Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection

Lab Report for PAN at CLEF 2010

Salha Alzahrani[1], Naomie Salim[2]

[1]Faculty of computer Science and Information Systems, Taif Univeristy, Taif, Saudi Arabia
[2]FSKSM, Universiti Teknologi Malaysia, Johor Bahru, Malaysia
s.zahrani@tu.edu.sa, naomie@utm.my

Abstract. This report explains our plagiarism detection method using fuzzy semantic-based string similarity approach. The algorithm was developed through four main stages. First is pre-processing which includes tokenisation, stemming and stop words removing. Second is retrieving a list of candidate documents for each suspicious document using shingling and Jaccard coefficient. Suspicious documents are then compared sentence-wise with the associated candidate documents. This stage entails the computation of fuzzy degree of similarity that ranges between two edges: 0 for completely different sentences and 1 for exactly identical sentences. Two sentences are marked as similar (i.e. plagiarised) if they gain a fuzzy similarity score above a certain threshold. The last step is post-processing whereby consecutive sentences are joined to form single paragraphs/sections. Our performance measures on PAN'09 training corpus for external plagiarism detection task (recall=0.3097, precision=0.5424, granularity=7.8867) indicates that about 54% of our detections are correct while we detect only 30% of the plagiarism cases. The performance measures on PAN'10 test collection is less (recall= 0.1259, precision= 0.5761, granularity= 3.5828), due to the fact that our algorithm handles external plagiarism detection but neither intrinsic nor cross-lingual. Although our fuzzy semantic-based method can detect some means of obfuscation, it might not work at all levels. Our future work is to improve it for more detection efficiency and less time complexity. In particular, we need to advance the post-processing stage to gain more ideal granularity.

1 Introduction

Plagiarism could be more fuzzy than clear, more complex than trivial copy and paste. Methods for plagiarism detection mostly track verbatim plagiarism; however, detecting excessive paraphrasing is a difficult task. Many current techniques rely on exactly matched substrings or some kinds of textual fingerprinting. But that may not be sufficient as cases of rephrasing and rewording the content treated as different (i.e. not plagiarised). Therefore, this work considers the problem of *finding the suspected fragments that have the same semantics with the same/different syntax.* In this regard, matching fragments of text becomes approximate or vague and can be implemented as a spectrum of values between 1 (i.e. exactly matched) and 0 (entirely different). The scale can be defined in a way similar to a human's judgement as in Figure 1.

Our work is similar to the work by Yerra and Ng (2005). In their paper, a copy detection approach for web documents was developed using fuzzy information retrieval (IR) model. The fundamental concept in fuzzy IR shows that words in a document have certain degree with a fuzzy set that contains words with related meaning and two documents are considered similar although their semantic content may be different if they gain high similarity degree with the fuzzy set. Thus, fuzzy IR has proved to work well for partially related semantic content in web retrieval. Subsequent to the previous work, Koberstein and Ng (2006) developed a reliable tool using fuzzy IR for determining the degree of similarity between two web documents and clustering the collection based on words. In addition, Alzahrani and Salim (2009) adapted the fuzzy IR model for use with Arabic scripts. By using Arabic plagiarism corpus of 4477 source statements and 303 query/suspicious statements, the similarity score of two documents is the averaged similarity among statements treated as plagiarised even if they are restructured or reworded. Experimental results showed that fuzzy IR can find to what extent two Arabic statements are similar or dissimilar. On the other hand, semantic similarity between short passages can be obtained by using the information extracted from a structured lexical database and corpus statistics (Li et al., 2006). The similarity of two sentences is derived from two similarities: semantic and order. The semantic vectors for two pairs of sentences are obtained by using unique terms in both sentences along with their synonyms from WordNet besides term weighting in the corpus. The order similarity defines that different words order may convey different meaning and should be count into total string similarity. Our work combines the fuzzy similarity model (Yerra and Ng, 2005) and semantic similarity model derived from a lexical database (Li et al., 2006). Instead of constructing a fuzzy thesaurus derived from word-to-word correlation factors as in Yerra and Ng's (2005) model, we choose to work with synonyms extracted from WordNet lexical database as in Li et al. (2006).

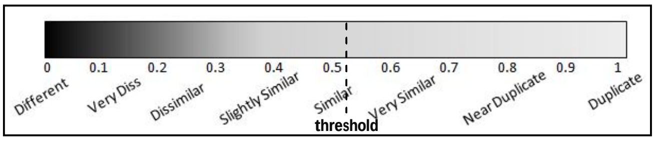


Figure1. Fuzzy Similarity Indication with Vague Boundaries

2 Problem Statement

In this report, we have considered the problem stated as follows.

Problem: Given a suspicious document dataset D_q and a large source collection D**,** find all suspicious parts s_q from $d_q{:}d_q \epsilon D_q$ that are similar to parts s_x from $d_x{:}d_x \epsilon D_x$ based on fuzzy semantic-based similarity approach as will be described in section 3.

Requirements: First, extract a set of features for each $d_q \epsilon D_q$ and $d \epsilon D$. Second, find a list of most promising documents D_x where $D_x \subset D$ based on shingling and Jaccard similarity coefficient known in IR. Third, perform sentence-wise in-depth analysis using fuzzy semantic-based approach. Last, perform post-processing operations to merge subsequent similar statements into passages or paragraphs.

Limitations: Neither intrinsic (i.e. variations in writing styles) nor cross-language plagiarism detection is handled by this algorithm. That is, the languages of both suspicious and candidate documents are considered homogeneous.

3 External Plagiarism Detection

3.1 Operational Framework

We implement an algorithm that tackles monolingual external plagiarism detection. In particular, it is designed to detect different degrees of obfuscation by using a fuzzy semantic-based approach. The operational framework is shown in Figure 2. The process starts with a set of source collection D and suspicious/query documents D_q. Then new representatives are generated d' and d'_q for each cleansed and tokenised d_q and d documents respectively. d' and d'_q are then used for shingling and computing the similarity between the shingles. The list of most similar documents for each $d_q \epsilon D_q$ is called candidate set D_x whereby $D_x \subset D$. After generating D_x, more sentence-wise detailed analysis is performed to obtain similar sentences (s_q, s_x) where $s_q \in d_q$, $s_x \in d_x$. The similarity score is gained by implementing a fuzzy semantic-based similarity measure between words in both sentences as will be seen shortly. Finally, the system performs post-processing in order to merge similar sentences into passages (p_q, p_x) such that $p_q \in d_q$, $p_x \in d_x$. Subsequent sections detail each stage.

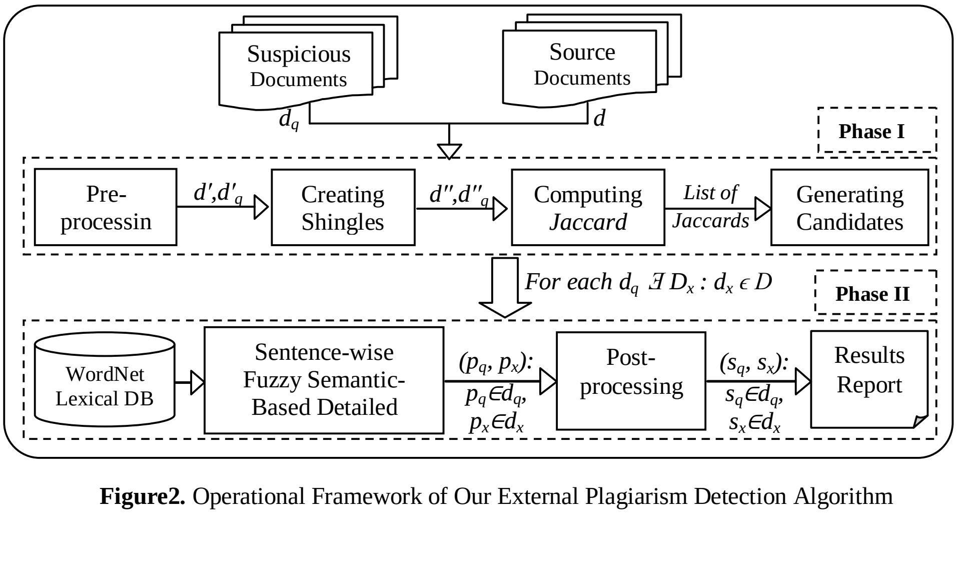


Figure2. Operational Framework of Our External Plagiarism Detection Algorithm

3.2 Phase I: Retrieval of Similar Documents

Near duplicate detection methods can be used to bring similar sources and discard dissimilar ones. We use shingling and Jaccard coefficient approach (Manning, Raghavan, & Schütze, 2008). The *k*-shingle (or word-*k*-gram) referred to a sequence of consecutive words of size *k*. The value for *k* is typically 3 or 4. Intuitively, two documents *A* and *B* are similar if they share enough *k*-shingles. By performing union and intersection operations between the *k*-shingles, we can find the Jaccard similarity coefficient between *A* and *B* as stated in equation (1).

$$J(A,B)=|\text{shingles of } A \cap \text{shingles of } B| \,/\, |\text{shingles of } A \cup \text{shingles of } B| \qquad (1)$$

Therefore for each suspicious document d_q, documents of Jaccard coefficient above a threshold value are taken to form the set of candidate documents D_x. We set the threshold of $Jaccard \geq 0.1$ because we found that this value derives about 1 to 30 candidate documents for each d_q. It was found that when we compare documents of Jaccard similarity less than 0.1, either none or about 1-2 plagiarised statements are detected. Also by using this method, we find that some suspicious documents do not have any candidates. That means that they might contain intrinsic plagiarism or do not contain plagiarism at all. More interesting, using this approach assumes the number of candidates for each suspicious document dynamic and may be small. That saves the computation time in contrast to having a fixed number of candidates for each suspicious document.

3.3 Phase II: In-Depth Detailed Analysis of (d_q , d_x) pairs

At this stage, a sentence-wise detailed analysis between each suspicious document d_q and its candidate document $d_x \in D_x$ is performed. At first, d_q and d_x are segmented into sentences S_q and S_x respectively using end-of-sentence delimiters. To obtain the degree of similarity between two sentences (s_q, s_x), a term-to-sentence correlation factor for each term w_q in s_q and the sentence s_x is computed as:

$$\mu_{q,x}= 1-\prod_{w_k \in s_x}(1 - F_{q,k}) \qquad (2)$$

where w_k are words in s_x and $F_{q,k}$ is a fuzzy similarity between w_q and w_k that we defined as follows:

$$F_{q,k} = \begin{cases} 1 & \text{if } w_k \text{ and } w_q \text{ are identical} \\ 0.5 & \text{if } w_k \text{ is in the synset of } w_q \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

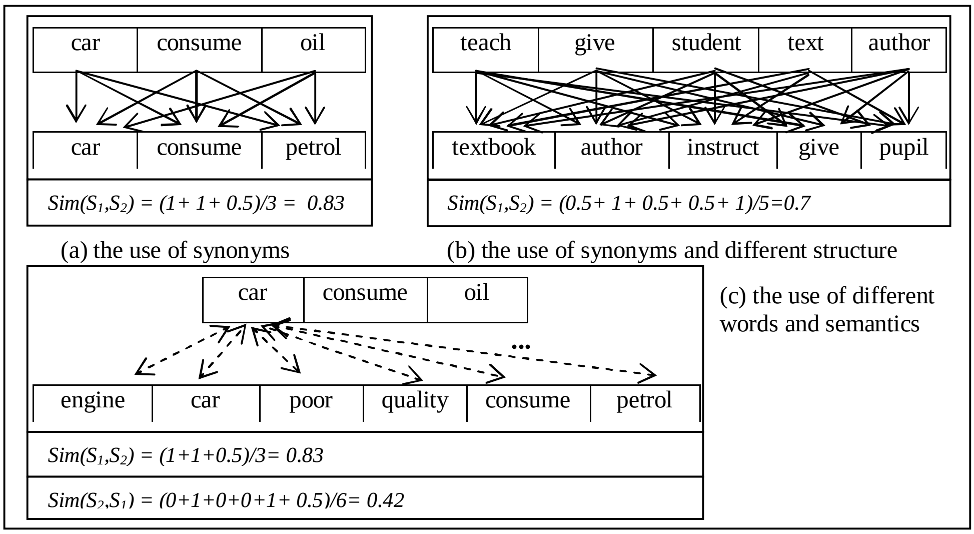
The synset of w_q is extracted by querying the WordNet lexical database (Miller, 1995). For example, the sentences S_1=“*this car consumes a lot of oil*” and S_2=“*this car consumes a lot of petrol*” are almost identical except the word *oil* replaced with *petrol.* Since the word *petrol* is found in the synonym set (synset) of *oil*, both sentences convey the same meaning and the degree of similarity between (s_q, s_x) is expressed as a fuzzy number between 0 and 1 using the equation

$$Sim(s_q, s_x) = (\mu_{1,x} + \mu_{2,x} + \ldots + \mu_{q,x} + \ldots + \mu_{n,x}) / n \qquad (4)$$

where n is the total number of words in s_q. Thus, we can calculate the degree of similarity between the S_1 and S_2 as shown in Figure 3 (a) taking into account that stop words were removed and non-stop words were stemmed. Another example is the sentences S_1=“*the teacher gives each student a text that he authored*” and S_2=“*a textbook authored by the instructor is given to his pupils*” where the later was paraphrased from the first. Since the following word pairs (*teach*, *instruct*), (*student*, *pupil*), (*text*, *textbook*) are synonyms, and the rest of words are identical, these sentences gain high similarity score of 0.7 as shown in Figure 3 (b). This indicates that sentences are semantically alike. It is noticeable that stemming the words can handle some means of obfuscation. Both of the previous examples have sentences with equal number of words. An example of a sentence pair with different lengths and semantics is S_1= “*this car consumes a lot of oil*” and S_2=“*the engine of this car is of poor quality and consumes a lot of petrol*”. In this case, $Sim(S_1,S_2) \neq Sim(S_2,S_1)$. Thus to judge two sentences as equal (i.e. plagiarised), the minimum similarity score should be above a threshold value (α >0.65) as in (5). According to this, the last pair shown in Figure 3 (c) is considered dissimilar because the minimum similarity is less than α.

$$EQ(s_q, s_x) = \begin{cases} 1 & \text{if } MIN(Sim(s_q, s_x), Sim(s_q, s_x)) \geq \alpha \\ 0 & otherwise \end{cases} \qquad (5)$$

Finally, the output of this algorithm is a list of sentence pairs (s_q, s_x): $s_q \in d_q$, $s_x \in d_x$, $d_x \in D$ marked as similar/plagiarised. Because of using sentences as comparison scheme, post-processing is required to merge subsequent sentences marked as plagiarised into passages. Also, we consider small distances under the predicate *less than or equal to 100 characters* to merge subsequent plagiarised sentences into passages pairs (p_q, p_x) : $p_q \in d_q$, $p_x \in d_x$, $d_x \in D_x$.


car
consume
oil
car
consume
petrol
Sim(S1,S2) = (1+ 1+ 0.5)/3 = 0.83
teach
give
student
text
author
textbook
author
instruct
give
pupil
Sim(S1,S2) = (0.5+ 1+ 0.5+ 0.5+ 1)/5=0.7
car
consume
oil
...
engine
car
poor
quality
consume
petrol
Sim(S1,S2) = (1+1+0.5)/3= 0.83
Sim(S2,S1) = (0+1+0+0+1+ 0.5)/6= 0.42

(a) the use of synonyms

(b) the use of synonyms and different structure

(c) the use of different words and semantics

Figure 3. Examples of different sentence pairs.

4 Experimental Setup

4.1 Instrumentation

Our algorithm has been built using C#.NET 2008. By using different libraries such as Linq, we perform sets operation to compute Jaccard similarity. We used a server with 4-core processors, 2.8 GHz. To utilise all cores, we have migrated our code to work on Visual Studio.NET 2010 which has introduced the concept of parallel computing[1].

4.2 Code Configuration

Below is a list of some parameters and settings that we have configured in our code.

- For pre-processing, stop words removal and porter stemmer (Porter, 1980) algorithms were used.
- For generating *k-shingles,* the k was set to 3 (i.e. *word-3-grams*).
- For computing Jaccard similarity and finding candidates, a threshold value of *Jaccard=0.1* was set to filter out non-candidate documents.
- For semantic-based analysis, WorldNet v3.0 using MySQL[2] was used to query the *Synset* table and extract synonyms of the words.
- For fuzzy similarity between two sentences, the equations (2)-(5) was employed, and the threshold in (5) was set to α= 0.65 which was found to be the most suitable based on our experimental trials.

5 Evaluation and Discussion on PAN’09 and PAN’10

In the PAN’09 extrinsic part, the recall was 0.3097 and the precision was 0.5424. In PAN’10, we submitted partial results of about 56.25% of the suspicious documents at first. Our full results on PAN’10 are presented in this paper (recall= 0.1259, precision= 0.5761, granularity= 3.5828). Our results are of both PAN’09 and PAN’10 are comparable as shown in Table 1. The results in PAN’10 showed that we detected about 12% of the plagiarism cases and about 57% of the detections were correct. The low recall might be for the reasons: (i) the algorithm was designed for extrinsic plagiarism task and did not tackle intrinsic nor cross-lingual plagiarism, (ii) we used stems instead of lemmas in pre-processing; however, WordNet needs lemmas which needs to be corrected in the future model, and (iii) the candidates compared were not enough to find more plagiarism cases. The precision of the algorithm shows that 57% of the detections were correct. Two words may be synonyms but with different senses and hence different meaning that make sentences not plagiarised which may lead to more false positives by our algorithm. Moreover, statements of short lengths might get a fuzzy similarity score of more than 0.65 easily; another reason for false positives. The ability of detecting each plagiarism case at once was bigger than 1

[1] http://channel9.msdn.com/learn/courses/VS2010/Parallel/

[2] http://wordnet.princeton.edu/wordnet/related-projects/#SQL

because the algorithm enabled the merging process of sentences if and only if they are subsequent or with few characters in between.

Table 1. Results of our fuzzy semantic-based approach in PAN’09 and PAN’10.

Dataset		***Recall***	***Precision***	***Granularity***	***Plag. Score***
PAN’09	PAN’09 Extrinsic Part	0.3097	0.5424	7.8867	0.1251
PAN’10	PAN’10 Extrinsic Part	0.1548	0.5758	3.5919	0.1109
	PAN’10 All (partial)	0.0464	0.3460	17.3057	0.0195
	PAN’10 All (complete)	0.1259	0.5761	3.5828	0.0941

Future Work

Our future work is to improve it for more detection efficiency and less time complexity. We will consider the following work: (i) using word-k-grams instead of sentences, (ii) using a Lemmatiser instead of the stemmer to get better results from WordNet, and (iii) modifying the post-processing stage to gain more ideal granularity.

Acknowledgements

We thank the organizers of PAN’10 evaluation lab, Martin Potthast in particular, for his generous support and instantaneous answer through the PAN’s mailing list.

References

Alzahrani, S. M., & Salim, N. (2009). *On the Use of Fuzzy Information Retrieval for Gauging Similarity of Arabic Documents*. Paper presented at the Second International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2009), London Metropolitan University, UK.

Koberstein, J., & Ng, Y.-K. (2006). Using Word Clusters to Detect Similar Web Documents. In *Knowledge Science, Engineering and Management* (pp. 215-228).

Li, Y., McLean, D., Bandar, Z. A., O'Shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering, 18*(8), 1138-1150.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Web search basics: Near-duplicates and shingling. In *Introduction to Information Retrieval* (pp. 437-442): Cambridge University Press.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM, 38*(11), 39-41.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130−137.

Yerra, R., & Ng, Y.-K. (2005). A Sentence-Based Copy Detection Approach for Web Documents. In *Fuzzy Systems and Knowledge Discovery* (pp. 557-570).