Procesamiento del Lenguaje Natural, Revista nº 45, septiembre 2010, pp 277-280 recibido 21-06-10 revisado 05-07-10 aceptado 06-07-10

# Application of Information Retrieval Techniques to Document Filtered Set Generation for External Plagiarism Detection

## *Aplicación de Técnicas de Recuperación de Información a la Generación de Conjuntos Filtrados de Documentos para la Detección de Plagios Externos*

**Daniel Micol, Óscar Ferrández, and Rafael Muñoz**
Research Group on Natural Language Processing and Information Systems
Department of Software and Computing Systems
University of Alicante
San Vicente del Raspeig, E-03080 Alicante, Spain
{dmicol, ofe, rafael}@dlsi.ua.es

**Resumen:** En este artículo presentamos un método para la generación de conjuntos filtrados de documentos empleando técnicas de recuperación de información. Esto se presenta en el contexto de la detección de plagios externos, aunque las técnicas detalladas en este artículo son aplicables a cualquier tipo de documentos o consultas. La producción de conjuntos filtrados, y por ende la limitación del espacio de búsqueda del problema, puede resultar en una gran mejora de rendimiento y es utilizada hoy en día en gran cantidad de aplicaciones reales, como buscadores web. Respecto a la detección de plagios en documentos, la base de datos de textos con los que comparar el candidato sospechoso es potencialmente grande, y por lo tanto es muy recomendable aplicar técnicas de generación de conjuntos filtrados.
**Palabras clave:** Conjunto Filtrado, Recuperación de Información, Detección de Plagios.

**Abstract:** In this paper we present an approach to generate document filtered sets using information retrieval techniques. This is presented in the context of external document plagiarism detection, although the techniques detailed in this paper are applicable to any sort of documents or queries. Producing filtered sets, and hence limiting the problem's search space, can be a tremendous performance improvement and is used today in many real world applications such as web search engines. With regards to document plagiarism detection, the database of documents to match the suspicious candidate against is potentially fairly large, and hence it becomes very recommendable to apply filtered set generation techniques.
**Keywords:** Filtered Set, Information Retrieval, Plagiarism Detection.

## *1 Introduction*

External plagiarism detection is a complex task that attempts to determine if a suspicious document is an appropriation of another document which belongs to a set of candidates (Potthast et al., 2009). It is a very expensive task given that the number of documents to compare against is potentially large. To perform this operation in an efficient way, it is recommended to generate a filtered set of the mentioned candidate documents, in order to be able to later on apply over this subset a more complex and costly function that will detect the corresponding plagiarized document, if any (Stein, zu Eissen, and Potthast, 2007).

To develop and measure the approaches described in this paper we have used the *1st International Competition on Plagiarism Detection* (Potthast et al., 2009) as framework. The committee of this competition provides annotated corpora as well as a definition of the measures to evaluate our methods. Concretely, it provides a source documents corpus, containing those that may have been plagiarized, and a suspicious documents corpus, containing those that might include the plagiarism itself. The documents included are mostly written in English, but some of them are also in Spanish and German.

The system that we are developing to

ISSN 1135-5948 

detect document plagiarism is composed of three major modules, similar to what is described in (Stein, zu Eissen, and Potthast, 2007): corpus document filtering, document selection and passage matching. In this paper we will describe the first of these components, which reduces the candidate document search space by identifying those that are likely to be the source of the plagiarized document. To accomplish this we create an inverted index of the source documents from our corpus and perform n-gram queries against it, similar to what is described in (Kasprzak, Brandejs, and Křipač, 2009).

The remainder of this paper is structured as follows. In the second section we will describe the methods implemented in our system. The third one contains the experimental results, and the fourth and last discusses such results and proposes future work based on our current research.

## *2 Methods*

The component that we have developed contains an indexing phase and the filtering itself. To implement this, we needed to allow the storage of our corpus' contents into an inverted index, and querying against it. Therefore, we required to have a full-text search engine, and we chose Lucene (Gospodnetic, Hatcher, and McCandless, 2009) for this purpose.

### 2.1 Indexing

The first step in order to produce a filtered set is to index the documents from our corpus of candidate documents to be the source of the plagiarism, given that we will have a large amount and we require fast retrieval. Lucene creates an inverted index that will allow efficient multi-word queries.

### 2.2 Filtering

In order to produce a filtered set of documents, we must calculate the similarity between the suspicious document and every candidate that we have indexed. Then, we will extract those documents that have the highest similarity scores, and these will compose the corresponding filtered set.

The main difference between our approach and the one described in (Kasprzak, Brandejs, and Křipač, 2009) is the document scoring function. The aforementioned paper proposes a method to calculate this score based on the number of words that appear in both documents. In our case, however, we apply a more complex function that takes into consideration several factors. Concretely, we use the document scoring function implemented by Lucene, called *Lucene's Practical Scoring Function*, which is applied over a query and a document, and is defined as shown in the following equation (Gospodnetic, Hatcher, and McCandless, 2009; Manning, Raghavan, and Schütze, 2008):

$$\begin{aligned} score(q,d) &= C(q,d) \cdot QN(q) \cdot \\ &\cdot \sum_{t \in q} (tf(t) \cdot idf(t)^2 \cdot \\ &\cdot B(t) \cdot N(t,d)) \end{aligned}$$

where $q$ is a query, $d$ is a document, $tf(t)$ is the term frequency of term $t$ in document $d$, $idf(t)$ is the inverse document frequency of term $t$, $C(q,d)$ is a score factor based on how many of the query terms are found in the specified document, $QN(q)$ is a normalizing factor used to make scores between queries comparable, $B(t)$ is a search time boost of term $t$ in the query $q$, and $N(t,d)$ encapsulates a few boost and length factors. The result of this function will be a normalized similarity value between 0 and 1.

In our system, the queries will be n-grams extracted from the documents that we want to classify as plagiarized or not. We have experimented with two ways of extracting these n-grams: fixing a given n-gram size, or using sentences (i.e. n-grams delimited by punctuation symbols). The score of a document will be the maximum given by any of its n-grams.

## *3 Experimentation*

To experiment with the system described in this paper, we used the external plagiarism corpus from the *1st International Competition on Plagiarism Detection* (Potthast et al., 2009). Concretely, we indexed all 14,429 source documents and extracted an automatically generated random set of 80 documents from the suspicious ones. Our goal is, for every plagiarized document, to create a filtered set that is minimal in size and contains the aforementioned one.

We used three metrics to evaluate our system, all of them returning percentage values. The first one, recall, measures how many of

the plagiarized documents are in the generated filtered set. Then, precision measures how many elements have been filtered and do not belong to the filtered set, regardless of whether the plagiarized ones are included or not. Finally, f-score is defined as a combination of recall and precision.

Given that there is a trade-off in our experiments between recall and precision, we used f-score to determine the best result, given that this measure is a combination of the previous two.

### 3.1 Filtered set size

Our first experiment consisted in fixing the filtered set size and observing what is the impact of this value in recall, precision and f-score. The filtered set will be filled with the documents that contain the highest similarity scores. The queries used by the information retrieval system are sentences extracted from every suspicious document. We identify sentences based on spacing and punctuation symbols. The results from this experiment are shown in Table 1.

Table 1: Metrics using different filtered set sizes and based on sentence queries.

| Size | Recall | Precision | F-score |
|---|---|---|---|
| 1 | 0.12 | 1.0 | 0.21 |
| 10 | 0.37 | 1.0 | 0.54 |
| 100 | 0.59 | 0.99 | 0.74 |
| 500 | 0.75 | 0.97 | 0.85 |
| 1000 | 0.81 | 0.94 | 0.87 |
| 2000 | 0.87 | 0.88 | 0.88 |
| 5000 | 0.97 | 0.74 | 0.84 |
| 10000 | 0.99 | 0.58 | 0.73 |

As we see in the previous table, recall increases with the filtered set size, and precision decreases (by definition). We find that the maximum f-score value happens when we have a filtered set of between $1,000$ and $2,000$ elements, being this about one fourteenth or one seventh of our source documents corpus, given that this one contained more than $14,000$ documents. Recall for this configuration is relatively high, between 81% and 87%, which means that we can reduce the problem size by about one fourteenth or one seventh and still have a good coverage.

One big advantage of this approach is that it transforms the complexity of the problem from linear to constant, based on the number of elements that will compose the filtered set. This will allow us to ensure that our plagiarism detection software finishes within a given time window, making it suitable for real-life use cases. Other authors, such as (Grozea, Gehl, and Popescu, 2009), have also chosen to convert this problem into linear complexity as a way of making it computationally tractable.

### 3.2 Similarity score threshold

The second approach that we experimented with to generate filtered sets was to use a similarity score threshold, and only include those candidate documents that had the highest scores for any given suspicious document n-gram query. We considered different thresholds, including in the filtered set only those documents that had a similarity score equal or greater than this value. In this experiment we used a fixed n-gram size of 50 words. The results from this experiment are shown in Table 2.

Table 2: Metrics using different similarity score thresholds and based on n-gram queries of size 50.

| Threshold | Recall | Precision | F-score |
|---|---|---|---|
| 0.1 | 1.00 | 0.53 | 0.69 |
| 0.2 | 0.93 | 0.90 | 0.92 |
| 0.3 | 0.81 | 0.98 | 0.89 |
| 0.4 | 0.70 | 1.00 | 0.82 |
| 0.5 | 0.56 | 1.00 | 0.72 |
| 0.6 | 0.52 | 1.00 | 0.68 |
| 0.7 | 0.32 | 1.00 | 0.49 |
| 0.8 | 0.18 | 1.00 | 0.30 |
| 0.9 | 0.18 | 1.00 | 0.30 |
| 1.0 | 0.01 | 1.00 | 0.03 |

As we can see in the previous table, recall decreases as the threshold grows, and the opposite happens for precision. This is because higher thresholds are more restrictive and therefore will lead to smaller filtered sets. The f-score measure has a maximum value when the threshold is 0.2, so this would be the approximate optimal value for a similarity score threshold.

### 3.3 N-gram size

Finally, we experimented using different n-gram sizes as information retrieval queries. In addition, we kept the 0.2 threshold restriction. The results from this experiment are shown in Table 3.

Table 3: Metrics using different n-gram sizes as queries and a score threshold of 0.2.

| Size | Recall | Precision | F-score |
|---|---|---|---|
| 10 | 1.00 | 0.42 | 0.59 |
| 20 | 1.00 | 0.64 | 0.78 |
| 50 | 0.93 | 0.90 | 0.92 |
| 100 | 0.96 | 0.95 | 0.95 |
| 200 | 0.80 | 0.97 | 0.87 |
| 500 | 0.47 | 0.97 | 0.63 |
| 1000 | 0.25 | 0.97 | 0.40 |

As we see in the previous table, recall decreases when the n-gram size increases, and the opposite applies to precision. However, this tendency is inverted for recall between sizes 50 and 100. This could be due to noise and the nature of the corpora used in our experimentation. Therefore, the optimal n-gram sizes are those two values.

## 4 Conclusions and future work

In this paper we have described a method for filtered set generation using information retrieval techniques and changing different parameters such as filtered set size, similarity score threshold and n-gram size. As expected, looser restrictions produce larger filtered sets with high recall, whereas stronger restrictions produce smaller filtered sets with low recall. Choosing the appropriate parameter values will depend on the user's needs. However, we believe that fixing the filtered set size is the best option given that it will transform the complexity of the problem from linear to constant, hence ensuring that the corresponding plagiarism detection software finishes within a given time window.

We believe the methods described in this paper to be useful for external document plagiarism detection, as they reduce by a large factor the number of documents to compare against, improving performance considerably. Furthermore, most of these methods are language agnostic, so they can be easily applied to iberian languages, for instance.

As future work we would like to build a query level ranker that is applied over the filtered set documents and extracts those n-grams that have been plagiarized. This can be done using textual entailment recognition techniques, for instance. These approaches are computationally expensive, so we will most likely choose stronger restrictions to produce small filtered sets, sacrificing by a few points the accuracy of our overall system in favor of better performance.

## Acknowledgements

This research has been partially funded by the Spanish Ministry of Science and Innovation (grant TIN2009-13391-C04-01), the Conselleria d'Educació of the Spanish Generalitat Valenciana (grants PROMETEO/2009/119 and ACOMP/2010/286), and the University of Alicante post-doctoral fellowship program funded by Fundación CajaMurcia.

## References

Gospodnetic, Otis, Erik Hatcher, and Michael McCandless. 2009. *Lucene in Action*. Manning Publications, 2nd edition.

Grozea, Cristian, Christian Gehl, and Marius Popescu. 2009. ENCOPLOT: Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection. In *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 10–18.

Kasprzak, Jan, Michal Brandejs, and Miroslav Křipač. 2009. Finding Plagiarism by Evaluating Document Similarities. In *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 24–28.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Potthast, Martin, Benno Stein, Andreas Eiselt, Alberto Barrón Cedeño, and Paolo Rosso. 2009. Overview of the 1st International Competition on Plagiarism Detection. In *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 1–9.

Stein, Benno, Sven Meyer zu Eissen, and Martin Potthast. 2007. Strategies for retrieving plagiarized documents. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 825–826.