Approaches for Intrinsic and External Plagiarism Detection

Notebook for PAN at CLEF 2011

Gabriel Oberreuter Gallardo
goberreu@ing.uchile.cl
University of Chile - September 2011

Group members: Gastón L’Huillier, Sebastián Ríos and Juan D. Velásquez
Who are we?

- Group of students, professionals and professors from University of Chile
- We work on studying plagiarism in academia
- www.docode.cl
Who we are

Docode Engine

**DOCODE Engine**
Integration of document copy detection algorithms, collecting and indexing documents, queuing large scale copy detection petitions.

Docode ASP

**Application Service DOCODE**
Web application for DOCODE, design of document copy detection reporting tools, software engineering, features and requirement analysis.

Docode Impact

**DOCODE Impact Analysis**
Surveys and interviews in schools, social analysis of copy and paste phenomenon, evaluation of the impact of such tools in education.
Index

1. Where we were @2010
2. External Plagiarism Detection
3. Intrinsic Plagiarism Detection
4. Results @PAN2011
5. Conclusions
Where we were @2010
Where we were @2010

- Focused on external plagiarism detection.
- No cross-lingual consideration.
- Based on word bi-grams and word tri-grams.
- Selecting *samples* for each document in order to reduce compute time.
External Comparison between 2 documents
Where we were @2010

External Comparison between 2 documents
Monolingual
Based on word bi-grams and word tri-grams.
No intrinsic detection

Acceptable precision, detecting half of cases and good granularity.

Comparison computed on two eight-core Servers, each with 6 GB of RAM.
Java Implementation.
Reducing Search Space: ~20 Hours.
Finding Plagiarized Passages: ~12 Hours.
External Plagiarism Detection
External Plagiarism Detection

From 2010 experience, we decided to focus on:

- Better precision
- Better recall
- Reduce processing time

External detector @2011

- Uses word 4-grams, removing SW for search space reduction
- Uses word 3-grams for exhaustive search
External Plagiarism Detection

Some results on 2010 PAN Corpus (includes intrinsic and external plagiarism):

<table>
<thead>
<tr>
<th>Algorithm Version</th>
<th>Overall</th>
<th>Recall</th>
<th>Precision</th>
<th>Granularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>0.61</td>
<td>0.48</td>
<td>0.85</td>
<td>1.001</td>
</tr>
<tr>
<td>2011</td>
<td>0.73</td>
<td>0.6</td>
<td>0.94</td>
<td>1.001</td>
</tr>
</tbody>
</table>

- Dual core notebook with 4GB RAM.
- Java Implementation.
- Reducing Search Space: ~2 Hours. (0.3% promising doc pairs)
- Exhaustive Search: ~1 Hour.
Intrinsic Plagiarism Detection
Intrinsic Plagiarism Detection

Given a document, determine whether one of its paragraphs belong to the average writing style.
Intrinsic Plagiarism Detection

Following Stamatatos’ (2009) approach:

- Divide the document in partitions
- Compare each partition’s *writing style characterization* against the whole document’s style
- If a partition deviates from the mean value past some threshold, flag it

<table>
<thead>
<tr>
<th>Rank</th>
<th>Overall score</th>
<th>F-measure</th>
<th>Precision</th>
<th>Recall</th>
<th>Granularity</th>
<th>Participant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2462</td>
<td>0.3086</td>
<td>0.2321</td>
<td>0.4607</td>
<td>1.3839</td>
<td>E. Stamatatos University of the Aegean, Greece</td>
</tr>
<tr>
<td>2</td>
<td>0.1955</td>
<td>0.1956</td>
<td>0.1091</td>
<td>0.9437</td>
<td>1.0007</td>
<td>B. Hagbi and M. Koppel Bar Ilan University, Israel</td>
</tr>
<tr>
<td>3</td>
<td>0.1766</td>
<td>0.2286</td>
<td>0.1968</td>
<td>0.2724</td>
<td>1.4524</td>
<td>M. Granitzer, M. Muhr, M. Zechner, and R. Kern Know-Center Graz, Austria</td>
</tr>
<tr>
<td>4</td>
<td>0.1219</td>
<td>0.1750</td>
<td>0.1036</td>
<td>0.5630</td>
<td>1.7049</td>
<td>L. M. Seaward and S. Matwin University of Ottawa, Canada</td>
</tr>
</tbody>
</table>
Intrinsic Plagiarism Detection

On the characterization of writing style…

If some of the words used on the document are author-specific, one can think that those words could be concentrated on the paragraphs (or more general, on the segments) that the mentioned author wrote.
Intrinsic Plagiarism Detection

Fundamentals:

- Divide document in partitions of equal length
- Word Frequencies
- No stopword removal
- Only chars from a-z
Comparing the partitions against the whole document…

Algorithm 1 Intrinsic plagiarism evaluation

Require: $C$, $v$, $m$, $\delta$

1: for $c \in C$ do
2:     $d_c \leftarrow 0$
3:     build $v_c$ using term frequencies on segment $c$
4:     for word $w \in v_c$ do
5:         $d_c \leftarrow d_c + \frac{\mid freq(w,v) - freq(w,v_c)\mid}{\mid freq(w,v) + freq(w,v_c)\mid}$
6:     end for
7: end for
8: style $\leftarrow \frac{1}{|C|} \sum_{c \in C} d_c$
9: for $c \in C$ do
10:    if $d_c < style - \delta$ then
11:       Mark segment $c$ as outlier and potential plagiarized passage.
12:    end if
13: end for
Example 1: Document written by single author
Example 2: Document written by multiple authors
Some results on 2009 PAN Intrinsic Corpus:

<table>
<thead>
<tr>
<th></th>
<th>Overall</th>
<th>Recall</th>
<th>Precision</th>
<th>Granularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stamatatos</td>
<td>0.25</td>
<td>0.46</td>
<td>0.23</td>
<td>1.38</td>
</tr>
<tr>
<td>Oberreuter</td>
<td>0.34</td>
<td>0.31</td>
<td>0.39</td>
<td>1.01</td>
</tr>
</tbody>
</table>

- Dual core notebook with 4GB RAM.
- Java Implementation.
- Run under 10 minutes (~6,000 documents).
Results @PAN2011
External Plagiarism Detection Performance

- Ranked third after Grman&Ravas (0.56) and Grozea&Popescu (0.42)
- Overall good precision, but low recall for obfuscated plagiarism and simulated plagiarism
Results @PAN2011

Intrinsic Plagiarism Detection Performance

- Ranked first with a good recall-precision balance
- Overall score of 0.32, with better results with medium- and long-length documents

<table>
<thead>
<tr>
<th></th>
<th>plagDet</th>
<th>Recall</th>
<th>Precision</th>
<th>Granularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>overall</td>
<td>0.3254817</td>
<td>0.3397965</td>
<td>0.3123243</td>
<td>1.0000000</td>
</tr>
<tr>
<td>doc-length-long</td>
<td>0.3787308</td>
<td>0.3828166</td>
<td>0.3747313</td>
<td>1.0000000</td>
</tr>
<tr>
<td>doc-length-medium</td>
<td>0.4001631</td>
<td>0.3660643</td>
<td>0.4412672</td>
<td>1.0000000</td>
</tr>
<tr>
<td>doc-length-short</td>
<td>0.2811900</td>
<td>0.2479395</td>
<td>0.3247399</td>
<td>1.0000000</td>
</tr>
<tr>
<td>translated-obfuscation</td>
<td>0.3131128</td>
<td>0.2789482</td>
<td>0.3568141</td>
<td>1.0000000</td>
</tr>
<tr>
<td>translated-manual-obfuscation</td>
<td>0.1095166</td>
<td>0.1276579</td>
<td>0.0958898</td>
<td>1.0000000</td>
</tr>
</tbody>
</table>
Conclusions

- Word tri-grams and word 4-grams can be used effectively as tokens for external plagiarism detection.
- The effectiveness of the approach is strongly correlated to the ability to detect those dense coincidence zones.
- When no sources are available, the use of words appear to be a good starting point to model the writing style present in documents.
- Best result in self-information task, but the scores are overall still too low.