Similarity Overlap Metric and Greedy String Tiling at PAN 2012: Plagiarism Detection

Notebook for PAN at CLEF 2012

Arun kumar Jayapal

The University of Sheffield, UK
arunkumar.jeyapal@gmail.com

Abstract This paper reports the best performed approach followed for the candidate document retrieval task and the approach used for the detailed comparison task of the Plagiarism detection track in PAN 2012. The aim of the participation was to understand a few of the computer-assisted approaches used for plagiarism detection. The plagiarism detection is dependent on two broad tasks, (1) the candidate document retrieval task and (2) the detailed comparison task. The N-gram similarity overlap metric was used for candidate document retrieval task and the greedy string tiling algorithm for detailed comparison task. The evaluation results suggested that the approach used for the candidate document retrieval task was highly competitive, but the approach used for detailed comparison task need much more improvement.

Index Terms: plagiarism detection, candidate document retrieval, detailed comparison, similarity overlap, greedy string tiling

1 Introduction

Plagiarism may be referred to as idea(s) taken from or words copied or replicated from any given source document without referring to the original work or the source [6]. As the number of online resources is increasing and chances of the risk of plagiarism is high, there is a need to detect plagiarism from online resources. There are a number of automatic plagiarism detection tools currently in existence to prevent the unauthorized use of others work/ideas. But there is a need to identify the right set of documents from the online resources and the right sections of text which are plagiarized at a good detection rate. The candidate document retrieval task and detailed comparison task were conducted as a part of Plagiarism detection track in PAN 2012. This paper reports the best performed approach followed for these tasks in PAN 2012.

2 Candidate Document Retrieval

2.1 Query formulation:

A set of suspicious documents were provided for the candidate document retrieval task which had text with paragraphs. The text from each suspicious document were split

into sentences using the OpenNLP's [3] maximum entropy classifier [4] and the model trained on opennlp training data for english [3]. To obtain the most useful information from the provided data, it was identified from experimentation that a minimum of four lines together had useful information to search. The useful information represents the context of the paragraph, which were mostly dependent on the pronouns, nouns and verbs. Further details on extracting the useful information is discussed further.

Upto four sentences were considered together for formulating the query. The query included upto 10 words accepted by ChatNoir search engine through the ChatNoir API [2]. Each sentence was tagged using OpenNlp's [3] pre-built model, trained on the Penn Tree Bank [5] dataset. After each sentence was pos tagged, stopwords were removed from the sentences. Also, the words with tags other than a noun, pronoun, verb and adjective were removed and the remaining unique useful words were used as search terms for the chatnoir search engine. When the number of search terms were above 10 per paragraph, only the first 10 unique words were considered for the search since the chatnoir search API had a limit of upto 10 search terms per query. The search terms were then formulated as a query and provided as input the ChatNoir API. The search engine returned search results in JSON format and they were logged for further analysis.

2.2 Search results analysis:

The ChatNoir [2] search engine, based on the query terms, returned the search results in JSON format. For each of the provided suspicious documents, N-JSON data were returned for N-paragraphs (a paragraph is a set of four sentences) identified from the suspicious document. Each of the JSON data had the unique identifiers of the relevant documents returned as search results. Also, when the query terms did not match any criteria in the search index, no results were reported. The logged search results were parsed using JSON-simple parser [1], and the ChatNoir API [2] was used to download the content of the search results as well. Also, repetitive search results were not downloaded to ensure that the bandwidth and the server load is reduced. Each of the relevant documents downloaded were compared with the suspicious document and a similarity overlap metric was computed. The similarity overlap metric as provided in [7], given below was identified as a suitable metric to determine the relevant documents.

$$Sim_{(overlap)}(suspDoc, srcDoc) = \frac{intersect(set(suspDoc,Ngram),set(srcDoc,Ngram))}{min(set(suspDoc,Ngram),set(srcDoc,Ngram))}$$

To compute the similarity score between the suspicious document and the source document, each document (i.e., the suspicious document and the source document) was split into n-grams. For the PAN, the number of grams was set to 5. After splitting the text of the souce document and the suspicious documents into 5-grams, the unique set of 5-grams from each document were passed through an intersection function. The intersection function produced the number of common 5-grams between the source document and the suspicious document. The intersection value was then divided by the mininum number of unique 5-grams identified. As provided in [7], the overlap score always lies between 0 and 1, where 0 means there is no similarity between the suspicious document and the source document and 1 means that the suspicious document

and the source document are similar to each other. Based on the similarity score obtained for each source document against the suspicious document, the documents were sorted in descending order and the most relevant documents were identified as potentially plagiarized documents. The PAN 2012 evaluation results with *precision 0.6582* & *recall 0.2775* suggested that, the said approach is one of the best approaches used for identifying the candidate documents.

3 Detailed Comparison

3.1 Algorithm used:

The basic algorithm used to detect the plagiarized sections of the text for the detailed comparison task was Greedy String Tiling (GST) algorithm. The GST algorithm identifies the longest plagiarized sequence of substrings from the text of the source document and returns the sequence as tiles (i.e., the sequence of substrings) from the source document and the suspicious document. The GST algorithm was implemented based on running Karp-Rabin matching [9].

3.2 System:

Given the training data [8], the idea was to initially classify each of the document pair into one among the classes of no-plagiarism, no-obfuscation, artificial-low, artificial-high, translation and simulated-paraphrase and to set the minimum tile length accordingly. The idea behind the initial classification was to improve the precision. A short experiment was conducted to test how GST works on the different training data. During the experiment, it was noticed that GST with minimum tile length of 10 did not identify all the plagiarized sections of the documents. Therefore the minimum tile length was varied based on the classifier output. For classification purposes, the training data [8] provided for the detailed comparison task was used. Weka's [10] Random Subspace Classifier was chosen for the training and classifying the document pair into one among the said classes. Based on short experimentation on different classifiers using weka GUI experimenter, the Random Subspace Classifier was chosen based on the 10 fold cross validation results. To train the classifier, the following features were used:

1. Suspicious unigram count
2. Source unigram count
3. Count of unigrams found in both suspicious and source text
4. Number of suspicious n-grams
5. Number of source n-grams
6. Similarity score computed between the source and suspicious documents
7. Intersection score computed between the source and suspicious documents
8. Number of tiles identified between the document pair
9. Is source document translated

The entire training set which included one thousand document pairs for each class was used for training the Random subspace classifier. The 10-fold cross-validation across

the training set produced 0.798 precision, 0.79 recall and 0.791 F-measure. Although there was over-fitting, the classifier was not optimized due to time constraints. The output of the classifier was also used to reduce the computation time especially when no-plagiarism pairs were provided for the detailed comparison task. Also when a document pair is classified as translated, bing translator api was used to identify the provided source document's language and to translate the identified language to english. The translated document is then passed through string tiling algorithm. Moreover, based on the document pair identified by the classifier, the minimum tile length was varied. The minimum tile length was set for each classification based on experimentation with different settings. The minimum tile length set for each class is summarized in the table 1.

Table 1. Classification and Minimum Tile Length

Plagiarism Class	Tile Length
no-plagiarism	0
no-obfuscation	15
artificial-low	7
artificial-high	5
translation	3
simulated-paraphrase	10

The PAN evaluation results with scores *plagDet 0.0452519, precision 0.6229785, recall 0.0758258* and *granularity 6.9317042*, suggested that the system need to be optimized on further research and experimentation with different settings.

4 Conclusions

In this paper, the best performing approach for candidate document retrieval and the GST algorithm used for detailed comparison task in PAN 2012 is described. While the candidate document retrieval approach followed state of the art similarity overlap metric for identifying overlap score between the suspicious document and the retrieved set of documents, a relatively different approach was used to identify the query terms from the suspicious document for search. Though, GST algorithm being used as the state of the art algorithm to detect plagiarism, the system's performance was not as expected. At the time of development few parameters were tested due to time constraints. In the next opportunity, further enhancements to the system will be incorporated on further research. It was a very good learning experience for the participation of PAN 2012.

References

1. Json-simple. Online (January 2009), http://code.google.com/p/json-simple/w/list
2. Grassegger, J., Hagen, M., Michel, M., Potthast, M., Stein, B., Tippmann, M., Welsch, C.: Chatnoir search engine. Online (2009), http://webis15.medien.uni-weimar.de/
3. Kottmann, J., Margulies, B., Ingersoll, G., Drost, I., Kosin, J., Baldridge, J., Goetz, T., Morton, T., Silva, W., Autayeu, A., Galitsky, B.: Apache opennlp. Online (May 2011), http://opennlp.apache.org
4. Manning, C.D., Schutze, H.: Foundations of statistical natural language processing. MIT Press (1999)
5. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: the penn treebank. Computational Linguistics 19 (1992), ftp://ftp.cis.upenn.edu/pub/treebank/doc/cl93.ps.gz
6. Maurer, H., Kappe, F., Zaka, B.: Plagiarism - a survey. Journal of Universal Computer Science 12 no. 8, 1050 – 1084 (2006)
7. Nawab, R.M.A., Stevenson, M., Clough, P.: University of sheffield lab report for pan at clef 2010 (2010)
8. Potthast, M., Stein, B., Hagen, M., Gollub, T., BarrÃşn-CedeÃśo, A., Rosso, P., Gupta, P.: Pan12 detailed comparison training corpus. http://pan.webis.de/ (March 2012)
9. Wise, M.J.: String similarity via greedy string tiling and running karp-rabin matching. Online (December 1993)
10. Witten, I.H., Frank, E., Trigg, L., Hall, M., Holmes, G., Cunningham, S.J.: Weka: Practical machine learning tools and techniques with java implementations (1999)