Monolingual Text Similarity Measures: A Comparison of Models over Wikipedia Articles Revisions

Alberto Barrón-Cedeño[1], **Andreas Eiselt**[2] **and Paolo Rosso**[1]
[1]Natural Language Engineering Lab. - ELiRF
Department of Information Systems and Computation
Universidad Politécnica de Valencia, Spain
[2] Web Technology and Information Systems
Bauhaus-Universität Weimar, Germany
{lbarron, prosso}@dsic.upv.es , andreas.eiselt@uni-weimar.de

Abstract

Measuring the similarity of texts is a common task in detection of co-derivatives, plagiarism and information flow. In general the objective is to locate those fragments of a document that are derived from another text.

We have carried out an exhaustive comparison of similarity estimation models in order to determine which one performs better on different levels of granularity and languages (English, German, Spanish, and Hindi). In connection with the comparison we introduce a publicly available corpus specially suited for this task. Furthermore we introduce some modifications to well known algorithms in order to demonstrate their applicability to this task.

Amongst others, our experiments show the strengths and weaknesses of the different models with respect to the granularity of the processed texts.

1 Introduction

In Information Retrieval (IR) the selection of relevant documents from a set of documents D is a basic but important task. A query is often composed of a short set of keywords without further structure. Nevertheless a query may even consist of an entire document d_q. In this case each document $d \in D$ is ranked by its relevance in terms of its similarity to d_q.

Measuring the similarity or difference among a set of texts is relevant in different tasks such as information flow tracking (Metzler et al., 2005), document clustering and categorization (Bigi, 2003), multi-document summarization (Goldstein et al., 2000), version control (Hoad and Zobel, 2003), text reuse analysis (Clough et al., 2002) and plagiarism detection (Maurer et al., 2006).

Special interest is given to the analysis of co-derivatives. A co-derivative is defined as a pair of documents which are revisions or plagiarism of each other (Hoad and Zobel, 2003). We address the problem by analyzing the textual content of the implied texts (other methods perform the analysis by considering the document structure (Buttler, 2004)). Our main interest is the selection of good techniques for automatic plagiarism detection.

In this paper we present a comparison of different methods for the measurement of similarity between texts. The remainder is laid out as follows. Section 2 gives an overview of 7 methods for text similarity measurement covering vector space, fingerprinting, and probabilistic models. Section 3 describes the construction of a corpus for detection of monolingual derivatives. The corpus, derived from Wikipedia, is freely available and includes numerous texts in English, German, Spanish and Hindi. Section 4 describes the experiments carried out, which have been conducted at document- as well as section-level. The obtained results are discussed in the same section. Finally, in Section 5 we draw some conclusions and give an overview of our current work.

2 Similarity Measures

In order to measure the similarity value $sim(d, d_q)$ between two texts d and d_q, different types of approaches have been proposed. In general, we consider a similarity threshold $[0, 1]$. $sim(d, d_q) = 0$ implies that the documents d and d_q are not similar at all, whereas $sim(d, d_q) = 1$ reflects the equality of d and d_q. In those cases where the calculation of similarity may return results higher than 1, the values are normalized to fit the expected range (Sections 2.1.3 and 2.3.1-2.3.3). The methods presented are just outlined. A detailed description may be found in the included references. The common notation is summarized in Table 1.

Proceedings of ICON-2009: 7th International Conference on Natural Language Processing
Macmillan Publishers, India. Also accessible from http://ltrc.iiit.ac.in/proceedings/ICON-2009

Table 1: Description of the notation used

Elements	
d	Document
d_q	Query document
t	Term
D / D_q	Set of documents d / d_q
Measurements	
$tf_{t,d}$	Frequency of t in d
idf_t	Inverse document frequency
df_t	Number of documents containing t
$\langle x \rangle$	Absolute value of x
$\|X\|$	Cardinality of X (tokens)
$\|X\|_t$	Cardinality of X (types)
$sim(a, b)$	Similarity between a and b

2.1 Vector Space Models

In Vector Space Models (VSM) a document is represented as a vector of index terms. The two common representation schemes for the vectors are: (1) binary, in which the existence/non-existence of a term is indicated by 1/0 (Section 2.1.1) and (2) weighted, in which each term is weighted by values between 0 and 1 (Sections 2.1.2 and 2.1.3). The idea behind the VSMs is to carry out a comparison between vectors in order to define how close the represented texts are.

2.1.1 Jaccard Coefficient

The Jaccard coefficient (Jaccard, 1901) is a binary VSM in which a document d is represented by its vocabulary v_d. Due to its simplicity and quality, it is one of the most widely used boolean models in IR. The similarity between two documents is computed as:

$$sim(d, d_q) = J(d, d_q) = \frac{|v_d \cap v_{d_q}|}{|v_d \cup v_{d_q}|} \ . \quad (1)$$

The model simply considers the amount of shared terms between d and d_q with respect to the number of terms in the entire vocabulary.

2.1.2 Cosine Similarity

The cosine similarity measure is a weighted VSM model extensively used in IR. It calculates the similarity by using the Euclidean cosine rule:

$$cos(d, d_q) = \frac{\sum_{t \in d \cap d_q} \left(\omega_{t,d} \cdot \omega_{t,d_q}\right)}{\sqrt{\sum_{t \in d} \left(\omega_{t,d}\right)^2 \cdot \sum_{t_q \in d_q} \left(\omega_{t,d_q}\right)^2}} \ , \quad (2)$$

where $\omega_{t,d}$ is the weight of term t in document d. In order to weight the terms we have used the well-known term-frequency (Manning and Schütze, 1999). In this way, the similarity between two documents is estimated as:

$$sim(d, d_q) = \frac{\sum_{t \in d \cap d_q} \left(tf_{t,d} \cdot tf_{t,d_q}\right)}{\sqrt{\sum_{t \in d} \left(tf_{t,d}\right)^2 \cdot \sum_{t \in d_q} \left(tf_{t,d_q}\right)^2}} \ . \quad (3)$$

2.1.3 Word Chunking Overlap

This is another weighted VSM model (Shivakumar and García-Molina, 1995) and a classic method for copy-detection based on the so called *asymmetric subset measure* for document pairs. Such subset is defined as:

$$subset(d, d') = \frac{\sum_{t_i \in c(d,d')} tf_{t,d} \cdot tf_{t,d'}}{\sum_{t_i \in d} tf_{t_i,d}^2} \ , \quad (4)$$

where $c(d, d_q)$ is a closeness set containing those terms $t \in d \cap d_q$ matching the condition $tf_{t,d} \sim tf_{t,d_q}$. A term t belongs to the closeness set if the following condition is accomplished:

$$\epsilon - \left(\frac{tf_{t,d}}{tf_{t,d'}} + \frac{tf_{t,d'}}{tf_{t,d}}\right) > 0 \ . \quad (5)$$

The parameter ϵ defines how close the frequency of t in both documents must be in order to be included in the closeness set. In agreement with Shivakumar and García-Molina (1995), we consider $\epsilon = 2.5$. This value has offered a good balance between Precision and Recall in previous experiments over netnews articles.

The preliminary similarity between documents d and d_q based on word chunking overlap is defined as:

$$sim'(d, d_q) = max\left\{subset(d, d_q), subset(d_q, d)\right\} \ . \quad (6)$$

The value of $sim'(d, d_q)$ may be higher than 1. Due to this reason, the similarity between a query d_q and all the documents $d \in D$ must be normalized in order to fit the similarity range $[0, 1]$:

$$sim(d, d_q) = \frac{sim'(d, d_q)}{\max_{d' \in D} sim'(d', d_q)} \quad (7)$$

2.2 Fingerprint Models

Document fingerprinting is a family of models designed to efficiently compare texts by using a set of characteristics instead of its entire content. These characteristics are compiled to a so called fingerprint that represents the text. The following described algorithms use hashes to represent texts. The set of hashes selected from a document d composes its fingerprint H_d^*. The comparison of H_d^* and $H_{d_q}^*$ allows an approximate calculation of the similarity between documents d and d_q.

In this research we have considered two fingerprinting models. The first one is based on character-level chunks while the second one is based on word-level chunks (Sections 2.2.1 and 2.2.2).

Algorithm 1: Given the document d:

$d' = clean(d)$
$G = \{$sequence of q-grams in $d'\}$
Initialize H
for each q-gram $\in G$:
    $append(H, hash(q\text{-gram}))$
$W_H = create\_windows(H)$
// Winnowing
Initialize H^* // The selected hashes
for each $v \in W_H$:
    $h^* \Leftarrow min_{hash}(v)$
    $insert(H^*, (h^*, pos(h^*)))$
$fingerprint(d) = H^*$

Figure 1: Winnowing fingerprinting process. $clean()$ removes spaces and punctuation marks; $insert(S, x)$ inserts x to the sequence S; $create\_windows(H)$ generates a sequence of overlapping hash windows w, $|w| = t - q + 1$; $min_{hash}(v)$ gives the hash with minimum value in v, if more than one hash contains the minimum value, the rightmost is selected; $pos(h^*)$ is the absolute position of the hash value in the entire text (the first position is 0).

2.2.1 Winnowing Fingerprinting

Winnowing is a fingerprinting algorithm that uses character-level q-grams of d and d_q from which spaces and punctuation marks are preliminarily discarded (Schleimer et al., 2003).

The method is based on the selection of chunks obtained by a sliding window passing over the text. In order to select those chunks, which will be hashed to compose the fingerprint H_d^*, two parameters must be carefully defined: (1) the noise threshold q, which defines the level of the q-grams (the larger q is, the more sensible the method becomes with respect to modifications between d and d_q); and (2) the guarantee threshold t, which is used in order to define the length of the sliding window. The fingerprinting process is described in Fig. 1. We decided to use $q = 50$ and $t = 100$ as these values have previously given good results (Schleimer et al., 2003).

Two things have to be noted in this process: (1) given the sequence of hashes $h_1 h_2 \dots h_n$ each position $1 \leq i \leq n - w + 1$ defines a window $h_i \dots h_{i+w-1}$; (2) it is expected that different windows select the same hash value, so $|H^*| \ll |W_H|$.

The similarity is then approximated on the basis of the *resemblance* measure (Broder, 1997), which is defined as:

$$sim(d, d_q) = r(H_d^*, H_{d_q}^*) = \frac{|H_d^* \cap H_{d_q}^*|}{|H_d^* \cup H_{d_q}^*|} \ . \qquad (8)$$

Note that this is in fact the well-known Jaccard coefficient (Eq. 1).

2.2.2 SPEX algorithm

The idea behind SPEX is that "if any sub-chunk of any chunk can be shown to be unique, then the chunk in its entirety must be unique" (Bernstein and Zobel, 2004). This means that if a chunk 't_1 t_2' is unique, the chunk 't_1 t_2 t_3' is unique as well. Applying this to a collection of documents means that all hashes of word n-grams that occur only in one document could be discarded as they are not relevant.

Given a collection of documents D, the task is to identify those chunks appearing in more than one document $d \in D$. The first step is to generate a list h_1 of 1-grams over D and to count in how many documents each of them occur. In the next steps h_n is built by selecting only those n-grams g fulfilling the condition that h_{n-1} contains $g_{[0,n-1]}$ and $g_{[1,n]}$ and both are counted two times. This step is repeated until n reaches a given threshold l. In agreement with Bernstein and Zobel (2004), we consider $l = 8$. The similarity between documents d and d_q is computed as:

$$sim(d, d_q) = \frac{1}{mean(|d|, |d_q|)} \sum_{c \in d \wedge c \in d_q} 1 \ , \qquad (9)$$

where $mean(|d|, |d_q|)$ is the mean length of the documents d and d_q.

A weakness of the SPEX method is that D must be a closed set of documents. In order to add a new document to D the index of hashes h_l has to be built up from scratch.

2.3 Probabilistic Models

In this case a document is characterized by the probability associated to its tokens/words. In this way, the similarity between two documents can be approached by calculating the probability of their relation. We have considered three pseudo-probabilistic methods (their output is not ranged in $[0, 1]$). Due to this deviant behavior further calculations must be carried out in order to normalize the values.

2.3.1 Machine Translation

In statistical Machine Translation (MT), the task is: given a text e written in a language L, to find the most likely translation f, in a language L'. One of the most well-known models in MT is the IBM Model 1 (M1) (Brown et al., 1993).

In this approach we adapt it to estimate the similarity between monolingual texts. By considering $L = L'$ the application of M1 has reached promising results in monolingual IR (Berger and Lafferty, 1999). In fact, adaptations of the M1 have been already applied to monolingual measures of similarity between sentences (Metzler et al., 2005) and even to cross-language plagiarism analysis (Pinto et al., 2009).

In the cross-language case the estimation of translation probability and similarity may be joined into a single process (Barrón-Cedeño et al., 2008). The same could be applied for the "monolingual" translation. On the basis of the M1, we define the similarity measure between two documents as:

$$sim(d, d_q) = \varrho(d)\ w(d_q \mid d)\ . \quad (10)$$

$\varrho(d)$ is a *length model probability* that depends on the expected length of the translation of $d_q \in L$ to $d \in L'$ (as in this case $L = L'$, $\varrho(d) = 1$). $w(d_q \mid d)$ is a tailored version of the known as *translation model probability* (Brown et al., 1993). It is defined as:

$$w(d_q \mid d) = \prod_{x \in d_q} \sum_{y \in d} p(x, y)\ , \quad (11)$$

where $p(x, y)$ is a dictionary containing the probability that word x is a translation of word y. As we are not performing an actual translation, it is assumed that $p(x, y) = 1$ if $x = y$ and 0 otherwise. While Eq. 11 shows good results in the processing of sentences, it has to be adapted as in Eq. 12 in order to handle entire documents.

$$w(d_q \mid d) = \sum_{x \in d_q} \sum_{y \in d} p(x, y)\ . \quad (12)$$

For each word $x \in d_q \setminus d$, a penalization ϵ is applied to $w(d_q \mid d)$. We consider $\epsilon = -0.1$. As the obtained result may exceed the range $[0, 1]$, the same normalization as for the word chunking overlap method is applied (Eq. 7).

2.3.2 Kullback-Leibler Distance

The Kullback-Leibler distance (KL_δ) is a symmetric version of the Kullback-Leibler Divergence (Kullback and Leibler, 1951). This measure has been applied to text clustering (Bigi, 2003) as well as plagiarism analysis (Barrón-Cedeño et al., 2009). Given an event space, KL_δ is defined as in Eq. 13 (Bigi, 2003). Over a feature vector $\mathcal{X}$, it measures how close two probability distributions P and Q are.

$$KL_\delta(P_{d_q} \mid\mid Q_d) = \sum_{x \in \mathcal{X}} (P(x) - Q(x)) log \frac{P(x)}{Q(x)}\ . \quad (13)$$

P_{d_q} and Q_d are probability distributions composed of a set of features (terms) characterizing d and d_q. The probability distribution P_{d_q} is composed of the top 20% of the terms in d_q ranked by the standard *tf-idf*. The probability associated to the selected terms is $P(t \mid d_q) = tf_{t,d_q} / \sum_{t' \in d_q} tf_{t',d_q}$. In order to compare d_q to d, the probability distribution Q_d must be composed of the same terms of P_{d_q}. Due to the fact that there will be terms such that $t \in P_{d_q} \setminus Q_d$, the probabilities associated to the terms in Q_d must be smoothed with respect to their tf value (Barrón-Cedeño et al., 2009).

KL measures the distance instead of the similarity. A value of $KL_\delta(P_{d_q} \mid\mid Q_d) = 0$ implies that $P_{d_q} = Q_d$ and the implied documents are quite similar. In this case the final similarity between d_q and the documents in D is estimated as:

$$sim(d, d_q) = -\left(\frac{KL_\delta(P_{d_q} \mid\mid Q_d)}{max_{d'} KL(P_{d_q} \mid\mid Q_d)} - 1 \right)\ . \quad (14)$$

2.3.3 Okapi BM25

The BM25 weighting scheme extends the approach of idf by additionally considering tf and document length (Spärck Jones et al., 2000). Including newer variants such as BM25F (Zaragoza et al., 2004), it represents one of the state-of-the-art approaches in query based document retrieval. It can be formalized as:

$$BM25(d, d_q) = \sum_{t \in d_q} idf_t \cdot \alpha_{t,d} \cdot \beta_{t,d_q}\ , \quad (15)$$

where

$$\alpha_{t,d} = \frac{(k_1 + 1)\, tf_{t,d}}{k_1 \left((1 - b) + b \cdot \frac{|d|}{L_{avg}} \right) + tf_{t,d}}\ . \quad (16)$$

$k_1 \geq 0$ and $0 \leq b \leq 1$ are used in order to calibrate the document term frequency and document length scaling. $k_1 = 0$ corresponds to a binary model (term frequency is not considered). Considering $b = 0$ corresponds to no length normalization whereas $b = 1$ corresponds to a full scaling of the term weight to the document length. In agreement with Spärck Jones et al., (2000) we consider $k_1 = 1.2$ and $b = 0.75$. Finally, L_{avg}

is the average document length in the collection. β_{t,d_q} is defined as:

$$\beta_{t,d_q} = \frac{(k_3+1)\,tf_{t,d_q}}{k_3 + tf_{t,d_q}} \quad . \tag{17}$$

β is used to normalize the tf of the terms in d_q. Due to the fact that the queries in our experiments consist of full-text, we consider a value of $k_3 = 2$. The values k_1 of α and k_3 of β are calibrators of the tf. Okapi BM25 is a ranking method (the estimated values are not in the range $[0, 1]$). The values obtained by the function $BM25(d, d_q)$ must be normalized as in Eq. 7 in order to estimate similarity.

3 Corpus Construction

The corpus has been generated on the basis of Wikipedia articles. Wikipedia has been frequently used as source in other related research, for example in near-duplicates detection (Potthast and Stein, 2008). In Section 3.1 we describe how we have acquired the Wikipedia documents. Section 3.2 describes the construction and preprocessing of the corpus[1].

3.1 Documents Acquisition

The corpus was composed on the basis of the following three rules: (1) the languages considered are English, German, Spanish and Hindi (en, de, es, hi); (2) the set of documents consists of the 500 most frequently accessed articles in each language; and (3) for each article we obtained 10 revisions that were, as far as possible, equally distributed over the 500 most recent revisions.

Wikipedia articles are often affected by vandalism (Potthast et al., 2008), which particularly describes the deletion or modification with malicious intention. In order to avoid the consideration of such content, revisions that have been rejected by reviewers were not included into the corpus. The same applies for revisions with only minimal changes to assure that each revision has a different level of similarity with respect to the newest revision of the article (such characteristics are specially tagged in Wikipedia).

The corpus pre-processing includes whitespace normalization, sentence detection, tokenization and case folding. In numerous IR applications stemming is used to improve the results. However, in plagiarism and co-derivative detection this is not the case. Previous experiments have shown that in these tasks stemming does not improve the results and can even deteriorate them (Hoad and Zobel, 2003; Barrón-Cedeño and Rosso, 2009).

[1]The corpus is available at http://users.dsic.upv.es/grupos/nle/downloads.html

Table 2: Corpus statistics (per document). $D_q \rightarrow$ collection of query-documents; $D \rightarrow$ collection of document revisions ($D_q \subset D$), $|d_{avg}|_t \rightarrow$ average number of types per document; $|d_{avg}| \rightarrow$ average number of tokens per document; $|D|_t \rightarrow$ types in D

Lan	$\lvert D_q\rvert$	$\lvert D\rvert$	$\lvert d_{avg}\rvert_t$	$\lvert d_{avg}\rvert$	$\lvert D\rvert_t$
	Before stopwords elimination				
de	500	5,000	1,812	5,229	261,370
en	500	5,000	2,243	8,552	183,414
hi	500	5,000	302	672	78,673
es	500	5,000	1,216	4,116	133,595
	After stopwords elimination				
de	500	5,000	1,707	3,474	261,146
en	500	5,000	2,149	6,008	183,288
hi	500	5,000	270	495	78,577
es	500	5,000	1,142	2,415	133,339

3.2 Corpus Composition

In order to compose an experimental framework we have defined a text collection D for each language that consists of Wikipedia articles A_n. Each article A_n is represented by 10 revisions $\{d_{n,1}, \ldots, d_{n,10}\}$. Furthermore we define D_q as a set of query-documents $\{d_{1,1}, \ldots, d_{m,1}\}$ with $D_q \subset D$ assuming that $d_{n,1}$ is the most recent revision of the article A_n and m is the total number of articles in the corpus. By defining $D_q \subset D$, we aim to consider samples of co-derivatives which are in fact exact copies. Some corpus statistics are included in Table 2.

Figure 2 shows, for each language, the average evolution of similarity among the different articles revisions with respect to d_q, the newest one. It can be observed that the similarity decreases for more distant revisions. The evolution of the English revisions is clearly slighter than in the other languages, whereas the evolution of Spanish and German seems quite similar. On the opposite the revisions in Hindi show an obviously stronger evolution. This factor will be relevant during the analysis of the experiments results (Section 4.3). The tendency of the similarity in the four languages might be explained by the maturity of the articles (a topic for further research).

4 Experiments

In order to evaluate the different similarity measures, we carried out experiments considering different languages, text lengths and similarity levels.

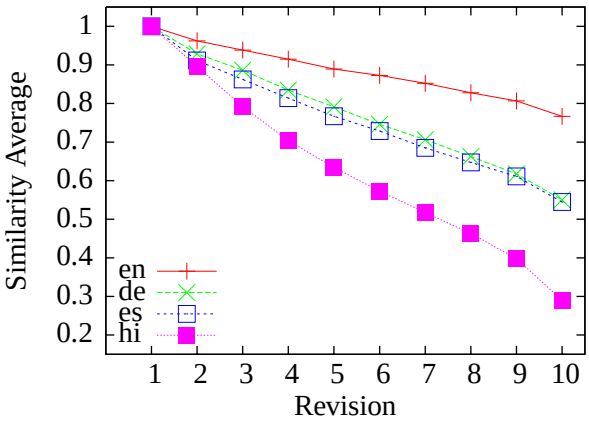

Figure 2: Evolution of the similarity between d_q and its preceding revisions. Similarities estimated by the Jaccard Coefficient.

Table 3: Corpus statistics (per section). $D'_q \rightarrow$ collection of query-sections; $D^* \rightarrow$ sections of all documents in D; $|d^*_{avg}|_t \rightarrow$ average number of types per section; $|d^*_{avg}| \rightarrow$ average number of tokens per section; $|D^*|_t \rightarrow$ types in D^*

Lan	$\|D'_q\|$	$\|D^*\|$	$\|d^*_{avg}\|_t$	$\|d^*_{avg}\|$	$\|D^*\|_t$
	Before stopwords elimination				
de	7726	133,171	124	198	261,370
en	8043	114,216	187	378	183,414
hi	345	27,127	76	125	78,673
es	4696	86,092	126	241	133,595
	After stopwords elimination				
de	7726	133,171	98	132	261,146
en	8043	114,196	159	266	183,288
hi	345	27,125	64	92	78,577
es	4696	86,076	103	142	133,339

The following two sections describe the experiments as well as the metrics used for evaluation. Section 4.3 discuses the obtained results.

4.1 Experiments Description

The approached problem is the detection of co-derivatives given a query text. Such detection process requires the analysis of the similarity between texts of different lengths. Therefore, the experiments have been divided into two independently evaluated stages: (1) document level analysis and (2) section level analysis[2]. Following, we describe both parts of the experiment.

Document level analysis

For each document $d_q \in D_q$ the documents in D are ranked with respect to their similarity $sim(d, d_q)$ (Section 2). The ranking is defined as a list r_q of documents, which is sorted in descending order with respect to the similarity between d and d_q. Hence, it is expected that d_q is co-derived from the documents on top of r_q with high probability. r_q is the input for the following stage.

Section level analysis

The sections corresponding to the top 50 documents in r_q are considered in order to compose the set D' of co-derivative candidate sections. Furthermore D'_q is composed of sections in $d_q \in D_q$. In order to perform an objective evaluation of this stage: (a) D'_q is composed only of those sections of d_q which have been equally named in the corresponding 10 revisions; and (b) the sections of all revisions of d_q have to be included in D' even if the corresponding revisions were not under the top 50 documents in the ranking. Statistics of the sections in D can be found in Table 3.

For each section $d'_q \in D'_q$ the sections in D' are ranked with respect to their similarity $sim(d', d'_q)$. The ranking is defined as a list r'_q of sections d' that are sorted in descending order with respect to their similarity to d'_q. Again, it is expected that those sections in the top of r'_q are actual co-derivatives of d'_q.

4.2 Evaluation Metrics

A text d is considered relevant to d_q if d_q is a co-derivative text of d. In order to estimate how well the models retrieve the relevant documents for d_q, our evaluations are based on the Precision and Recall metrics, defined as in Eqs. 18 and 19 (Manning and Schütze, 1999). In order to calculate them, an amount of m documents is retrieved from top of r_q. For Precision, m is set to 10 $(P@10)$ as 10 is the amount of relevant texts for each query. Recall $(R@m)$ is measured by considering $m = \{10, 20, 50\}$. Note that in this case $P@10$ and $R@10$ are equal as the number of relevant and retrieved documents from r_q is the same.

$$P = \frac{|relevant\,documents\,retrieved|}{|documents\,retrieved|}\ , \qquad (18)$$

$$R = \frac{|relevant\,documents\,retrieved|}{|relevant\,documents|}\ . \qquad (19)$$

Additionally, two measures specifically designed to evaluate methods for co-derivatives analysis are considered (Hoad and Zobel, 2003): Highest False Match (HFM) and Separation (sep). Such measures have been designed to estimate the distance of the correctly retrieved documents in r_q with respect to those incorrectly re-

[2]We take advantage of the Wikipedia articles structure, where the different sections are explicitly tagged.

trieved and are only significant if they are considered together. In order to estimate HFM and sep, all relevant documents for d_q have to be included among the retrieved documents, i.e. $R@50 = 1.0$.

Given a ranking of documents r_q for the query document d_q, the maximum similarity value s^* is defined as $s^* = \max_{d \in D} sim(d \in r_q, d_q)$. It represents a similarity percentage of 100% with respect to d_q. We define d^- as the highest ranked document which is not relevant concerning d_q. HFM is the similarity percentage assigned to d^- and is computed as:

$$HFM = \frac{100 \cdot sim(d^-, d_q)}{s^*} \quad (20)$$

On the other hand, we define d^+ as the lowest ranked document which is relevant concerning d_q. LTM, the Lowest True Match, is computed as $LTM = 100 \cdot sim(d^+, d_q)/s^*$. The separation is defined as $sep = LTM - HFM$ and can be simply computed as :

$$sep = \frac{100 \cdot \big(sim(d^+, d_q) - sim(d^-, d_q)\big)}{s^*} \quad . \quad (21)$$

Note that $sep > 0$ implies that the highest rated documents in r_q are all those related to d_q. A value of $sep < 0$ means that other documents were ranked before those relevant to d_q.

By considering the $R@m$, we measure how many relevant documents have been ranked under the top m documents in r_q. By considering HFM and sep we additionally estimate how good the relevant and irrelevant documents are differentiated in the final ranking.

4.3 Results Discussion

The obtained results for the four languages are summarized in Fig. 3. In order to analyze these results, they have to be interpreted taking into account the statistics shown in Fig. 2. The first observation that could be made is that the values of $R@10$ are in the majority of cases nearly equal to $R@20$ and $R@50$. This means that the relevant documents for d_q (the query text) are concentrated in the top-10 of the ranking.

For the experiments on document level (Fig. 3(a)) the results obtained for English, German, and Spanish by the different methods are quite similar. The only exception appears in the case of Okapi BM25. The reason of this behavior is that this method is actually designed for keyword based retrieval. Even by tuning the implied parameters, the results are not comparable to those obtained by the other methods.

By comparing the results of all four languages it might be erroneously considered that the retrieval of documents in Spanish and Hindi is more complicated. However, the reason for the worse results is in fact justified by Fig. 2. While the actual similarity between documents is decremented, the retrieval task becomes more complicated. The figure shows that for instance the difference between the first and the last revision in the English articles is in average 0.23. Hindi at the other extreme shows an average similarity distance of 0.72. We advocate that further investigation should be done on the process of discriminating derivatives from documents on the same topic.

As we have established, different methods obtain similar results in terms of Recall. In order to determine which one is better, it is necessary to consider the HFM and sep of the rankings (Fig. 3(c)). It is clear that at document level the best approaches are those based on fingerprinting. For Winnowing the value assigned to the HFM is on average only 2.8%. In the case of SPEX the values are quite similar. Additionally, the separation values are also the highest in these methods. This means that there is a clear border between relevant and irrelevant documents.

With respect to the second experiment, carried out at section level (Fig. 3(b)), the supremacy of the fingerprinting models is not maintained any more. The reason is that if two entire documents have a fingerprint collision, it is highly probable that they are related and, in some cases, co-derived. However, at section level shorter texts are represented by few hashes. Over such conditions the probability of a fingerprint collision decreases. Due to this reason, some co-derived sections are not retrieved properly. It must also be considered that the input to this stage is a set of documents that are already highly related to the query. Due to this optimal conditions, the quality of the final output is much better than in the first step.

In this case Okapi BM25, that in the previous experiment performed worst, now obtained comparable results to the other vector and probabilistic models in terms of Recall. It is again necessary to look at the HFM and sep values in order to figure out which methods perform better. Jaccard, Cosine and MT have practically the same quality in terms of Recall, HFM and sep. Due to the

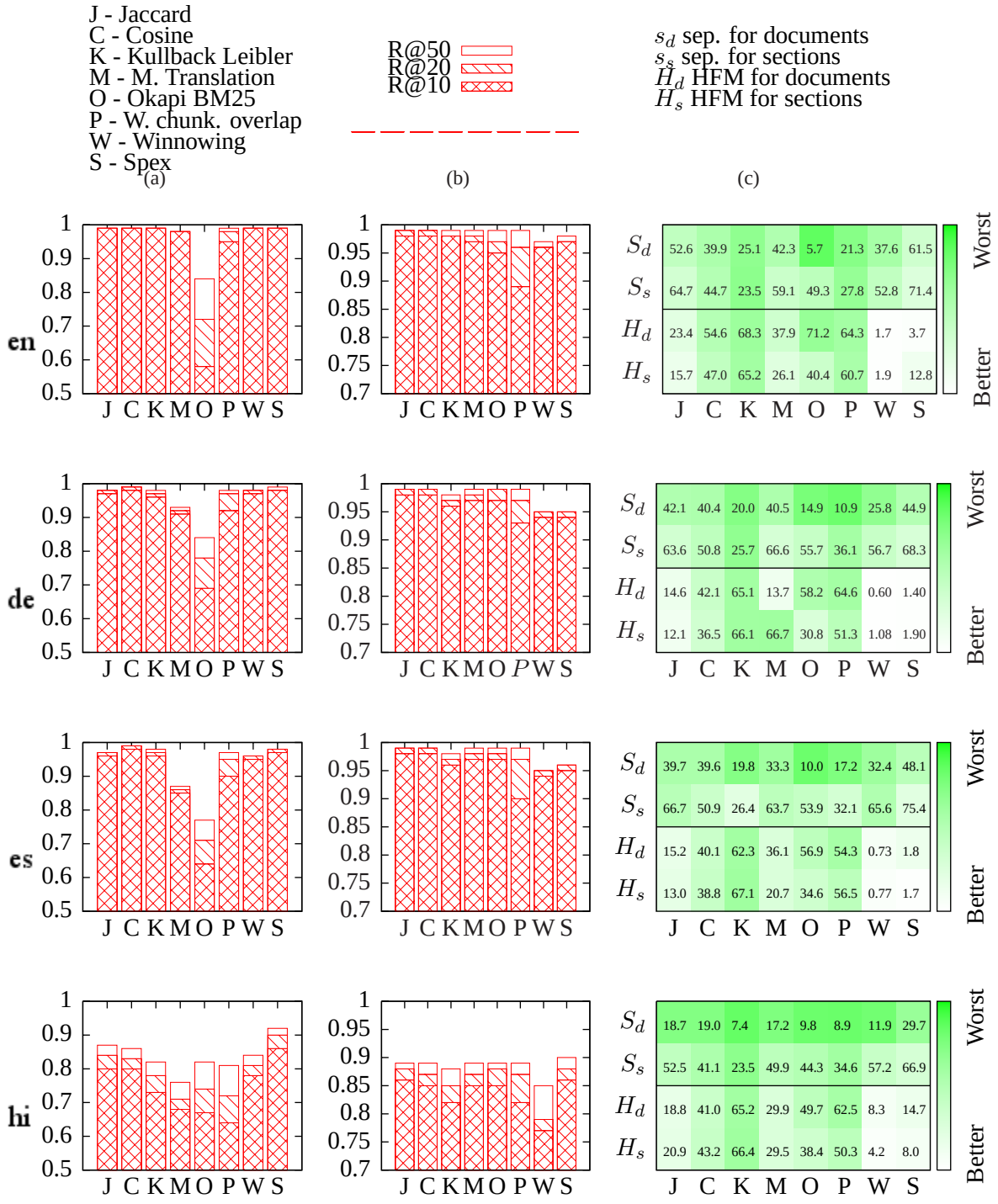

Figure 3: Obtained results. In (a) we show the results of the comparison at document-level, whereas in (b) we show the results of the comparison at section-level (both in *R*ecall terms). In (c) we show the *sep*aration and HFM together. For each square the first and third row shows Sep/HFM at document-level, whereas the second and fourth row shows Sep/HFM at section-level.

Table 4: Stopword removal and HFM experiments. SWR shows if the best results have been obtained by previously applying the stop word removal or not. $\%E_{HFM}$ represents the percentage of experiments for which it was possible to estimate HFM and sep for documents / sections.

Model	SWR	$\%E_{HFM}$			
		en	de	es	hi
Jaccard	YES	96/98	91/97	88/98	60/76
Cosine	YES	97/98	96/97	95/98	63/76
KL	NO	98/98	93/94	93/95	56/73
MT	YES	94/97	77/95	65/96	37/74
Okapi BM25	YES	79/98	79/97	69/98	62/77
W.C. overlap	YES	94/97	93/95	89/96	56/75
Winnowing	NO	97/92	90/87	87/88	56/62
SPEX	NO	96/95	82/92	80/92	48/61

simplicity, the Jaccard coefficient seems to be the best option in open retrieval environments (when the collection of documents D is not predefined). Reducing the q-grams and window levels of Winnowing might be a good option for closed retrieval environments (when the collection of documents D is closed and predefined).

Both experiments have been carried out before and after applying stopword removal (Fig. 3 only includes the best obtained results). Table 4 shows in which cases the best results have been obtained before or after stopword removal. In fact, the difference between applying a stopword removal or not is minimal in Recall terms. However, in terms of HFM and sep, which represent the quality that we could expect from the compared models, the difference becomes larger.

A different way of comparing the models is presented in Table 4. The percentage of comparisons in which it was possible to calculate HFM and sep specifies in how many cases all the relevant documents for a query have been included among the top-50 documents in r_q. For entire documents cosine similarity and Kullback-Leibler perform better. At section level Okapi BM25, Jaccard and cosine are the best options.

5 Conclusions

In this paper we have analyzed and compared different text similarity models for co-derivative and plagiarism detection. The following models have been applied without further adaptation: Jaccard Coefficient, Cosine Similarity, Word Chunk Overlap, Okapi BM25, Winnowing and SPEX. Additionally, two models have been adapted in order to measure similarity between texts: Kullback-Leibler distance and Machine Translation. Evaluations have been included in terms of Recall and Precision as well as Highest False Match and separation. Combining such measures makes it possible to estimate not only if all the relevant documents have been retrieved, but also the distance between the similarity values calculated for relevant and irrelevant documents. By considering these three factors more comprehensive information is available to select the most suitable method.

We have carried out experiments at document and section level over a corpus composed of revisions of Wikipedia articles. The obtained results show that, as it is expected, at document level Winnowing and SPEX have the best results. The advantage of Winnowing is that the generation of a fingerprint for a given document is independent from the others. However it must be considered that if derivation or plagiarism implies further modifications, Winnowing does not seems to be the better option. This is reflected in the experiment carried out at section level. In this case the statistical and vector space models (Jaccard coefficient, cosine measure, Kullback-Leibler distance and Machine Translation outperform those based on fingerprints.

In our current work we are designing a method for the automatic alignment of derived sentences in documents. With this information, it will be possible to carry out further experiments at sentence level. Additionally, in order to accurately compare the difference in the retrieval complexity for different languages, further experiments must be carried out by considering documents with closer similarity thresholds.

Acknowledgements

We would like to thank the Wikimedia Foundation for making the Wikipedia contents freely available. This work has been partially supported by the TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 research project as well as the CONACyT-Mexico 192021 grant.

References

Alberto Barrón-Cedeño and Paolo Rosso. 2009. On the Relevance of Search Space Reduction in Automatic Plagiarism Detection. In *Proceedings of the 25th Annual Conference of the Spanish Society for Natural Language Processing*, Donostia-San Sebastian, Spain. Spanish Society for Natural Language Processing.

Alberto Barrón-Cedeño, Paolo Rosso, David Pinto, and Alfons Juan. 2008. On Cross-lingual Plagiarism Analysis Using a Statistical Model. In Stein, Stamatatos, and Koppel, editors, *Proceedings of the ECAI'08 PAN Workshop: Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 9–13, Patras, Greece.

Alberto Barrón-Cedeño, Paolo Rosso, and José-Miguel Benedí. 2009. Reducing the Plagiarism Detection Search Space on the Basis of the Kullback-Leibler Distance. In Alexander F. Gelbukh, editor, *CICLing 2009*, volume 5449 of *Lecture Notes in Computer Science*, pages 523–534, Mexico, Mexico. Springer.

Adam Berger and John Lafferty. 1999. Information Retrieval as Statistical Translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–229, Berkeley, CA. ACM.

Yaniv Bernstein and Justin Zobel. 2004. A Scalable System for Identifying Co-Derivative Documents. In *Proceedings of the Symposium on String Processing and Information Retrieval*, pages 55–67. Springer.

Brigitte Bigi. 2003. Using Kullback-Leibler Distance for Text Categorization. In *Proceedings of the 25th ECIR'03, Springer-Verlag*, volume LNCS (2633) Advances in Information Retrieval, pages 305–319, Pisa, Italy.

Andrei Z. Broder. 1997. On the Resemblance and Containment of Documents. In *Compression and Complexity of Sequences (SEQUENCES'97)*, pages 21–29. IEEE Computer Society.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

David Buttler. 2004. A Short Survey of Document Structure Similarity Algorithms. In *5th International Conference on Internet Computing*, pages 152–159, Las Vegas, NV.

Paul Clough, Robert J. Gaizauskas, Scott L. Piao, and Yorick Wilks. 2002. Measuring Text Reuse. In *Proceedings of Association for Computational Linguistics (ACL2002)*, pages 152–159, Philadelphia, PA.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-Document Summarization By Sentence Extraction. In *NAACL-ANLP 2000 Workshop on Automatic Summarization*, pages 40–48, Seattle, WA. Association for Computational Linguistics.

Timothy C. Hoad and Justin Zobel. 2003. Methods for Identifying Versioned and Plagiarized Documents. *Journal of the American Society for Information Science and Technology*, 54(3):203–215.

Paul Jaccard. 1901. 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.

S. Kullback and R.A. Leibler. 1951. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

Hermann Maurer, Frank Kappe, and Bilal Zaka. 2006. Plagiarism - A Survey. *Journal of Universal Computer Science*, 12(8):1050–1084.

Donald Metzler, Yaniv Bernstein, W. Bruce Croft, Alistair Moffat, and Justin Zobel. 2005. Similarity Measures for Tracking Information Flow. In Chowdhury, Fuhr, Ronthaler, Schek, and Teiken, editors, *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 517–524, Bremen, Germany. ACM Press.

David Pinto, Jorge Civera, Alberto Barrón-Cedeño, Alfons Juan, and Paolo Rosso. 2009. A Statistical Approach to Crosslingual Natural Language Tasks. *Journal of Algorithms*, 64(1):51–60.

Martin Potthast and Benno Stein. 2008. New Issues in Near-Duplicate Detection. *Data Analysis, Machine Learning and Applications*, pages 601–609.

Martin Potthast, Benno Stein, and Robert Gerling. 2008. Automatic Vandalism Detection in Wikipedia. In Macdonald, Ounis, Plachouras, Ruthven, and White, editors, *30th European Conference on IR Research, ECIR 2008, Glasgow*, volume 4956 LNCS of *Lecture Notes in Computer Science*, pages 663–668, Berlin Heidelberg New York. Springer.

Saul Schleimer, Daniel S. Wilkerson, and Alex Aiken. 2003. Winnowing: Local Algorithms for Document Fingerprinting. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, New York, NY. ACM.

Narayanan Shivakumar and Hector García-Molina. 1995. SCAM: A Copy Detection Mechanism for Digital Documents. In *Proceedings of the 2nd Annual Conference on the Theory and Practice of Digital Libraries*.

Karen Spärck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments. In *Information Processing and Management*, volume 36, pages 779–840.

Hugo Zaragoza, Nick Craswell, Michael Taylor, Suchi Saria, and Stephen Robertson. 2004. Microsoft Cambridge at TREC-13: Web and HARD tracks. In *Proceedings of TREC 2004*.