Diverse Queries and Feature Type Selection for Plagiarism Discovery

Notebook for PAN at CLEF 2013

Šimon Suchomel, Jan Kasprzak, and Michal Brandejs

Faculty of Informatics, Masaryk University
{suchomel,kas,brandejs}@fi.muni.cz

Abstract This paper describes approaches used for the Plagiarism Detection task in PAN 2013 international competition on uncovering plagiarism, authorship, and social software misuse. We present modified three-way search methodology for Source Retrieval subtask and analyse snippet similarity performance. The results show, that presented approach is adaptable in real-world plagiarism situations. For the Detailed Comparison task, we discuss feature type selection and global postprocessing. Resulting performance is significantly better with the described modifications, and further improvement is still possible.

1 Introduction

In PAN 2013[1] competition on plagiarism detection we participated in both the Source Retrieval and the Text Alignment subtasks. In both tasks we adapted methodology used in PAN 2012[2] [11]. Section 2 describes querying approach for source retrieval, where we used three different types of queries. We present a new type of query based on text paragraphs. The query execution was controlled by its type and by preliminary similarities discovered during the searches. Section 3 describes our approach for the text alignment (pairwise comparison) subtask. We briefly introduce our system, and then we discuss the feature types, which are usable for pairwise comparison, including the evaluation of their feasibility for this purpose. We then describe the global (corpus-wide) optimizations used, and finally we discuss the results achieved and further development.

2 Source Retrieval

The source retrieval is a subtask in a plagiarism detection process during which only a relatively small subset of documents are retrieved from the large corpus. Those candidate documents are usually further compared in detail with the suspicious document. In PAN 2013 source retrieval subtask the main goal was to identify web pages which have been used as a source of plagiarism for test corpus creation.

The test corpus contained 58 documents each discussing one topic only. Those documents were created intentionally by semiprofessional writers, thus they featured nearly

[1] See [6] for an overview of PAN 2013 plagiarism detection campaign.

[2] See [5] for an overview of PAN 2012 plagiarism detection campaign.

realistic plagiarism cases [8]. Resources were looked up in the ClueWeb[3] corpus. Such conditions are similar to a realistic plagiarism detection scenario.The main difference between real-world corpus of suspicious documents such as for example corpus created from theses stored in the Information System of Masaryk University and the corpus of suspicious documents used during the PAN 2013 competition is that in the PAN corpus each document contains plagiarized passages. Therefore we can expected existence of a plagiarized passage and deepen the search during the process in certain parts of the document where no similar passage has yet been found.

An online plagiarism detection can be viewed as a reverse engineering task where we need to find original documents from which the plagiarized document was created. During the process the plagiarist locates original documents with the use of a search engine. The user decides what query the search engine to ask and which of the results from the result page to use.

The same methodology – utilizing a search engine; is used for source retrieval. This approach is based on the fact that we do not possess enough resources to download and effectively process the whole corpus. In the case of PAN 2013 competition we utilized the ChatNoir [7] search engine which indexes the English subset of the ClueWeb.

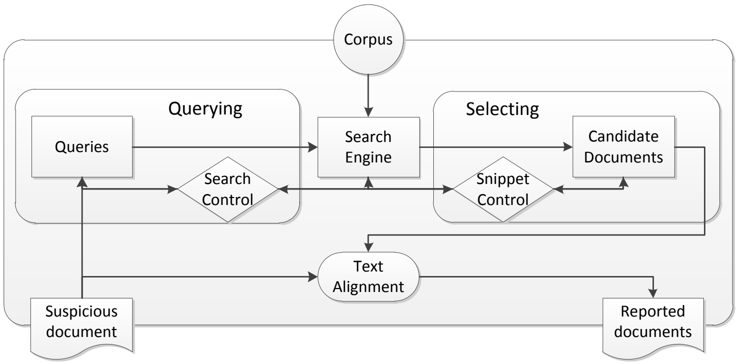

Figure 1. Source retrieval process.

The reverse engineering decision process resides in creation of suitable queries on the basis of the suspicious document and in decision what to actually download and what to report as a plagiarism case from the search results.

These first two stages are shown in Figure 1 as Querying and Selecting. Selected results from the search engine are then textually aligned with the suspicious document (see section 3 for more details). If there is any continuous passage of reused text detected, the result document is reported and the continuous passages in the suspicious document are marked as *discovered* and no further processing of those parts is done.

[3] http://lemurproject.org/clueweb09.php/

2.1 Querying

Querying means to effectively utilize the search engine in order to retrieve as many relevant documents as possible with the minimum amount of queries. We consider the resulting document relevant if it shares some of text characteristics with the suspicious document. In real-world queries as such represent appreciable cost, therefore their minimization should be one of the top priorities.

We used 3 different types of queries[4]: i) keywords based queries, ii) intrinsic plagiarism based queries, and iii) paragraph based queries. Three main properties distinguish each type of query: i) Positional; ii) Phrasal; iii) Deterministic. Positional queries carry extra information about a textual interval in the suspicious document which the query represents. A phrasal query aims for retrieval of documents containing the same small piece of text. They are usually created from closely coupled words. Deterministic queries for specific suspicious document are always the same no matter how many times we run the software.

Keywords Based Queries. The keywords based queries are composed of automatically extracted keywords from the whole suspicious document. Their purpose is to retrieve documents concerning the same theme. As a method for automated keywords extraction, we used a frequency based approach described in [11]. The method combines term frequency analysis with TF-IDF score. As a reference corpus we used English web corpus [1] crawled by SpiderLink [12] in 2012 which contains 4.65 billion tokens.

Each keywords based query was constructed from five top ranked keywords consecutively. Each keyword was used only in one query. In order to direct the search more at the highest ranked keywords we also extracted their most frequent two and three term long collocations. These were combined also into queries of 5 words. Resulting the 4 top ranked keywords alone can appear in two different queries, one from the keywords alone and one from the collocations.

The keywords based queries are non-positional, since they represent the whole document. They are also non-phrasal since they are constructed of tokens gathered from different parts of the text. And they are deterministic; for certain input document the extractor always returns the same keywords.

Intrinsic Plagiarism Based Queries. The second type of queries purpose to retrieve pages which contain text detected as different, in a manner of writing style, from other parts of the suspicious document. For this purpose we implemented vocabulary richness method [2] together with sliding windows concept for text chunking as described in [11].

A representative sentence longer than 6 words was randomly selected among those that apply from the suspicious part of the document. The query was created from the representative sentence leaving out stop words. The intrinsic plagiarism based queries are positional. They carry the position of the representative sentence.They are phrasal,

[4] We used similar three-way based methodology in PAN 2012 Candidate Document Retrieval subtask. However, this time we completely replaced the headers based queries with paragraph based queries, since the headers based queries did not pay off in the overall process.

since they represent a search for a specific sentence. And they are nondeterministic, because the representative sentence is selected randomly.

Paragraph Based Queries. The purpose of paragraph based queries is to check some parts of the text in more depth. Those are parts for which no similarity has been found during previous searches. For this case we considered a paragraph as a minimum text chunk for plagiarism to occur. Despite the fact, that paragraphs differ in length, we represent one paragraph by only one query.

From each paragraph we extracted the longest sentence from which the query was constructed. Ideally the extracted sentence should carry the highest information gain. The query was maximally 10 words in length which is the upper bound of ChatNoir and was constructed from the selected sentence by omitting stop words.

2.2 Search Control

For each suspicious document we prepared all three types of queries during the first phase at once. Queries were executed stepwise. After processing each query the results were evaluated. From all textual similarities between each result and the suspicious document, the document intervals of those similarities were marked as *discovered*.

Firstly, there were always all of the keywords based queries executed. Secondly the intrinsic plagiarism based queries were executed according to their creation sequence. During the execution, if any of the query position intersected with any of the *discovered* interval, the query was dropped out. Analogically, the last there were paragraph based queries processed.

This search control results in two major properties. Firstly, the source retrieval effort was increased in parts of the suspicious document, where there has not yet been found any textual similarity. And secondly, after detection a similarity for a certain part of the text, no more intentionally retrieval attempts for that part were effectuated. Meaning that all discovered search engine results were evaluated, but there were executed no more queries regarding that passage.

2.3 Result Selection

The second main decisive area about source retrieval task is to decide which from the search engine results to download. This process is represented in Figure 1 as Selecting.

The ChatNoir offers snippets for discovered documents. The snippet generation is considered costless operation. The snippet purpose is to have a quick glance at a small extract of resulting page. The extract is maximally 500 characters long and it is a portion of the document around given keywords. On the basis of snippet, we needed to decide whether to actually download the result or not.

2.4 Snippet Control

Since the snippet is relatively small and it can be discontinuous part of the text, the text alignment methods described in section 3 were insufficient in decision making over

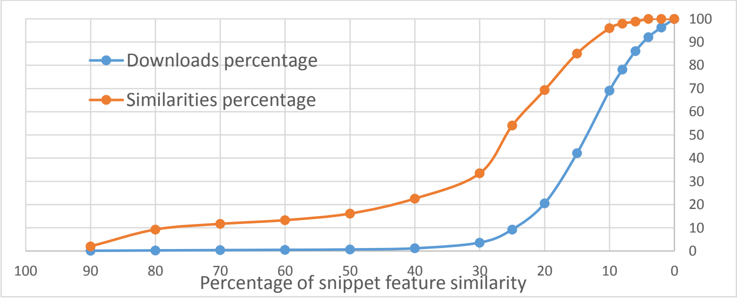

Figure 2. Downloads and similarities performance.

document download. Therefore we chose to compare existence of snippet word tuples in the suspicious document.

We used 2-tuples measurement, which indicates how many neighbouring word pairs coexist in the snippet and in the suspicious document. We decided according to this value whether to download the source or not. For the deduction of the threshold value we used 4413 search results from various queries according to documents in the training corpus. Each resulting document was textually aligned to its corresponding suspicious document. In this way we calculated 248 similarities in total after downloading all of the 4431 documents.

The 2-tuples similarity performance is depicted in Figure 2. Horizontal axis represents threshold of the 2-tuples similarity percentage between the snippet and the suspicious document. The graph curves represent obtained resource percentage according to the snippet similarity threshold. A profitable threshold is the one with the largest distance between those two curves. We chose threshold of the snippet similarity to 20%, which in the graph corresponds to 20% of all downloads and simultaneously with 70% discovered similarities.

2.5 Source Retrieval Results

In PAN 2013 Source Retrieval subtask we competed with other 8 teams. There can not be selected the best approach because there were several independent performance measures. Possibly each approach has its pros and cons and many approaches are usable in different situations.

We believe that in the realistic plagiarism detection the most important is to keep the number of queries low and simultaneously maximize recall. It is also advisable to keep the number of downloads low, but on the other hand, it is relatively cheep and easily scalable operation.

Our approach had the second best ratio of recall to the number of used queries, which tells about the query efficacy. The approach with the best ratio used few queries (4.9 queries per document which was 0.4 of the amount we used), but also obtained the lowest recall (0.65 of our recall). The approach with the highest recall (and also lowest precision) achieved 2.8 times higher recall with 3.9 times more queries compared to ours.

Our approach achieved also low precision, which means we reported many more results and they were not considered as correct hits. On the other hand each reported result contained some textual similarity according to text alignment subtask score, which we believe is still worthwhile to report.

3 Text Alignment

Our approach at the text alignment subtask of PAN 2013 uses the same basic principles as our previous work in this area, described in [11], which in turn builds on our work for previous PAN campaigns [3], [4]:

We detect *common features* between source and suspicious documents, where the features we currently use are word n-grams, and stop-word m-grams [10]. From those common features (each of which can occur multiple times in both source and suspicious document), we form *valid intervals*[5] of characters from the source and suspicious documents, where the interval in both of these documents is covered “densely enough” by the common features.

We then postprocess the valid intervals, removing the overlapping detections, and merging the detections which are close enough to each other.

For the training corpus, our unmodified software from PAN 2012 gave the following results[6]:

$\textit{plagdet} = 0.7235, \textit{recall} = 0.6306, \textit{precision} = 0.8484, \textit{granularity} = 1.0000$

We take the above as the baseline for further improvements. In the next sections, we summarize the modifications we did for PAN 2013.

3.1 Alternative Features

In PAN 2012, we used word 5-grams and stop-word 8-grams. This year we experimented with different word n-grams, and also with contextual n-grams as described in [13]. Modifying the algorithm to use contextual n-grams created as word 5-grams with the middle word removed (i.e. two words before and two words after the context) yielded better score:

$\textit{plagdet} = 0.7421, \textit{recall} = 0.6721, \textit{precision} = 0.8282, \textit{granularity} = 1.0000$

We then made tests with plain word 4-grams, and to our surprise, it gave even better score than contextual n-grams:

$\textit{plagdet} = 0.7447, \textit{recall} = 0.7556, \textit{precision} = 0.7340, \textit{granularity} = 1.0000$

It should be noted that these two quite similar approaches (both use the features formed from four words), while having a similar plagdet score, have their precision and recall values completely different. Looking at the training corpus parts, plain word 4-grams were better at all parts of the corpus (in terms of plagdet score), except the 02-no-obfuscation part.

In our final submission, we used word 4-grams and stop-word 8-grams.

[5] See [4] for the algorithm for computing valid intervals; a similar approach is also used in [10].

[6] See [9] for definition of *plagdet* and the rationale behind this type of scoring.

3.2 Global Postprocessing

For PAN 2013, the algorithm had access to all of the source and suspicious documents at once. It was not limited to a single document pair, as it was in 2012. We have rewritten our software to handle all of the documents in one run, in order to be able to do cross-document optimizations and postprocessing, similar to what we did for PAN 2010.

For PAN 2010, we used the following postprocessing heuristics: If there are overlapping detections inside a suspicious document, keep the longer one, provided that it is long enough. For overlapping detections up to 600 characters, drop them both. We implemented this heuristics, but found that it led to a lower score than without this modification. Further experiments with global postprocessing of overlaps led to a new heuristics: we unconditionally drop overlapping detections with up to 250 characters both, but if at least one of them is longer, we keep both detections. This is probably a result of plagdet being skewed too much towards recall (because the percentage of plagiarized cases in the corpus is way too high compared to real-world), so it is favourable to keep the detection even though the evidence for it is rather low.

The global postprocessing improved the score even more:
$\textit{plagdet} = 0.7469, \textit{recall} = 0.7558, \textit{precision} = 0.7382, \textit{granularity} = 1.0000$

3.3 Evaluation Results and Future Work

The evaluation on the competition corpus had the following results:
$\textit{plagdet} = 0.7448, \textit{recall} = 0.7659, \textit{precision} = 0.7251, \textit{granularity} = 1.0003$

This is quite similar to what we have seen on a training corpus, with only the granularity different from 1.000 being a bit surprising. Compared to the other participants, our algorithm performs especially well for human-created plagiarism (the 05-summary-obfuscation sub-corpus), which is where we want to focus for our production systems[7].

We plan to experiment further with combining more than two types of features, be it continuous n-grams or contextual features. This should allow us to tune down the aggressive heuristics for joining neighbouring detections, which should lead to higher precision, hopefully without sacrificing recall.

As for the computational performance, it should be noted that our software is prototyped in a scripting language (Perl), so it is not the fastest possible implementation of the algorithm used. The code contains about 800 non-comment lines of code, including the parallelization of most parts and debugging/logging statements.

The system is mostly language independent. The only language dependent part of the code is the list of English stop-words for stop-word n-grams. We use no stemming or other kinds of language-dependent processing.

4 Conclusions

We introduced querying strategy with snippet similarity measure. In source retrieval subtask the strategy performed with the second best ratio of recall to the number of

[7] Our production systems include the Czech National Archive of Graduate Theses, http://theses.cz

used queries. We focused our queries on selected parts of text and on parts with no discovered external similarities. Unfortunately the ChatNoir search engine currently does not support phrasal search, therefore it is possible that evaluated results may be quite distorted in this manner.

In the text alignment subtask, we have achieved a significant improvement with respect to our system from PAN 2012. Further development in this area is still possible. For a real-world system, however, a completely different set of parameters and heuristics needs to be used, as a result of plagdet score together with the structure of the competition corpus being too different from the real world.

References

1. Sketch Engine EnTenTen Corpus. http://trac.sketchengine.co.uk/wiki/Corpora/enTenTen (2012)
2. Eissen, S.M.Z., Stein, B.: Intrinsic plagiarism detection. In: Proceedings of the European Conference on Information Retrieval (ECIR-06) (2006)
3. Kasprzak, J., Brandejs, M.: Improving the reliability of the plagiarism detection system. In: Notebook Papers of CLEF 2010 LABs and Workshops. Citeseer (2010)
4. Kasprzak, J., Brandejs, M., Křipač, M.: Finding plagiarism by evaluating document similarities. In: SEPLN'09: The 25th edition of the Annual Conference of the Spanish Society for Natural Language Processing (2009)
5. Potthast, M., Gollub, T., Hagen, M., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeño, A., Gupta, P., Rosso, P., Stein, B.: Overview of the 4th international competition on plagiarism detection. In: CLEF 2012 Evaluation Labs and Workshop (2012)
6. Potthast, M., Gollub, T., Hagen, M., Tippmann, M., Kiesel, J., Stamatatos, E., Rosso, P., Stein, B.: Overview of the 5th international competition on plagiarism detection. In: CLEF 2013 Evaluation Labs and Workshop (September 2013)
7. Potthast, M., Hagen, M., Stein, B., Graßegger, J., Michel, M., Tippmann, M., Welsch, C.: ChatNoir: A Search Engine for the ClueWeb09 Corpus. In: Hersh, B., Callan, J., Maarek, Y., Sanderson, M. (eds.) 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12). p. 1004. ACM (Aug 2012)
8. Potthast, M., Hagen, M., Völske, M., Stein, B.: Crowdsourcing interaction logs to understand text reuse from the web. In: 51st Annual Meeting of the Association of Computational Linguistics (ACL 13) – (to appear). ACM (Aug 2013)
9. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An Evaluation Framework for Plagiarism Detection. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010). Association for Computational Linguistics, Beijing, China (Aug 2010)
10. Stamatatos, E.: Plagiarism detection using stopword n-grams. Journal of the American Society for Information Science and Technology (2011)
11. Suchomel, Š., Kasprzak, J., Brandejs, M.: Three way search engine queries with multi-feature document comparison for plagiarism detection. In: Forner, P., Karlgren, J., Womser-Hacker, C. (eds.) CLEF (Online Working Notes/Labs/Workshop). pp. 1–8 (2012)
12. Suchomel, V., Pomikálek, J.: Efficient web crawling for large text corpora. In: Kilgarriff, A., Sharoff, S. (eds.) Proceedings of the seventh Web as Corpus Workshop (WAC7). pp. 39–43 (2012)
13. Torrejón, D.A.R., Ramos, J.M.M.: Detailed comparison module in coremo 1.9 plagiarism detector. In: CLEF (Online Working Notes/Labs/Workshop). pp. 1–8 (2012)