Using sentence similarity measure for plagiarism source retrieval

Notebook for PAN at CLEF 2014

Zubarev Denis[1] and Sochenkov Ilya[2]

[1] Institute for Systems Analysis of Russian Academy of Sciences, Moscow, Russia
[2] Peoples' Friendship University of Russia, Moscow, Russia
zubarev@isa.ru, isochenkov@sci.pfu.edu.ru

Abstract This paper describes a method that was implemented in the software submitted to PAN 2014 competition for the source retrieval task. For generating queries we use the most important noun phrases and words of sentences selected from given suspicious document. To download documents that are likely to be sources of plagiarism we employ sentence similarity measure.

1 Introduction

The plagiarism detection track on PAN [3] is divided in two subtasks: source retrieval and text alignment. Detailed information about these tasks is provided in [7]. In this paper we present a method that is supposed to solve the former task. Search engines (Indri, ChatNoir) are used to retrieve a candidate source for a suspicious document. We need to formulate set of queries based on a suspicious document for a search engine. Process of formulating queries is very important because it determines the maximum possible recall that can be achieved by a source retrieval method. One of goals of this paper is to explore an influence of phrasal search on recall. We need also to pay attention to filtering of incorrect source candidates to keep precision high enough. It allows to save computational resources during second task solution. We employ a sentence similarity measure for filtering source candidates. If a candidate contains a sentence that is quite similar to some suspicious sentence we consider such candidate as a source, otherwise the candidate is filtered. Another goal is to minimize amount of queries to maximum possible extent. To achieve this goal we select the most important sentences, from which queries are formed, and by active filtering queries based on downloaded sources.

The rest of this paper is organized as follows: Section 2 describes used sentence similarity measure. Section 3 provides the details of the source retrieval method. Section 4 presents the performance of the software in PAN 2014 competition. Section 5 concludes the paper.

2 Sentence similarity measure

Let us introduce the core method used for comparison of sentences. Given two arbitrary sentences s_e and s_t, denote as $N(s_e, s_t)$ a set of pairs of words with the same normal form, where the first element is taken from s_e and the second one from s_t. We compare two sentences by considering words from the set $N(s_e, s_t)$. For calculating overall similarity measure of two sentences we compute multiple similarities measures and then combine its values. Employed similarities are described below.

2.1 IDF overlap measure

Similar to [6] we define IDF overlap as follows:

$$I_1(s_e, s_t) = \sum_{(w_e, w_t) \in N(s_e, s_t)} \upsilon(w_e, s_e) \tag{1}$$

where $\upsilon(w_e, s_e)$ is IDF weight of word w_e in a sentence s_e. Also there holds an equation

$$\sum_{w \in s_e} \upsilon(w) = 1 \tag{2}$$

2.2 TF-IDF measure

Let us define TF-IDF measure in the following way:

$$I_2(s_e, s_t) = \sum_{(w_e, w_t) \in N(s_e, s_t)} f(w_e, w_t)\upsilon(w_e, s_e)TF_{w_t} \tag{3}$$

where $\upsilon(w_e, s_e)$ is IDF weight of the word $w_e \in s_e$, restriction (2) is kept; TF_{w_t} is TF weight of the word $w_t \in s_t$.

Additional restriction for TF_{w_t} is

$$0 \leq TF_{w_t} \leq 1. \tag{4}$$

$f(w_e, w_t)$ is a kind of a penalty for mismatch of w_e, w_t forms :

$$f(w_e, w_t) = \begin{cases} 1.0, \text{if words' forms are the same} \\ 0.8, \text{otherwise} \end{cases} \tag{5}$$

2.3 Sentence syntactic similarity measure

To be able to measure this kind of similarity we need to generate syntactic dependency tree from each sentence. We define $Syn(s_e)$ as a set that contains triplets (w_h, σ, w_d), where w_h, w_d are normalized head and dependent word respectively, σ is type of syntactic relation. Then we define syntactic similarity in the following way:

$$I_3(s_e, s_t) = \frac{\sum\limits_{(w_h, \sigma, w_d) \in (Syn(s_e) \cap Syn(s_t))} \upsilon(w_h, s_e)}{\sum\limits_{(w_h, \sigma, w_d) \in Syn(s_e)} \upsilon(w_h, s_e)} \tag{6}$$

Rationale for this measure is to treat sentences not as a bag-of-words but as syntactically linked text. Value of this measure will be low for sentences in which the same words are used but they are linked in a different way. This measure is quite similar to one, described in [5].

2.4 Overall sentence similarity

The overall sentence similarity we define as a linear combination of described measures.

$$Sim(s_e, s_t) = \sum_{i=1}^{3} k_i I_i(s_e, s_t), \tag{7}$$

where $k_i, i = 1, 2, 3$ determine relative contributions of each similarity.

Due to definitions $0 \leq I_i(s_e, s_t) \leq 1, i = 1, 2, 3$. Given additional restriction $\sum\limits_{i=1}^{3} k_i = 1$ we can conclude that

$$0 \leq Sim(s_e, s_t) \leq 1 \tag{8}$$

3 Source retrieval method

Source retrieval method consists of several steps:

1. suspicious document chunking
2. query formulation
3. download filtering
4. search control

These steps are identical to those ones described in [7]. Algorithm 1 shows a pseudocode of our source retrieval algorithm, which we describe in details in the sections below.

As was mentioned earlier, we use linguistic information for measuring sentence similarity. To obtain this information we use Freeling[1]. It performs document splitting, part-of-speech tagging, and dependency parsing. In our methods we use such information as words' forms, words' dependencies.

3.1 Suspicious document chunking

Firstly, the suspicious document is split into sentences. The length of a sentence is limited to 50 words. For each sentence its weight is calculated. To calculate weight of a sentence we sum weights of all words and phrases in the sentence. Weight of a word is obtained using TF-IDF. We calculated beforehand inverse document frequency for each word in a collection. The collection consisted of English Wikipedia's articles (about 400,000 documents). We used IDF-weights of words from this collection. Words that comprise a two-word noun phrases do not contribute in an overall sentence weight. A phrase weight is calculated as follows: $W_{phr} = 2(W_h + W_d)$, where W_h is the weight of a head word and W_d is the weight of a dependent one.

After calculating weight we select some sentences using different filters. Firstly, sentence weight must exceed 0.45 value, due to low weight points to insignificance of information. Other parameters are taken into consideration, such as the maximum and minimum amount of non-stop words (nouns, verbs, adjectives) per sentence (33

Algorithm 1 General overview of source retrieval method

```
function SOURCERETRIEVAL($doc$)
    $misuses \leftarrow \varnothing$
    $sentences \leftarrow$ SPLIT($doc$)
    for all $s \in sentences$ do
        $DownloadedDocs \leftarrow \varnothing$
        if $s \notin misuses$ then
            $query \leftarrow$ FORMQUERY($s$)
            $results \leftarrow$ SUBMITQUERY($query$)                ▷ snippets and urls are returned
            for all $r \in results$ do
                if FINDSIMILAR($sentences, r$) $\neq \varnothing$ then
                    $DownloadedDocs \leftarrow$ DOWNLOAD($r$)
                end if
            end for
            for all $d \in DownloadedDocs$ do
                $pairs \leftarrow$ FINDSIMILAR($sentences, d$)          ▷ pairs of similar sentences
                for all $(suspSent, downSent) \in pairs$ do
                    $misuses \leftarrow suspSent$
                end for
            end for
        end if
    end for
end function
```

and 11 respectively). Rationale for this is to exclude too short sentences that may have rather high similarity value only because of a large IDF overlap similarity. Very large sentences are typically either errors of splitting or large lists. It is not likely to fetch snippet that contains the whole large sentence, so we omit them. The maximum N sentences, which satisfied these criteria, are selected for further analysis. We call them suspicious sentences. Experiments showed that $N = 83$ gives the best result.

3.2 Query formulation

Before formulating queries we delete articles, pronouns, prepositions as well as duplicate words or phrases from each selected sentence. We use two available search engines Indri[10] and ChatNoir[8] for submitting queries. Sentences which contain phrases are submitted to Indri. For that we take 6 the most important (the most weighted) entities (phrases or words) from sentence. Phrases are wrapped up by special operator to leverage Indri phrasal search. If sentence does not contain phrases, 6 of the most weighted words are submitted to ChatNoir as a query. The formed queries are sequentially submitted to search engines. This scheme is similar to approach that was used by Elizalde[2] in PAN 2013. It was experimentally found that not using phrasal search dramatically decrease recall.

3.3 Download filtering

Snippets of top 7 results returned by a search engine are downloaded and preprocessed by means of Freeling. For each sentence extracted from snippets similarity to each suspicious sentence is calculated by means of the method described in section 2. If similarity between any suspicious sentence and a snippet sentence exceeds $MinSim$, then a document to which a snippet belongs is scheduled for downloading. Such document is considered to be a source document. Following parameters were found empirically for a comparison with snippet sentences: $MinSim = 0.5, WIdf = 0.4, WTfIdf = 0.4, WSynt = 0.2$, where '$W*$' parameters denote a relative contribution of IDF overlap similarity, TF-IDF similarity, and syntactic similarity, respectively, and $MinSim$ is a minimal similarity threshold.

3.4 Search control

To filter the rest of queries we use downloaded documents. After downloading source documents are subjected to preprocessing by Freeling. Then for every suspicious sentence similarity to sentences from downloaded documents is calculated. If there is a sentence similar to a suspicious one, the latter is marked as misuse. Misuses are not used in query formulation process, since their sources have already been found. This approach is similar to one used by Haggag[4]. For this round of comparisons the following parameters were used: $MinSim = 0.4, WIdf = 0.5, WTfIdf = 0.0, WSynt = 0.5$.

3.5 Tuning parameters

There are many tunable parameters in the described method. In general they were tuned separately. For this purpose we fixed parameters that are responsible for query generating (amount of words or phrases in query, document chunking parameters). All possible snippets and full document texts were fetched for queries that were formulated based on fixed parameters. Downloaded data were preprocessed by Freeling and prepared for loading by our software. Using such preprocessed data, multiple combinations of $MinSim$, $WIdf$, $WTfIdf$, $WSynt$ parameters were tried. In the end we chose a combination that gave the best F-measure. For the source retrieval evaluation ChatNoir oracle was used.

4 Results

Table 1 shows results of our software on the test data (about 100 documents of Webis-TRC-12 corpus [9]). Results were obtained by means of the evaluation platform TIRA[3].

As can be seen from Table 1, our software achieved F1 score of 0.45. It was the second highest achieved the F1 score by all participants.

An average of 37.03 queries were submitted per suspicious document and 18.62 results downloaded. The amount of both queries and downloaded results were relatively low in comparison with the other softwares particapated in PAN. Only 37 sentences

Table 1. PAN 2014 Source retrieval final results

Submission	**Retrieval Performance**			**Workload**		**Time to 1st Detection**		**No Detection**
	F_1	Precision	Recall	Queries	Downloads	Queries	Downloads	
zubarev14	0.45	0.54	0.45	37.03	18.62	5.4	**2.25**	3
suchomel14	0.11	0.08	0.40	**19.05**	237.3	**3.1**	38.6	**2**
williams14	**0.47**	**0.57**	0.48	117.13	**14.41**	18.82	2.34	4
kong14	0.12	0.08	0.48	83.5	207.01	85.7	24.9	6
prakash14	0.39	0.38	**0.51**	59.96	38.77	8.09	3.76	7
elizalde14	0.34	0.40	0.39	54.5	33.2	16.4	3.9	7

were transformed into queries from 83 selected sentences on average. Such heavy filtering was crucial for achieving relatively high precision. It was experimentally found that the query filtering decreased recall but on the other hand significantly increased precision. According to results our software downloaded $18.62 * 0.54 = 10.05$ true positives for one suspicious document on average. This means that at least 3.7 queries are required for retrieving true positives results. This result proved that using phrases for candidate retrieval is quite reasonable and effective. However, the amount of queries to first detection is 5.4. It seems that sorting sentences by its weight is not the best strategy. Nevertheless, indicators that are measuring time to first detection are also low in comparison with the other participants results. On average, 2.25 full texts were downloaded until the first correct source document. It suggests that the snippets filtering based on employing sentence similarity measure worked relatively well.

No plagiarism sources were detected for 3 suspicious documents, which was about 3% of the suspicious documents in the test corpus. This result shows that the software is able to retrieve sources of plagiarism for the majority of documents.

5 Conclusions

This paper describes a software that can achieve relatively high retrieval performance while minimizing the workload. It is possible due to employing two approaches, namely the phrasal search for the candidates retrieving and using sentence similarity measure for the candidates filtering.

Many search engines support phrasal search to some extent. Some of them (e.g. Yandex, Yahoo) provide proximity search which makes it possible to search phrases, and the other search engine (e.g. Google) provide exact phrase search.

Our core filtering method is based on sentences comparison. This method works well when snippet is an original sentence (or some part of it). If snippets contains heavy overlapped fragments of one sentence divided by a delimiter, then it is hardly useful to employ sentence comparison. We experienced occasionally this problem during experiments with ChatNoir snippets. But we believe that snippets that are provided by popular search engines (e.g. Google, Yandex) contain original sentences. Therefore results of our method are supposed to be reproducible in real-world environment.

However, there is some room for further improvements. It is probably worth reducing the set of phrases only to phrasemes (e.g. collocations that made up great deal of the phraseme inventory). The rationale for this is based on an assumption that it is very easy to change a phrase using synonym of a head or a dependent word. But one cannot simply change any word in phraseme because the phrase will lose its meaning. So one needs to synonymize the whole phrase or to leave it as it is.

Acknowledgments

We gratefully acknowledge support by Russian Foundation for Basic Research under grant No. 14-07-31149 and the PAN Lab organizers for their effort in organizing PAN 2014.

References

1. Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., Padró, M.: Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In: Proceedings of LREC. vol. 6, pp. 48–55 (2006)
2. Elizalde, V.: Using statistic and semantic analysis to detect plagiarism. In: CLEF (Online Working Notes/Labs/Workshop) (2013)
3. Gollub, T., Potthast, M., Beyer, A., Busse, M., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Recent trends in digital text forensics and its evaluation. In: Information Access Evaluation. Multilinguality, Multimodality, and Visualization, pp. 282–302. Springer (2013)
4. Haggag, O., El-Beltagy, S.: Plagiarism candidate retrieval using selective query formulation and discriminative query scoring. In: CLEF (Online Working Notes/Labs/Workshop) (2013)
5. Liu, D., Liu, Z., Dong, Q.: A dependency grammar and wordnet based sentence similarity measure. Journal of Computational Information Systems 8(3), 1027–1035 (2012)
6. Metzler, D., Bernstein, Y., Croft, W.B., Moffat, A., Zobel, J.: Similarity measures for tracking information flow. In: Proceedings of the 14th ACM international conference on Information and knowledge management. pp. 517–524. ACM (2005)
7. Potthast, M., Gollub, T., Hagen, M., Tippmann, M., Kiesel, J., Rosso, P., Stamatatos, E., Stein, B.: Overview of the 5th International Competition on Plagiarism Detection. In: Forner, P., Navigli, R., Tufis, D. (eds.) Working Notes Papers of the CLEF 2013 Evaluation Labs (Sep 2013), http://www.clef-initiative.eu/publication/working-notes
8. Potthast, M., Hagen, M., Stein, B., Graßegger, J., Michel, M., Tippmann, M., Welsch, C.: Chatnoir: a search engine for the clueweb09 corpus. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. pp. 1004–1004. ACM (2012)
9. Potthast, M., Hagen, M., Völske, M., Stein, B.: Crowdsourcing Interaction Logs to Understand Text Reuse from the Web. In: Fung, P., Poesio, M. (eds.) Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 13). pp. 1212–1221. ACL (Aug 2013), http://www.aclweb.org/anthology/P13-1119
10. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language model-based search engine for complex queries. In: Proceedings of the International Conference on Intelligent Analysis. vol. 2, pp. 2–6. Citeseer (2005)