Classifying and Ranking Search Engine Results as Potential Sources of Plagiarism

Kyle Williams, Hung-Hsuan Chen and C. Lee Giles

Information Sciences and Technology
Computer Science and Engineering
The Pennsylvania State University
Introduction

- The Web has led to an exponential increase in the amount of publicly available information
  - Benefits in healthcare, education, disaster management and many other domains

- But, it has also made it easy to plagiarize
  - 36% of undergraduate college students admitted to plagiarizing from the Web (McCabe, 2004)
  - 1 in 3 American high school students admitted to plagiarizing from the Web (Character Counts, 2010)
Given a suspicious document and a potential source document for plagiarism, find all areas of overlapping text, which may have been subjected to obfuscation.

Assumes that candidate sources of plagiarism have been identified for comparison.
Before detecting where plagiarism occurs, we need to get a candidate set of sources for a suspicious document

- Cheaper than comparing all documents in a collection
- For large corpora, only feasible approach
Source Retrieval Problem (Problem 2)

*Given a suspicious document and a search engine, use the search engine to retrieve candidate documents that may be sources of plagiarism.*
Figure: Generic procedure to detect plagiarism (Reproduced from Potthast et al. (2013))
General Source Retrieval Strategy

1. Generate queries from a suspicious document
2. Submit queries to search engine
3. Retrieve results in some order

While:
1. Minimizing the number of queries
2. Minimizing the number of false positives retrieved
Contributions

● A strategy for the source retrieval problem involving:
  ○ The generation of queries from a suspicious document
  ○ An approach for submitting queries to increase the likelihood of true positives
  ○ A supervised method for classifying search results as potential sources of plagiarism without retrieving the documents themselves.

● An analysis of features for search result classification
Figure: An overview of the source retrieval strategy
Algorithm 1 General overview of source retrieval strategy

1: procedure SOURCE_RETRIEVAL(doc)
2:     paragraphs ← SPLIT_INTO_PARAGRAPHS(doc)
3:     for all $p \in$ paragraphs do
4:         $p \leftarrow$ PREPROCESS($p$)
5:         queries ← EXTRACT_QUERIES($p$)
6:         for $i = 0 \rightarrow n$ do ▷ n is the top n queries
7:             results ← SUBMIT_QUERIES(queries[$i$])
8:         end for
9:     results ← CLASSIFY_AND_RANK(results)
10:    for all result $\in$ results do
11:        if result is True then
12:            if PREVIOUS_SOURCE(result) = false then
13:                source ← DOWNLOAD(result)
14:                if IS SOURCE(source) then
15:                    print source
16:                PreviousSource ← source
17:                break
18:            end if
19:        end if
20:    end if
21: end for
22: end procedure
Main Insights

- The union of the results of multiple queries has a higher probability of containing a true positive that each query individually
  - So submit multiple queries and combine results

- The ranking of the search results does not reflect the probability of a true positive
  - So re-rank results

Submit multiple queries ➔ combine results ➔ rank combined results
### Query 1
- **plagiarism detection**

549,000 RESULTS Any time

- **Duplichecker - Plagiarism Checker - Free**
- Free plagiarism checker for avoiding plagiarism, best detecting plagiarism online.

- **Free Plagiarism Detector and Duplicity Check**
- plagiarism-detect.com
- Most accurate free online plagiarism detector ... Imp not work correctly last few weeks, due to change in th

- **Plagiarism detection - Wikipedia, the free**
- en.wikipedia.org/wiki/Plagiarism_detection
- Plagiarism detection is the process of locating insta
document. The widespread use of computers and the

### Query 2
- **detecting plagiarism**

806,000 RESULTS Any time

- **Plagiarism.org - Best Practices for Ensuring Originality in Written Work**
- www.plagiarism.org
- Welcome to Plagiarism.org, your source of information for ensuring originality in written work.

- **Plagiarism Detect**
- plagiarism-detect.com
- Important The plagiarism checking did not work in the Bing Search API terms. Sorry for the incon

- **Preventing and Detecting Plagiarism - Resources for Preventing and Detecting Plagiarism**
- faculty.millikin.edu/~mgeorge/plagiarism
- Resources for Preventing and Detecting Plagiarism of Web resources on preventing and detecting pl

### Query 3
- **how to detect plagiarism**

6,680,000 RESULTS Any time

- **How to Detect Plagiarism | eHow**
- www.ehow.com/Education/K-12/K-12-For-Educ
- How to Detect Plagiarism. If that beautiful sentence paper seems too good to be true, it may well be. P

- **Use These 10 Sites to Detect Plagiarism**
- mashable.com/2012/08/29/plagiarism-online-service
- Aug 29, 2012 - Plagiarism is a serious offense. Teachers' text, or avoid getting into academic hot w

- **Resources For Teachers: How to Detect Plagiarism**
- writing.mit.edu/wcc/resources/teachers/detectplag
- Plagiarism is unacceptable academic behavior. To prevent plagiarism from occurring in the first place

---

Combine all of these results and rank them
Query Generation

- Partition document into paragraphs consisting of 5 sentences

- Tag each word with its part of speech (POS)
  - Discard all POS except nouns, verbs and adjectives

- A query made up by combining each sequence of 10 words with no overlap
Query Submission and Retrieval

- Submit 3 queries per paragraph as a batch
- Retrieve top 3 results for each query
- Classify and rank each search result without retrieving the document
- Retrieve results in ranked order
  - Stop when a true positive is returned (as labeled by an Oracle)
Data Set

- 40 plagiarized documents from PAN 2013 competition on plagiarism detection
  - For each document, have a list of sources from which it was plagiarized
  - 20 documents used for training
  - 10 documents used for validation
  - 10 documents used for testing

<table>
<thead>
<tr>
<th>Table 1: Descriptive statistics for number of words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1873</td>
</tr>
</tbody>
</table>
Labelling Data

- An Oracle is provided by PAN that provides a label of whether a search result is a source of plagiarism for a given suspicious document
  - Used to label all search results returned by all queries as sources of plagiarism or not
Training data
- 2,737 queries constructed
- 5,740 search results returned
- 4,240 negative, 1,500 positive

Validation data
- 1,331 queries constructed
- 2,940 search results returned
- 2,365 negative, 575 positive

Testing data
- 1,303 queries constructed
- 2,991 search results returned
- 2,174 negative, 817 positive
Data Sampling

- Data is heavily skewed towards negative samples (73.87% of training data)
  - Most learning algorithms expect balanced classes
  - Upsample positive samples in training data

- SMOTE Algorithm creates synthetic samples from existing data points
  - K nearest neighbors are selected for each sample \( x \)
  - One nearest neighbor \( x^\wedge \) is randomly selected
  - New sample created:
    \[
    x_{\text{new}} = x_i + (\hat{x}_i - x_i) \times r \quad r \in [0, 1]
    \]
  - Set \( k=3 \), upsample positive class by 200%
Search Result Classification

● When search results are returned, we want to classify them as being sources of plagiarism or not and infer and ordering of the results
  ○ Only retrieve those that are classified as sources of plagiarism

● Use a supervised method and compare it to a baseline that achieved the highest F1 score in the PAN 2013 source retrieval task
Baseline

Classification based on 5-gram intersection of suspicious document \( (d) \) and snippet of each search result \( (s) \)

\[
Sim(s, d) = S(s) \cap S(d)
\]

Where \( S() \) is the set of 5-word sequences.

Re-rank by \( Sim(s,d) \)
Supervised Methods

- Linear Discriminant Analysis (LDA)
- Logistic Regression
- Random Forest
- AdaBoosting with Decision Trees

Ranking of results either based on order in which they were classified or probabilities output by classifiers

Details of methods and parameters in the paper.
Majority Voting Ensemble

- Majority voting ensembles combine several independent classifiers and allow them to cast a vote
  - The majority vote wins
  - Individual classifiers should be accurate and diverse

\[
M(x) = \sum_{i=1}^{n} w_i C_i(x)
\]

- \(M(x)\): majority vote
- \(n\): number of classifiers
- \(w_i\): weight of classifier \(i\)
- \(C_i(x)\): classification of classifier \(i\)
Features (1)

1. **Readability.** The readability of the result document as measured by the Flesh-Kincaid grade level formula [22] (ChatNoir).

2. **Weight.** A weight assigned to the result by the search engine (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.
Features (2)

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:

\[
Sim(s, d) = S(s) \cap S(d),
\]

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:

\[
\text{Cosine}(s, d) = \cos(\theta) = \frac{V_s \cdot V_d}{||V_s|| ||V_d||},
\]

where \( V_\cdot \) is a term vector.

13. **Title-document Cosine Similarity.** The cosine similarity between the result title and the suspicious document (Eq. 2).
Features (3)

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) [13].

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 [28].

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.

All of the features can be calculated without actually retrieving the document.
Experiment Methodology

- Ran source retrieval algorithm with different classifiers and ranking methodologies
- Evaluate output using precision, recall and F-1 score

\[
Precision = \frac{tp}{tp + fp} \quad \text{Recall} = \frac{tp}{tp + fn}
\]

\[
F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}
\]

tp: true positive, fp: false positive, fn: false negative
Results with No Ranking

Table 2: Precision, recall and the $F_1$ score for the baseline method and different supervised methods. No ranking of results is used, i.e. they are retrieved in the order they were classified.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$ Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.3735</td>
<td>0.8543</td>
<td>0.5198</td>
</tr>
<tr>
<td>LDA</td>
<td><strong>0.3894</strong></td>
<td><strong>0.8803</strong></td>
<td><strong>0.5399</strong></td>
</tr>
<tr>
<td>Logistic</td>
<td>0.3848</td>
<td>0.8629</td>
<td>0.5322</td>
</tr>
<tr>
<td>Random Forests (RF)</td>
<td>0.3625</td>
<td>0.8725</td>
<td>0.5122</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>0.3811</td>
<td>0.8414</td>
<td>0.5246</td>
</tr>
</tbody>
</table>

- LDA achieves best precision and recall
- All supervised methods except RF outperform the baseline on the F1 score
- A small increase in recall can lead to a large increase in finding plagiarism in large collections
Results with Probabilistic Ranking

Table 3: Precision, recall and the $F_1$ score for the baseline and different supervised methods. The search results were ranked by the probabilistic output of the classifiers.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$ Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.3735</td>
<td>0.8543</td>
<td>0.5198</td>
</tr>
<tr>
<td>LDA+ProbRank</td>
<td>0.4063</td>
<td>0.8681</td>
<td>0.5535</td>
</tr>
<tr>
<td>Logistic+ProbRank</td>
<td>0.4019</td>
<td>0.8553</td>
<td>0.5469</td>
</tr>
<tr>
<td>RF+ProbRank</td>
<td>0.3833</td>
<td>0.8651</td>
<td>0.5311</td>
</tr>
<tr>
<td>AdaBoost+ProbRank</td>
<td>0.4018</td>
<td>0.8367</td>
<td>0.5429</td>
</tr>
</tbody>
</table>

- LDA remains best and F1 score improves compared to no ranking
- All supervised methods achieve higher F1 scores
Results with Voting Ensemble

Table 4: Precision, recall and the $F_1$ score for the baseline and ensemble classifiers.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$ Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.3735</td>
<td>0.8543</td>
<td>0.5198</td>
</tr>
<tr>
<td>Ensemble-Top3</td>
<td><strong>0.3874</strong></td>
<td>0.8681</td>
<td><strong>0.5357</strong></td>
</tr>
<tr>
<td>Ensemble-Top5</td>
<td>0.3868</td>
<td><strong>0.8825</strong></td>
<td><strong>0.5379</strong></td>
</tr>
</tbody>
</table>

- Top 3 classifier and top 5 classifiers
- Improves on baseline but does not do better than LDA+ProbRank
- Classifiers not sufficiently diverse?
Feature Analysis

- Evaluated the importance of each feature in trained random forest
- Baseline feature is the most important
- Similarity between title and document
- Wikipedia is an important source of plagiarism
- The ranking of the search engine doesn’t matter

Table 5: Importance of different features in the random forest

<table>
<thead>
<tr>
<th>Rank</th>
<th>No.</th>
<th>Feature</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Doc-snippet intersection</td>
<td>11</td>
<td>0.39</td>
</tr>
<tr>
<td>2</td>
<td>Title-doc cosine</td>
<td>13</td>
<td>0.16</td>
</tr>
<tr>
<td>3</td>
<td>Wikipedia source</td>
<td>17</td>
<td>0.09</td>
</tr>
<tr>
<td>4</td>
<td>Snippet-doc cosine</td>
<td>12</td>
<td>0.07</td>
</tr>
<tr>
<td>5</td>
<td>#Adjectives</td>
<td>20</td>
<td>0.07</td>
</tr>
<tr>
<td>6</td>
<td>Proximity</td>
<td>3</td>
<td>0.06</td>
</tr>
<tr>
<td>7</td>
<td>Query-snippet cosine</td>
<td>14</td>
<td>0.03</td>
</tr>
<tr>
<td>8</td>
<td>Syllables</td>
<td>9</td>
<td>0.02</td>
</tr>
<tr>
<td>9</td>
<td>Sentences</td>
<td>6</td>
<td>0.01</td>
</tr>
<tr>
<td>10</td>
<td>BM25</td>
<td>5</td>
<td>0.01</td>
</tr>
<tr>
<td>11</td>
<td>Words</td>
<td>7</td>
<td>0.01</td>
</tr>
<tr>
<td>12</td>
<td>Title length</td>
<td>16</td>
<td>0.01</td>
</tr>
<tr>
<td>13</td>
<td>Query-title cosine</td>
<td>15</td>
<td>0.01</td>
</tr>
<tr>
<td>14</td>
<td>Characters</td>
<td>8</td>
<td>0.01</td>
</tr>
<tr>
<td>15</td>
<td>Weight</td>
<td>2</td>
<td>0.01</td>
</tr>
<tr>
<td>16</td>
<td>Readability</td>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>17</td>
<td>Rank</td>
<td>10</td>
<td>0.00</td>
</tr>
<tr>
<td>18</td>
<td>#Nouns</td>
<td>18</td>
<td>0.00</td>
</tr>
<tr>
<td>19</td>
<td>#Verbs</td>
<td>19</td>
<td>0.00</td>
</tr>
<tr>
<td>20</td>
<td>PageRank</td>
<td>4</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Conclusions

- Developed a supervised method for classifying search results as sources of plagiarism

- Supervised methods improve on the baseline
  - Precision: 3.28%, recall: 2.6%, F1: 3.37%

- Insight into which features are important

- Future work involves new features and new query generation strategies
Thanks

- Partial support by the National Science Foundation
- PAN organizers
References


Character Counts. 2010 Report: Honesty and Integrity, 2010
