Supervised Ranking for Plagiarism Source Retrieval
Notebook for PAN at CLEF 2013

Kyle Williams[†], Hung-Hsuan Chen[‡], and C. Lee Giles[†,‡]

[†]Information Sciences and Technology
[‡]Computer Science and Engineering
Pennsylvania State University
University Park, PA, 16802, USA
kwilliams@psu.edu, hhchen@psu.edu, giles@ist.psu.edu

Abstract Source retrieval involves making use of a search engine to retrieve candidate sources of plagiarism for a given suspicious document so that more accurate comparisons can be made. We describe a strategy for source retrieval that uses a supervised method to classify and rank search engine results as potential sources of plagiarism without retrieving the documents themselves. Evaluation shows the performance of our approach, which achieved the highest precision (0.57) and F_1 score (0.47) in the 2014 PAN Source Retrieval task.

1 Introduction

The advent of the Web has had many benefits in terms of access to information. However, it has also made it increasingly easy for people to plagiarize. For instance, in a study from 2002-2005, it was found that 36% of undergraduate college students admitted to plagiarizing [7]. Given the prevalence of plagiarism, there has been significant research and systems built for plagiarism detection [6].

There is an inherent assumption in most plagiarism detection systems that potential sources of plagiarism have already been identified that can be compared to a suspicious document. For small collections of documents, it may be reasonable to perform a comparison between the suspicious document and every document in the collection. However, this is infeasible for large document collections and on the Web. The source retrieval problem involves using a search engine to retrieve potential sources of plagiarism for a given suspicious document by submitting queries to the search engine and retrieving the search results.

In this paper we describe our approach to the source retrieval task at PAN 2014. The approach builds on our previous approach in 2013 [13], but differs by including a supervised search result ranking strategy.

2 Approach

Our approach builds on our previous approach to source retrieval [13], which achieved the highest F1 score at PAN 2013 [10]. The current approach involves four main steps: 1) query generation, 2) query submission, 3) result ranking and 4) candidate retrieval.

The key difference between our approach this year and our previous approach is in the result ranking step (step 3). Previously, we used a simple method whereby we re-ranked the results returned by the search engine based on the similarity of each result snippet and the suspicious document. This year, we make use of a supervised method for result ranking. Beyond that, the approach remains the same as in 2013. In the remainder of this section, we describe the approach while focusing on the supervised result ranking.

2.1 Query Generation

To generate queries, the suspicious document is partitioned into paragraphs with each paragraph consisting of 5 sentences as tagged by the Stanford Tagger [12]. Stop words are then removed and each word is tagged with its part of speech (POS). Following previous work [5][3], only words tagged as nouns, verbs and adjectives are retained while all others are discarded. Queries are then created by concatenating the remaining words to form queries consisting of 10 words each.

2.2 Query Submission

Queries are submitted in batches consisting of 3 queries and the results returned by each query are combined to form a single set of results. The intuition behind submitting queries in batches is that the union of the results returned by the three queries is more likely to contain a true positive than the results returned by a query individually. For each query, a maximum of 3 results is returned resulting in a set of at most 9 results. The intuition behind submitting 3 queries is that they likely capture sufficient information about the paragraph.

2.3 Result Ranking

We assume that the order of search results as produced by the search engine does not necessarily reflect the probability of the result being a source of plagiarism. We thus infer a new ranking of the results based on a supervised method. This is a major component of our approach and will be discussed in Section 3.

2.4 Candidate Document Retrieval

Having inferred a new ordering of the results, they are then retrieved in the new ranked order. For each result retrieved, the PAN Oracle is consulted to determine if that result is a source of plagiarism for the input suspicious document. If it is, then candidate retrieval is stopped and the whole process is repeated on the next paragraph. If it is not, then the next document is retrieved. Furthermore, a list is maintained of the URLs of each document retrieved and used to prevent a URL from being retrieved more than once since this does not improve precision or recall.

3 Supervised Result Ranking

The main difference between our current approach and our approach in the previous year is that now we make use of a supervised method. Specifically, we train a search result classifier that classifies each search result as either being a potential source of plagiarism or not. The classifier only makes use of features that are available at search time and without retrieving a search result unless it is classified as a potential source of plagiarism. Furthermore, an ordering of the positively classified search results is produced based on the probabilities output by the classifier.

3.1 Classifier

We make use of a Linear Discriminant Analysis (LDA) classifier that tries to find a linear combination of features for classification. We make use of the implementation of the LDA classifier from the *scikit-learn* machine learning toolkit [8]. The classifier produces a binary prediction of whether a search result is a candidate source of plagiarism. We sort the positively classified results by their probabilities of being positive as output by the classifier. This probability essentially reflects the confidence of the classifier in its prediction.

3.2 Features

For each search result that is returned by the search engine, we extract the following features, which we use for classification. All of these features are available at search result time and do not require the search result to be retrieved, which allows for classification to be performed as the search results become available. Many of these features are available from the ChatNoir search engine at search result time.

1. **Readability.** The readability of the result document as measured by the Flesh-Kincaid grade level formula [9] (ChatNoir).
2. **Weight.** A weight assigned to the result by the search engine (ChatNoir).
3. **Proximity.** A proximity factor [11] (ChatNoir).
4. **PageRank.** The PageRank of the result (ChatNoir).
5. **BM25.** The BM25 score of the result (ChatNoir).
6. **Sentences.** The number of sentences in the result (ChatNoir).
7. **Words.** The number of words in the result (ChatNoir).
8. **Characters.** The number of characters in the result (ChatNoir).
9. **Syllables.** The number of syllables in the result (ChatNoir).
10. **Rank.** The rank of the result, i.e. the rank at which it appeared in the search results.
11. **Document-snippet 5-gram Intersection.** The set of 5-grams from the suspicious document are extracted as well as the set of 5 grams from each search result snippet, where the snippet is the small sample of text that appears under each search result. A document-snippet 5-gram intersection score is then calculated as:

$$Sim(s, d) = S(s) \cap S(d), \tag{1}$$

where s is the snippet, d is the suspicious document and $S(\cdot)$ is a set of 5-grams.

12. Snippet-document Cosine Similarity. The cosine similarity between the snippet and the suspicious document, which is given by:

$$Cosine(s,d) = \cos(\theta) = \frac{V_s \cdot V_d}{||V_s||||V_d||}, \tag{2}$$

where $V.$ is a term vector.

13. Title-document Cosine Similarity. The cosine similarity between the result title and the suspicious document (Eq. 2).

14. Query-snippet Cosine Similarity. The cosine similarity between the query and the snippet (Eq. 2).

15. Query-title Cosine Similarity. The cosine similarity between the query and the result title (Eq. 2) [4].

16. Title length. The number of words in the result title.

17. Wikipedia source. Boolean value for whether or not the source was a Wikipedia article (based on the existence of the word "Wikipedia in title).

18. #Nouns. Number of nouns in the title as tagged by the Stanford POS Tagger [12].

19. #Verbs. Number of verbs in the title as tagged by the Stanford POS Tagger.

20. #Adjectives Number of adjectives in the title as tagged by the Stanford POS Tagger.

3.3 Training

We collect training data for the classifier by running our source retrieval method (as described above) over the training data provided as part of the PAN 2014 source retrieval task. However, instead of classifying and ranking the search results returned by the queries, we instead retrieve all of the results and consult the Oracle to determine if they are sources of plagiarism. This provides labels for the set of search results.

The training data was heavily imbalanced and skewed towards negative samples, which made up around 70% of the training data. It is well known that most classifiers expect an even class distribution in order for them to work well [2], thus oversampling is used to even the class distribution. The oversampling is performed using the SMOTE method, which creates synthetic examples of the minority class based on existing samples [2]. For each sample x_i of the minority class, the k nearest neighbors are identified and one of those nearest neighbors $\hat{x_i}$ is randomly selected. The difference between x_i and $\hat{x_i}$ is multiplied by a random number $r \in [0,1]$, which is then added to x_i to create a new data point x_{new}:

$$x_{new} = x_i + (\hat{x_i} - x_i) \times r. \tag{3}$$

The number of nearest neighbors considered is set to $k = 3$ and the minority class is increased by 200%. Since nearest neighbors are randomly selected, we train 5 classifiers. The final classification of a search result is based on the majority vote of the 5 classifiers and the probability output for ranking is based on the average probabilities produced by the 5 classifiers.

4 Evaluation

4.1 Retrieval Performance

Table 1 shows that performance of our approach on the test data as evaluated on Tira[1] [1].

Table 1. Performance of approach on test data

F_1	Precision	Recall	Queries	Downloads
0.47	0.57	0.48	117.13	14.41

Our approach achieved a relatively high precision of 0.57 and recall of 0.48. The F_1 score, which is the harmonic mean of precision and recall was 0.47. Overall, our approach was very competitive. Both the precision and F_1 score were the highest achieved among all participants. The number of queries submitted was relatively high compared to the other participants; however, as shown in [13], queries are relatively cheap from a bandwidth perspective though they do put additional strain on the search engine.

The features used for classification are also of interest to gain a better understanding of what features are important for source retrieval. While we do not discuss it here, a complete discussion is presented in [14].

5 Conclusions

We describe an approach to the source retrieval problem that makes used of supervised ranking and classification of search results. Overall, the approach was very competitive and achieved the highest precision and F_1 score among all task participants.

References

1. 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 Plagiarism Detection, Author Identification, and Author Profiling. In: Information Access Evaluation. Multilinguality, Multimodality, and Visualization. pp. 282–302 (2013)
2. He, H., Garcia, E.: Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263–1284 (Sep 2009)
3. Jayapal, A.: Similarity Overlap Metric and Greedy String Tiling at PAN 2012: Plagiarism Detection. CLEF (Online Working Notes/Labs/Workshop) (2012)
4. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02. pp. 133–142. ACM Press, New York, New York, USA (Jul 2002)

[1] The F_1 score is computed by averaging the F_1 score of each run rather than from the average precision and recall.

5. Liu, F., Pennell, D., Liu, F., Liu, Y.: Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 620–628. No. June (2009)
6. Maurer, H., Media, C., Kappe, F., Zaka, B.: Plagiarism - A Survey. Journal of Universal Computer Science 12(8), 1050–1084 (2006)
7. Mccabe, D.L.: Cheating among college and university students : A North American perspective. International Journal for Educational Integrity 1(1), 1–11 (2004)
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapear, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
9. Potthast, M., Gollub, T., Hagen, M., Graß egger, J., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-cede no, A., Gupta, P., Rosso, P., Stein, B.: Overview of the 4th International Competition on Plagiarism Detection pp. 17–20 (2012)
10. Potthast, M., Hagen, M., Gollub, T., Tippmann, M., Kiesel, J., Rosso, P., Stamatatos, E., Stein, B.: Overview of the 5th International Competition on Plagiarism Detection. In: CLEF 2013 Evaluation Labs and Workshop Working Notes Papers (2013)
11. Potthast, M., Hagen, M., Stein, B., Graß egger, J., Michel, M., Tippmann, M., Welsch, C.: ChatNoir: A Search Engine for the ClueWeb09 Corpus. In: Proceedings of the35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12). p. 1004 (Aug 2012)
12. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL ’03. vol. 1, pp. 173–180 (May 2003)
13. Williams, K., Chen, H., Choudhury, S., Giles, C.: Unsupervised Ranking for Plagiarism Source Retrieval - Notebook for PAN at CLEF 2013. In: CLEF 2013 Evaluation Labs and Workshop Working Notes Papers (2013)
14. Williams, K., Chen, H.H., Giles, C.L.: Classifying and Ranking Search Engine Results as Potential Sources of Plagiarism. In: ACM Symposium of Document Engineering (2014), (To appear)