On the Use of Similarity Search to Detect Fake Scientific Papers

Kyle Williams[1] and C. Lee Giles[1,2]

[1] Information Sciences and Technology
[2] Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802 USA

Abstract. Fake scientific papers have recently become of interest within the academic community as a result of the identification of fake papers in the digital libraries of major academic publishers [8]. Detecting and removing these papers is important for many reasons. We describe an investigation into the use of similarity search for detecting fake scientific papers by comparing several methods for signature construction and similarity scoring and describe a pseudo-relevance feedback technique that can be used to improve the effectiveness of these methods. Experiments on a dataset of 40,000 computer science papers show that precision, recall and MAP scores of 0.96, 0.99 and 0.99, respectively, can be achieved, thereby demonstrating the usefulness of similarity search in detecting fake scientific papers and ranking them highly.

Keywords: similarity search, fake papers, SciGen

1 Introduction

In recent years their has been increasing pressure on academics to publish large numbers of articles in order to sustain their careers, obtain funding and ensure prestige. As a result, it has been argued that there has been a decrease in the quality of articles submitted for publication [3] as well as a surge in the number of for-profit, predatory, and low quality journals and conferences to meet the demand for venues for publication [2]. As a result, it is reasonable to expect that fraudulent and and fake scientific papers may exist in document collections [8] and their identification and removal is important for many reasons.

In this paper we address the problem of using similarity search to detect fake scientific papers as generated by SciGen[3], which is a computer science paper generator. The benefit of using similarity search as opposed to other methods, such as supervised text classification, is that the latter requires training. When one has the code to automatically generate fake papers, as is the case for SciGen, then it is relatively simple to generate training data. However, in the case where the code for generating fake papers is not available, it becomes a tedious

[3] http://pdos.csail.mit.edu/scigen/

task to create training data since training cases first need to be identified. In contrast, similarity search only requires one sample of a fake paper if we make the assumption that there exists some regularity among fake papers generated by the same method.

We investigate the use of several methods for similarity search for detecting SCIGen papers, including state of the art near duplicate detection methods and simpler keyword and keyphrase-based methods and demonstrate their effectiveness in retrieving fake SCIGen papers. One of the challenges of this approach, however, is that it requires documents to have features in common in order for them to be retrieved. We exploit the fact that we expect some regularity to exist among automatically generated documents and devise a pseudo-relevance feedback mechanism to improve the performance of similarity search.

2 Related Work

There has already some work on identifying fake scientific papers. Early work was based on the intuition that in a SciGen generated paper, the references are to fake or non-existent papers [10]. Thus, by analyzing the references, one is able to determine if a paper is fake or not. Thus, the authors extract the references from fake papers and submit them to a public Web search engine and a paper is classified as being fake or not based on the extent to which its references match actual search results. While this method is useful, it can easily be fooled by making the references in fake papers actually refer to real papers.

Labbé and Labbé do an analysis of the extent to which fake and duplicate papers exist in the scientific literature [4]. Their method is based on calculating the inter-textual distances between documents based on the similarity and frequency of the words appearing in the documents. Once the inter-textual differences have been calculated, texts are grouped using agglomerative hierarchical clustering.

A problem related to fake paper detection is plagiarism detection since in both cases the goal is to detect suspicious text. There has already been several efforts to use similarity search to detect plagiarism. For instance, one of the tasks in the annual PAN workshop and competition on uncovering plagiarism, authorship, and social software misuse focuses on the source retrieval problem [7]. In this task, the input is a suspicious document and the goal is to retrieve potential sources of plagiarism from the Web. Most of the approaches in this task view the problem as a similarity search problem where the goal is to retrieve search results that are similar to the query document. Competitive approaches have considered both supervised and unsupervised solutions to the problem [7].

This section has discussed various studies that have dealt with fake academic papers or using similarity search to retrieve content of interest. An important thing to note is that relatively small datasets were used in all of the studies involving SciGen. For instance, in [4], most of the corpora only contained 10s or 100s of documents, though the corpora based on the Arxiv contained a few thousand documents. By contrast, we perform our experiments on a dataset containing over 40,000 real papers to which we add 100 fake papers.

3 Approach

Given a sample SCIGen paper q as a query, we seek to retrieve all SCIGen documents in a collection C. To do this, we perform automatic feature extraction on every document in the collection and index the documents. At query time, we select a retrieval feature and use it to automatically extract features from q, which we then use to retrieve all documents that have at least one feature in common with q and rank the results.

3.1 Feature Extractors

Shingle Features. Shingles are sequences of words that occur in documents and were originally used for calculating the similarity of documents [1] and are considered as state of the art for near duplicate detection. Due to space constrains, we do not describe the method for generating the shingle features but use the same approach as in [9]. We experimented with different shingle lengths and found a length of 5 to work well for this study.

Simhash Features. Simhash is a state of the art algorithm duplicate detection algorithm [5]. For each document, the simhash is calculated as described in [9] and the output is a 64-bit hash. Each hash is partitioned into $k + 1$ sub-hashes and these sub-hashes are indexed [5]. At query time, the simhash of the query document is also partitioned into $k + 1$ sub-hashes that are used to query the index and retrieve documents that have at least one sub-hash in common with the query document. We experimented with different values of k and found $k = 4$ to work well. Thus, we use this value of k in this study.

Keyphrase Features. We extract keyphrases from each document using the Maui tool [6] and use these keyphrases as features. For each document, the top 10 keyphrases are identified for querying. Thus a document will be retrieved if its text contains at least one of the top 10 keyphrases in the query document.

TF-IDF Features. We also investigate the use of features based on TF-IDF. Each term in a query document is scored using TF-IDF. We then form a Boolean OR query with the top 10 TF-IDF scored terms and a document will be retrieved if it contains a term that matches one of the top 10 terms in the query document.

For each document retrieved using the features extracted by one of the feature extractors, we perform full-text based ranking based on cosine similarity.

3.2 Dataset

43,390 ACM papers from the CiteSeerX collection constitute our collection of *real* scientific papers. We then used SciGen to generate 100 *fake* papers and added these to the existing collection of real papers. We then generated an additional 10 fake papers for testing. In our experiments, the goal is to use the testing papers to retrieve the 100 known fake papers in the dataset.

4 Experiments

4.1 Retrieving SCIGen Papers

We consider the use of the four feature extractors for retrieving SCIGen papers using similarity search. For each of the 10 query documents, we extract features which we use to formulate a query and we report the averages over the 10 documents. Figure 1 shows the different metrics for the different feature extractors (the Shingles+Feedback approach is described in Section 4.2).

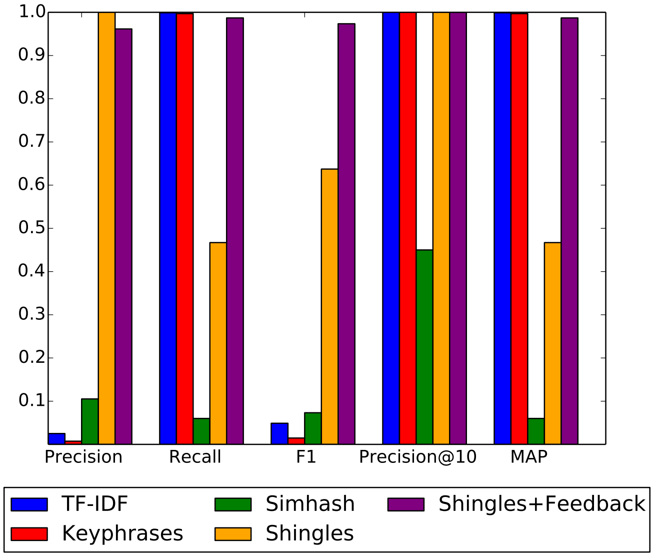

Fig. 1. Performance metrics for different feature extractors.

As can be seen from Figure 1, the different features perform quite differently in their ability to retrieve SCIGen papers. The first thing to notice is that almost perfect recall can be achieved by the TF-IDF and keyphrase-based methods, with average recall values of 0.999 and 0.997, respectively. This clearly indicates that these simple features are very good at identifying SCIGen papers; however, this comes at the cost of precision which, as can be seen from the figure, is very low for these two methods at 0.0251 and 0.0074, respectively. The reason for the very low precision for these methods is that many of the TF-IDF ranked terms and keyphrases are common among computer science papers and thus many documents are retrieved. The F-scores show that these methods perform worst overall in terms of overall retrieval with F-scores of 0.0489 and 0.0147. The TF-IDF scored keyword and keyphrase methods, however, achieve good rankings with Precision@10 of 1.0 and MAP of 0.999 and 0.997, respectively.

For the shingles method, the overall precision is perfect thereby implying that only SCIGen papers were retrieved. The downside of this approach, however, is that the recall is relatively low at 0.467. Overall though, the shingles method achieves the highest F1 score. Shingles also lead to perfect Precision@10; however, MAP is 0.467 since not all 100 SCIGen documents were retrieved.

The simhash method performs worst overall and achieves precision of 0.1052, recall of 0.06, Precision@10 of 0.45 and MAP of 0.06. This is somewhat expected since the simhash method is based on a single hash that represent a full document whereas the other methods are based on sub-documents. Since simhash is state of the art for near duplicate detection there is sufficient evidence to conclude that SCIGen documents are not similar enough to be called near duplicates.

The metrics that take into consideration the ranking of results are all relatively good and one can deduce from this that, in general, the cosine similarity-based ranking function is suitable since it places almost all retrieved SCIGen documents in the top 100 documents. While this is highly desirable, the one shortcoming is that, in this case, we know that there are 100 fake SCIGen documents and thus calculating MAP among the top 100 makes sense. However, in the general case, we do not know how many documents need to be detected. We are faced with the situation where we can achieve high recall at the cost of precision as is the case for the keyword and keyphrase-based methods, or we can achieve high precision at the cost of recall as is the case with the shingles-based method. In the next section, we describe a method whereby we can address this shortcoming. Due to space constrains, we focus on the case of shingles but the method is applicable to similarity search in general.

4.2 Improving Performance Through Pseudo-Relevance Feedback

In information retrieval, feature mismatch occurs when the terms that a user uses to describe a document do not match the terms used by the document authors. The standard way to address this problem is through query reformulation. We extend this approach to the detection of SCIGen papers where we expect some feature regularity among a sufficiently large number of SCIGen documents. We devise a pseudo-relevance feedback mechanism whereby after the initial query is submitted, we select the top k returned documents and submit each of them as a query using the same method as for the original document. We then combine and rank all the search results returned from the different query documents. The motivation behind this approach is that, while the initial query document may not have features in common with all relevant documents in the collection, documents that are retrieved might. The Shingles+Feedback bar in Figure 1 shows the effect of performing this pseudo-relevance feedback on the top 10 documents returned by the initial query with shingle features.

As can be seen in Figure, the effect of the pseudo relevance feedback has a large effect on recall, which was the initial shortcoming of the original shingle-based method. When the pseudo-relevance feedback is included, their is a slight decrease in overall precision from 1.0 to 0.96, however this comes at the benefit of an almost 2-fold increase in recall from 0.467 to 0.987. As a result, the recall becomes competitive with that achieved by the keyword and keyphrase-based methods. This increase in recall is reflected in the change in the F-score which increases from 0.64 to 0.97. The pseudo-relevance feedback has no effect on Precision@10, which remains at 1.0, but leads to a large increase in MAP which, like recall, goes from 0.467 to 0.987. Thus, there is clear evidence from this experiment that exploiting the expected regularity among automatically generated documents is a reasonable approach in order to improve retrieval performance.

5 Conclusions

We have described a method whereby similarity search can be used to detect fake scientific papers, which have increasingly become a problem as a result of the increasing pressure on academics to publish or perish. We described several methods for extracting features for similarity search and evaluated their use in detecting SCIGen papers. Inspired by the fact that we expect some form of regularity to exist among automatically generated documents, we devised a pseudo-relevance feedback mechanism to improve the performance of similarity search and showed how precision, recall and MAP scores of 0.96, 0.99 and 0.99, respectively, can be achieved. We only presented an evaluation of the pseudo-relevance feedback mechanism with shingle features; however, the approach is general enough that it can be applied to any set of features for similarity search.

Acknowledgments

We gratefully acknowledge partial support by the National Science Foundation.

References

1. Broder, A., Glassman, S., Manasse, M., Zweig, G.: Syntactic clustering of the Web. Computer Networks and ISDN Systems 29(8-13), 1157–1166 (Sep 1997)
2. Butler, D.: Investigating journals: The dark side of publishing. Nature 495(7442), 433–5 (Mar 2013)
3. Gad-el Hak, M.: Publish or perish - an ailing enterprise? Physics Today 57(3), 61–62 (Mar 2004)
4. Labbé, C., Labbé, D.: Duplicate and fake publications in the scientific literature: how many SCIgen papers in computer science? Scientometrics 94(1), 379–396 (Jun 2012)
5. Manku, G., Jain, A., Sarma, A.D.: Detecting near-duplicates for web crawling. In: WWW. pp. 141–149 (2007)
6. Medelyan, O., Frank, E., Witten, I.H.: Human-competitive tagging using automatic keyphrase extraction. In: EMNLP. vol. 3, pp. 1318–1327 (2009)
7. Potthast, M., Hagen, M., Beyer, A., Busse, M., Tippmann, M., Rosso, P., Stein, B.: Overview of the 6th International Competition on Plagiarism Detection. In: CLEF (2014)
8. Van Noorden, R.: Publishers withdraw more than 120 gibberish papers. Nature (Feb 2014)
9. Williams, K., Giles, C.L.: Near Duplicate Detection in an Academic Digital Library. In: DocEng. pp. 91–94 (2013)
10. Xiong, J., Huang, T.: An Effective Method to Identify Machine Automatically Generated Paper. In: KESE. pp. 101–102. IEEE (2009)