $R_c@10 = 0.86$, a value even higher than the one shown in Fig. 3(a) for $n = 12$ with the small samples.

Considering that 13% of the plagiarism cases in the corpus are cross-language (cf. [15]) and the method we propose here has no hope of retrieving such cases, we consider that a recall above 0.85 is a very good result.

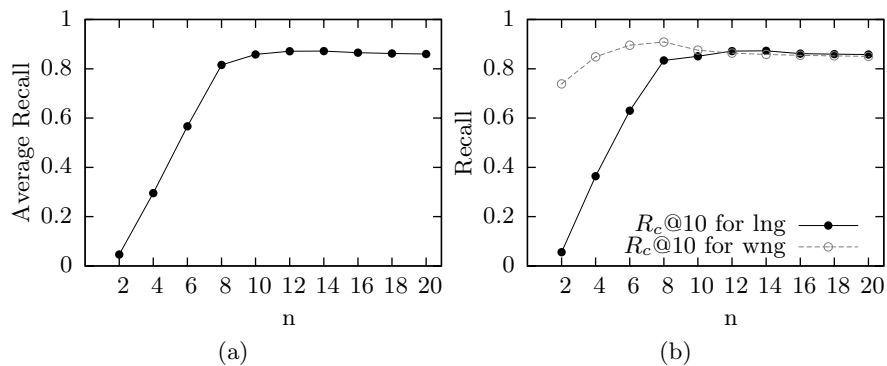


Fig. 3. Recall calculated as in Eq. (2) for the PAN-PC-09 corpus (a) by averaging over 100 samples of 150 random query texts and around 300 reference documents, with length encoding; and (b) compared to word n -grams.

Figure 3(b) shows the results of **exp2**. The obtained results confirm what we expected from Section 3: there exists a threshold for n , here around $n = 12$, above which the length encoding and the word n -gram methods are perfectly equivalent; to be true, here the encoding method performs always slightly better than word n -grams, for $n \geq 12$, but such small differences may not be reliable due to the fact that we are using a small subset of the corpus. This value of n is in concordance with the one used in fingerprinting models such as SPEX [4].

4.2 Experiments on the Co-derivatives Corpus

Experiment Outline. Even if (artificial) plagiarism and Wikipedia collaborative writing are very different phenomena, they can be considered as two sides of the same problem of text re-use identification, and can be stated in the same terms of query document - source documents association. Note, however, that from the viewpoint of the experimental setting there are two main differences between the two corpora. First of all, here the query set is composed of the last

revision of each article, and its 10 revisions, included itself, constitute the reference set: therefore, the set $\{d_q\}_{q \in Q}$ is in this case included in D , in agreement with [2]. Secondly, the set L_q of relevant sources for d_q has in this case a cardinality of 10 for each d_q ; therefore, it is even more natural here to choose $k = 10$ as the number of retrieved texts ($|D_q|$).

We followed for this corpus the same outline described in Section 4.1, except for the fundamental differences stated above. In the first experiment (**exp3** hereinafter) we calculated the Jaccard coefficient J_n for various values of n with word length n -grams. The second experiment (**exp4** hereinafter) was aimed at comparing the results obtained by considering the length encoding and the traditional word n -grams.

Measures of Performance. Since this corpus does not contain any character-level annotation for the revisions of Wikipedia articles, the only measure of recall which applies here is the global text average $R_t@k$, defined as follows:

$$R_t@k := \frac{\sum_q |L_q \cap D_q|}{\sum_q |L_q|} = \frac{\sum_q |L_q \cap D_q|}{k|Q|}, \quad (3)$$

i.e., simply the fraction of relevant documents that the method identifies as such. Since for this corpus the number of retrieved texts k corresponds, for each query text, to the number of relevant documents, the values of precision and recall coincide, i.e. $R_t@10 = P_t@10$.

Results. In Table 3 we report the recall $R_t@10$ for **exp3**, calculated as in Eq. (3), with $|L_q| = |D_q| = 10$ and $n \in \{2, 4, \dots, 20\}$.

Table 3. Recall $R_t@10$ for the co-derivatives corpus with the Jaccard coefficient on word-length n -grams, varying n

	2	4	6	8	10	12	14	16	18	20
en	0.02396	0.98970	0.99366	0.99465	0.99485	0.99485	0.99465	0.99426	0.99426	0.99406
de	0.22080	0.92911	0.96673	0.97663	0.97703	0.97604	0.97426	0.97208	0.97109	0.96812
es	0.10218	0.90159	0.96198	0.96812	0.96713	0.96495	0.96277	0.96040	0.95723	0.95485
hi	0.45545	0.74495	0.81683	0.84792	0.85010	0.84337	0.83525	0.82950	0.82257	0.81426

In concordance with [2], the results are much better for the English subcorpus than for the Hindi one; the other two languages are located in between, with quite good results. This could also be an effect of the difference in average length of the articles in the four different languages.

Table 4 shows the results of **exp4** with n ranging from 2 to 10 and for the Spanish corpus, which was chosen as the dataset here because the article length is proper and the similarity distribution is adequate for experiments. Again, the results of the two techniques are perfectly equivalent, with the length encoding performing slightly better than word n -grams for all values of n larger than 10.

Table 4. Recall $R_t@10$ for the Spanish section of the co-derivatives corpus, with word n -grams (wng) and length n -grams (lng), varying n

encoding	2	4	6	8	10	12	14	16	18	20
wng	0.9703	0.9762	0.9737	0.9697	0.9657	0.9622	0.9598	0.9566	0.9541	0.9497
lng	0.1022	0.9016	0.9620	0.9681	0.9671	0.9649	0.9628	0.9604	0.9572	0.9548

The very low recall obtained in all experiments with length bigrams has a very simple statistical explanation. Since the possible bigrams in the alphabet $\{1, \dots, 9\}$ are just $9^2 = 81$, and since we are considering only a combinatorial measure, disregarding any information about the frequency (this is the essence of the Jaccard coefficient), with high probability all the bigrams appear in each text of the corpus, giving a value 1 for J_2 in any case. Therefore, the selection of the first k neighbours corresponds to a random extraction of k source documents.

4.3 Experiments on Process Speed

We showed experimentally that the length n -gram model performs comparably to the word n -gram model for a proper value of n . Now, we compare the models in terms of processing speed.

In the first experiment (**exp5** hereinafter), we compared the time needed to encode a text document into either a set of word n -grams or a trie of length n -grams. For this estimation 1,000 random documents from the PAN-PC-09 corpus were considered.

In the second experiment (**exp6** hereinafter), we compared the time required to compare the document representation by calculating the Jaccard coefficient (*comparison*). In order to perform this experiment 50 suspicious and source documents from the PAN-PC-09 corpus were considered, resulting in 2,500 comparisons for each value of n .

The obtained results for both experiments are shown in Figure 4. In both cases different values of n were considered: $\{1, 3, 5, 9, 12\}$. From **exp5**, it is clear that the length encoding is much faster than the word encoding. On average, the length encoding takes a half of the time needed to perform the word encoding. This is due to two main reasons. First, in order to compare word n -grams the text must be converted to lowercase, an operation which is unnecessary for the length encoding. Additionally, as less memory is used to save the trie than the set of word n -grams, the resources are used in a more efficient way in the first case.

Experiment **exp6** clearly shows that also the time required to compare length n -grams is shorter than the time needed to compare word n -grams.

Disregarding the precise numerical results, which depend on the specific hardware used, the difference in performance is evident in both experiments and confirms this further advantage of length encoding.

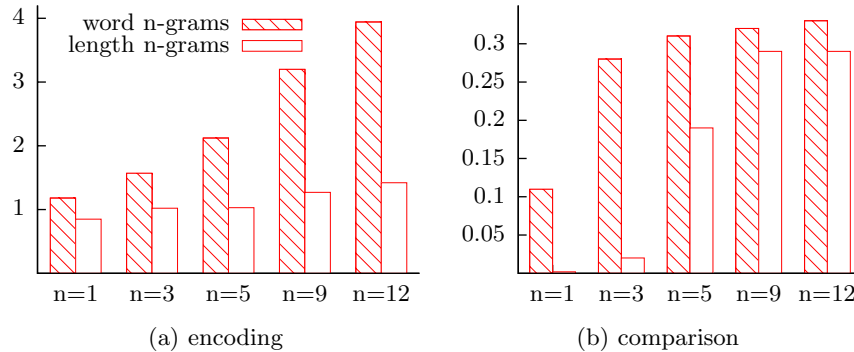


Fig. 4. Time needed for the encoding and comparison steps, by using word and length n -grams. The values are expressed in seconds and are averages of 2,500 processes.

5 Conclusions

In this paper we approached the problem of the preliminary selection of closely related texts, the first step for text-reuse, co-derivatives analysis, and automatic plagiarism detection. The method we proposed to solve this task encodes the documents on the basis of their word lengths. Whereas some efficient methods, such as fingerprinting, imply a loss of information between the actual document and its fingerprint, our method reduces such loss, as we empirically showed.

Retrieval experiments were performed on two corpora: the first one of simulated plagiarism and the second one of text co-derivatives. The obtained results show that representing the documents by the length encoding: (i) does not affect the performance of the retrieval process; (ii) favours a flexible comparison of documents as n -grams of any level are available in the text representation; and (iii) the entire encoding and comparison process is speeded up, an important factor when the amount of comparisons to perform is significant.

As future work we plan to combine this method with a selection of representative chunks on the basis of entropic methods. Moreover we will compare the similarity measure to a different one such as the Kullback-Leibler distance [11].

Acknowledgements. This work was partially funded by the CONACYT-Mexico 192021 grant, the Text-Enterprise 2.0 TIN2009-13391-C04-03 project, and the INdAM-GNFM Project for Young Researchers “Sequenze, sorgenti e fonti: sistemi dinamici per le misure di similarità”.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval, p. 192. Addison-Wesley Longman, Amsterdam (1999)
2. Barrón-Cedeño, A., Eiselt, A., Rosso, P.: Monolingual Text Similarity Measures: A Comparison of Models over Wikipedia Articles Revisions. In: Proceedings of the ICON 2009: 7th International Conference on Natural Language Processing, pp. 29–38. Macmillan Publishers, Basingstoke (2009)

3. Basile, C., Benedetto, D., Caglioti, E., Cristadoro, G., Degli Esposti, M.: A plagiarism detection procedure in three steps: selection, matches and “squares”. In: Stein, B., Rosso, P., Stamatatos, E., Koppel, M., Agirre, E. (eds.) SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2009), pp. 1–9. CEUR-WS.org (2009)
4. Bernstein, Y., Zobel, J.: A Scalable System for Identifying Co-Derivative Documents. In: Apostolico, A., Melucci, M. (eds.) SPIRE 2004. LNCS, vol. 3246, pp. 55–67. Springer, Heidelberg (2004)
5. Bigi, B.: Using Kullback-Leibler distance for text categorization. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 305–319. Springer, Heidelberg (2003)
6. Broder, A.Z.: On the Resemblance and Containment of Documents. In: Compression and Complexity of Sequences (SEQUENCES 1997), pp. 21–29. IEEE Computer Society, Los Alamitos (1997)
7. Clough, P., Gaizauskas, R., Piao, S., Wilks, Y.: Measuring Text Reuse. In: Proceedings of Association for Computational Linguistics (ACL 2002), Philadelphia, PA, pp. 152–159 (2002)
8. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles* 37, 547–579 (1901)
9. Jurafsky, D., Martin, J.H.: *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*, 2nd edn. Prentice-Hall, Englewood Cliffs (2009)
10. 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)
11. Kullback, S., Leibler, R.: On information and sufficiency. *Annals of Mathematical Statistics* 22(1), 79–86 (1951)
12. 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)
13. Maurer, H., Kappe, F., Zaka, B.: Plagiarism - A Survey. *Journal of Universal Computer Science* 12(8), 1050–1084 (2006)
14. Metzler, D., Bernstein, Y., Croft, B.W., Moffat, A., Zobel, J.: Similarity Measures for Tracking Information Flow. In: Conference on Information and Knowledge Management, pp. 517–524. ACM Press, New York (2005)
15. Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A., Rosso, P.: Overview of the 1st International Competition on Plagiarism Detection. In: Stein, B., Rosso, P., Stamatatos, E., Koppel, M., Agirre, E. (eds.) SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse, PAN 2009, pp. 1–9. CEUR-WS.org (2009)
16. Schleimer, S., Wilkerson, D.S., Aiken, A.: Winnowing: Local Algorithms for Document Fingerprinting. In: 2003 ACM SIGMOD International Conference on Management of Data. ACM, New York (2003)
17. Stein, B., Meyer zu Eissen, S., Potthast, M.: Strategies for Retrieving Plagiarized Documents. In: Clarke, C., Fuhr, N., Kando, N., Kraaij, W., de Vries, A. (eds.) 30th Annual International ACM SIGIR Conference, pp. 825–826. ACM, New York (2007)