Engineering a Tool to Detect Automatically Generated Papers

Minh Tien Nguyen, Cyril Labbé

Engineering a Tool to Detect Automatically Generated Papers

Nguyen Minh Tien, Cyril Labbé

Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France
Minh-tien.nguyen@imag.fr
Cyril.labbe@imag.fr

Abstract. In the last decade, a number of nonsense automatically-generated scientific papers have been published, most of them were produced using probabilistic context free grammar generators. Such papers may also appear in scientific social networks or in open archives and thus bias metrics computation. This shows that there is a need for an automatic detection process to discover and remove such nonsense papers. Here, we present and compare different methods aiming at automatically classifying generated papers.

1 Introduction

In this paper, we are interested in detecting fake *academic-papers* that are automatically created using a Probabilistic Context Free Grammar (PCFG). Although these kind of texts are fairly easy to detect by a human reader, there is a recent need to automatically detect such texts. This need has been highlighted by the *Ike Antkare*[1] experiment [1] and other studies [2]. Detection methods and tools are useful for open archives [3] and surprisingly also important for high profile publishers [4].

Thus, the aim of this paper is to compare the performances of SciDetect[2] – an open source program – with other detection techniques.

Section 2 gives a short description of fake paper generators based on PCFG and also provides an overview of different existing detection methods. Section 3 details detection approaches based on distance/similarity measurement. Section 4 presents a tuned classification process used by the SciDetect tool. Section 5 shows comparison results obtained by the different methods for fake paper detection. Section 6 concludes the paper and makes proposals for future work.

2 Fake Paper Generator and Detection

The field of Natural Language Generation (NLG) — a sub field of natural language processing (NLP) has flourished. The data-to-text approach [5] has been adopted for many useful real life applications, such as weather forecasting [6], review summarization [7], or medical data summarization [8]. However, NLG is also used in a different way as presented in section 2.1. While section 2.2 presents some of the existing detection methods.

[1] A fake researcher became one of the most highly cited author on Google Scholar, although all of his papers were automatically generated.

[2] http://scidetect.forge.imag.fr/

```
Many SCI_PEOPLE would agree that, had it not been for SCI_GENERIC_NOUN ...
SCI_BUZZWORD_ADJ SCI_BUZZWORD_NOUN and SCI_THING_MOD have garnered LIT_GREAT ...
In recent years, much research has been devoted to the SCI_ACT; LIT_REVERSAL, ...
SCI_THING_MOD and SCI_THING_MOD, while SCI_ADJ in theory, have not until ...
The SCI_ACT is a SCI_ADJ SCI_PROBLEM XXX
```

Fig. 1. Some possibles opening sentence of the introduction section (SCIgen PCFG).

2.1 Generators of Scientific Papers

The seminal generator SCIgen[3] was the first realization of a family of scientific oriented text generators: SCIgen-Physic[4] focuses on physics, Mathgen[5] deals with mathematics, and the *Automatic SBIR Proposal Generator*[6] (Propgen in the following) focuses on grant proposal generation. These four generators were originally developed as hoaxes whose aim was to expose "bogus" conferences or meetings by submitting meaningless, automatically generated papers.

At a very-quick glance, these types of papers appear to be legitimate with a coherent structure as well as graphs, tables, and so on. Such papers might mislead naive readers or an inexperienced public. They are created using PCFG – a set of rules for the arrangement of the whole paper as well as for individual sections and sentences (see Figure 1). The scope of the generated texts depends on the generator but they are typically quite limited when compared to a real human written text in both structure and vocabulary [9].

2.2 Fake Paper Detection

Some methods have been developed to automatically identify SCIgen papers. For example, [10] checks whether references are proper references, a paper with a large proportion of unidentified references will be suspected as being a SCIgen paper. [11] uses an ad-hoc similarity measure in which the reference section plays a major role whereas [12] is based on observed compression factor and a classifier. [13] in line with [4] propose to measure the *structural distance* between texts. [14] proposes a comparison of topological properties between natural and generated texts, and [15] studies the effectiveness of different measures to detect fake scientific papers. Our own study goes further on that track by including untested measures such as the ones used by ArXiv and Springer [3].

3 Distance and Similarity Measurements

In this paper, we are interested in measuring the similarity between documents as a way to identify specific ones as being automatically generated. Thus, we investigated four different measures: *Kullback-Leibler divergence*, *Euclidean distance*, *cosine similarity* and *textual distance*.

[3] http://pdos.csail.mit.edu/scigen/
[4] https://bitbucket.org/birkenfeld/scigen-physics
[5] http://thatsmathematics.com/mathgen/
[6] http://www.nadovich.com/chris/randprop/

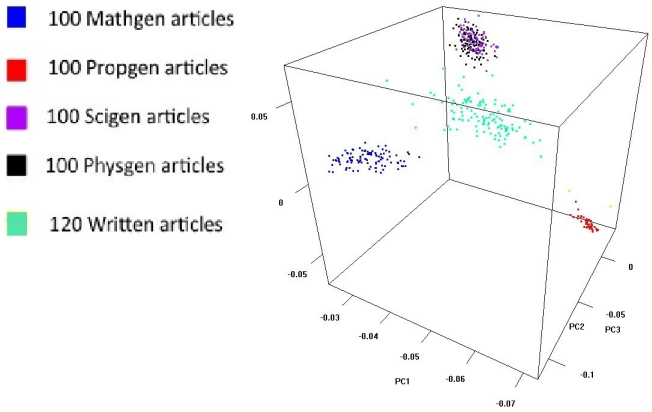

Fig. 2. principal-component analysis plots of stop-words in different corpora

In the following, for a text A of length N_A (number of tokens), let $F_A(w)$ denote the absolute frequency of a word w in A (the number of times word w appears in A) and $P_A(w) = \frac{F_A(w)}{N_A}$ be the relative frequency of w in A.

Kullback-Leibler divergence: this method measures the difference between two distributions. Typically, one under test and a *true* one. Thus it can be used to check the observed frequency distributions in a text against frequency distributions observed in generated text. With a text under test B and a set of true generated texts A, the (non-symetric) Kullback-Leibler divergence of B from A is computed as follows:

$$D_{KL}(A, B) = \sum_{i \in Sw} P_{A(i)} \log \frac{P_{A(i)}}{P_{B(i)}}$$

This approach (with Sw a set of stop words found in A) seems to be currently used by ArXiv. [3] shows a principal-component analysis plots (similar to Figure 2) where computer-generated articles are arranged in tight clusters well separated from genuine articles.

Euclidean Distance: each documents can be considered as a vector of absolute frequencies of all the words that appeared in it. Hence, the distance between two documents A and B is calculated as:

$$dE_{(A,B)} = \sqrt{\sum_{w \in A \vee B} (F_A(w) - F_B(w))^2}$$

While it is simple to compute, it is often regarded as not well suited for computing similarities between documents.

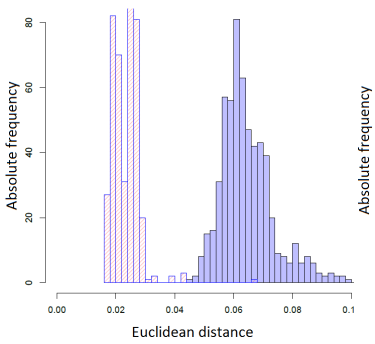

Fig. 3. Distribution of Euclidean distance to the nearest neighbour of generated text(red) and genuine text(blue)

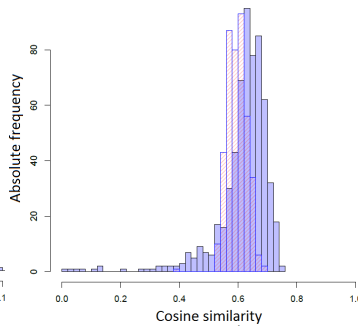

Fig. 4. Distribution of cosine similarity to the nearest neighbour of generated texts (red) and genuine texts (blue)

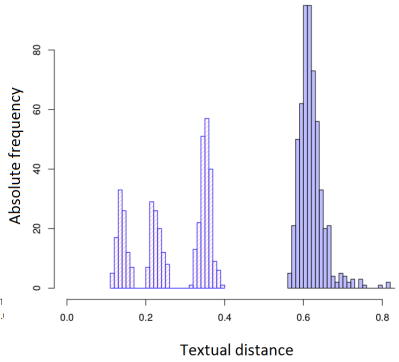

Fig. 5. Distribution of textual distance to the nearest neighbour of generated texts (red) and genuine texts (blue)

Cosine Similarity [16]: similar to euclidean distance, documents are considered as a vector of words frequencies. The cosine of the angle between them defines the cosine similarity:

$$dC_{(A,B)} = \frac{\sum_{w \in A \vee B} P_A(w) \times P_B(w)}{\sqrt{\sum_{w \in A \vee B} P_A(w)^2} \times \sqrt{\sum_{w \in A \vee B} P_B(w)^2}}$$

It is one of the most commonly used method in information retrieval to determine how similar two document are (often using $tf - idf$ instead of absolute/relative frequencies).

Textual Distance[4]: is a method to compute the differences in proportion of word tokens between two texts. The distance between two texts A and B where $N_A < N_B$ is:

$$d_{(A,B)} = \frac{\sum_{w \in A \vee B} |F_A(w) - \frac{N_A}{N_B} F_B(w)|}{2N_A}$$

where $d_{(A,B)} = 0$ means A and B share the same word distribution and $d_{(A,B)} = 1$ means there is no common word in A and B.

Figure 3 and 4 show the distribution of distances from a set of genuine texts (blue) to theirs nearest neighbour in a sample corpora of 400 texts from four generators as well as the distribution of distances from generated texts (red) to theirs nearest neighbour in the same sample of generated texts. These figures show that using Euclidean distance and cosine similarity might not be the best method. While with cosine similarity the two distributions tend to overlap heavily; Euclidean distance was able to somewhat separate the two distributions but with a really close margin. However, Figure 5 shows

that using textual distance creates a clear separation in distance to the nearest neighbour between 400 generated papers and genuine ones. This shows that the fake papers form a compact group for individual type of generator that are clearly separated from genuine texts (Scigen and Physgen were merged together because of their close relation). Thus, in the next section, we present our SciDetect system using textual distance and nearest neighbour classification with custom thresholds.

4 A Tool to Detect Automatically Generated Paper

In this section we present our SciDetect system, which is based on inter-textual distance using all the words and nearest neighbour classification. To avoid mis-classifications caused by text length, texts shorter than 10000 characters were ignored and texts longer than 30000 characters were split into smaller parts. To determine the genuineness of a text, we used different thresholds for each type of generator. We have performed a number of tests in order to set these thresholds.

For each generator (SCIgen, Physgen, Mathgen and Propgen) a set of 400 texts were used as test corpora (a total of 1600 texts). For each text, the distance to its nearest neighbour in the sample sets, which was composed of an extra 100 texts per generator (400 additional texts) was computed. The nearest neighbour was always of the same nature as the tested text; columns 1,2,3, and 4 of Table 1 show statistical information about the observed distances.

Along with that, to determine an upper threshold for genuine texts, a set of 8200 genuine papers from various fields were used. The nearest neighbour for each genuine text was computed using the same sample sets.

The first two rows of Table 1 show that, for a genuine paper, the minimal distance to the nearest neighbour in the sample set (0.52) is always greater than the maximal distance to the nearest neighbour of a fake paper (0.40).

	Scigen	Physgen	Mathgen	Propgen	Genuine
Min distance to NN	0.30	0.31	0.19	0.11	**0.52**
Max distance to NN	**0.40**	0.39	0.28	0.22	0.99
Mean distance to NN	0.35	0.35	0.22	0.14	0.69
Standard deviation	0.014	0.012	0.014	0.015	0.117
Median	0.35	0.35	0.22	0.14	0.64

Table 1. Statistic summary of textual distances between papers and their nearest neighbour (the nearest neighbour is always of the same kind).

By observing the results, we concluded that there would always be a close grouping of the generated texts that are separated from the group of real texts with a considerable gap in between. It is safe to say that we can classify the text based on thresholds. Thus, two thresholds for each generator were set: a lower threshold for generated papers based on the second row of Table 1 and an upper threshold for genuine papers (vary from 0.52 to 0.56 depending on the generator). Hence, a paper can be identified as possibly generated in two different ways. First if the distance is lower than the specific threshold

for a generated paper then it is considered as a confirmed case of generated. Second, if the distance is between the thresholds for generated and genuine paper, it is considered as a suspicious case.

5 Comparative Evaluation Between Different Methods

To thoroughly evaluate SciDetect and other methods, we decided to conduct a comparative test using different known methods

5.1 Test Candidates

Pattern Matching: Since automatically generated text has a very limited base of sentences, it is possible to believe that simply applying a pattern matching technique to scan a given document and report a specific score whenever a familiar pattern (a string of words) is encountered might work. In this research, we used a pattern matching tool that was developed and used internally at Springer that ranks the score as follows: For each detected phrase (string token) that matches a particular pattern the score is 10. If the phrase contains five to nine matching words, the score is 50, or 100 for phrases that have more than nine matching words. The final score is then compared with a threshold to determine whether the paper is automatically generated or not. If the score is less than 500, the paper is considered genuine; a score between 500 to 1000 is suspicious (it may be genuine or fake); and if the score is more than 1000, the paper is considered a fake.

This method might not be really reliable since the patterns can be easily modified. In addition, it is difficult to maintain and update the checker for a new type of generator for which the grammar is not available. Such approach is also quite sensitive to the length of the text: the longer the text:, the higher the chance that some specific pattern will appear.

Kullback-Leibler Divergence: As presented before, this method seems to be currently used by ArXiv. We implemented our own system that uses a list Sw of 571 stop-words [17] to classify texts. A profile for the average distribution of the stop word frequencies for each generator was created using the same 400 generated texts in the sample corpora of SciDetect. Two thresholds for each generator were also established in the same manner as in section 4 namely, a generate threshold for the maximum KL-divergence between a profile and a generated text from the test corpus; and a written threshold with the minimum KL-divergence between a profile and genuine written texts.

SciDetect We would also like to verify the usefulness of our SciDetect system as presented in Section 4.

5.2 Test Corpora

We used three different corpora to conduct the test:

- Corpus X: 100 texts from known generators (25 for each type of generator) without any modification.

method	corpus	True Positive		False Positive		True Negative	False Negative
		confirm	suspect	confirm	suspect		
Pattern Matching	X	25%	4%	0	0	0	71%
	Y	8%	16%	0	0	0	76%
	Z	0	0	0	0.01%	99.99%	0
Kullback-Leibler Divergence	X	87%	13%	0	0	0	0
	Y	79%	21%	0	0	0	0
	Z	0	0	0	1.65%	98.35%	0
SciDetect	X	100%	0	0	0	0	0
	Y	100%	0	0	0	0	0
	Z	0	0	0	0	100%	0

Table 2. Results of the different methods on the three corpora

- Corpus Y: 100 generated texts (25 from each generator) that have been modified by randomly changing a word every two to nine words with a word taken from a genuine research paper. The aim of this corpus is to test the robustness of these methods against not only pure generated texts but also modified versions which have some what different word distribution compared to the samples.
- Corpus Z: 10.000 real texts with a different length ranging from two pages to more than 100 pages.

5.3 Results

These experiments aim at determining the performance of the different methods for detecting generated papers. The results are shown in Table 2 whereby: true negative and true positive are respectively when a genuine paper or a generated paper is correctly identified and vice versa for false negative and false positive.

Close study of these results highlights several interesting aspects. Considering the current state of generators, current classifiers all work relatively well (all achieved a perfect precision rate). Difficult cases (Corpus Y) were marked as suspicious thus requiring further investigation. Particularly, SciDetect was proven as the most reliable method–all tests passed at 100%. Furthermore, despite the fact that pattern matching was designed to only match SCIgen patterns, it was able to recognize three papers from Scigen-Physic as suspected SCIgen; however, when applied to Corpus Y, one modified SCIgen paper was mistakenly listed as genuine. One case of a false positive in the pattern checker with Corpus Z was caused by a large file with more than 110 pages which triggered an out of memory error.

6 Conclusion

There is a need for automatic detection of computer generated papers in scientific literature. There are also several ways to accomplish this task. Among them, textual distance was demonstrated to provide the best results and this method was adopted in SciDetect. Furthermore, SciDetect was tested against pattern matching and Kullback-Leibler

divergence between stop-words. It proved to be the most reliable method for classification.

However, against other techniques of text generation like Markov chains, SciDetect and other current methods are impractical since they have an identical word's distribution rate as a human written paper and no fixed pattern. This calls for more in-depth research [9] such as checking the meaning of words [18], citation context[19] or evaluating sentence construction as well as the styles of generated texts [20].

Acknowledgments

This research was funded by Springer Nature. We would like to thanks our colleagues from PCM department of Springer Nature who provided valuable insights, expertise as well as test data that greatly assisted our research; especially to Jeff Iezzi for his continuous support throughout the process. Also to the reviewers who supply valuable criticisms of our work.

References

1. Labbe, C.: Ike Antkare one of the great stars in the scientific firmament. ISSI Newsletter **6**(2) (2010) 48–52
2. Beel, J., Gipp, B.: Academic search engine spam and google scholars resilience against it. Journal of Electronic Publishing (December 2010)
3. Ginsparg, P.: Automated screening: ArXiv screens spot fake papers. **- 508**(- 7494) (March 2014) – – 44
4. Labbé, C., Labbé, D.: Duplicate and fake publications in the scientific literature: How many scigen papers in computer science? Scientometrics **94**(1) (January 2013) 379–396
5. Labbé, C., Roncancio, C., Bras, D.: A personal storytelling about your favorite data. In: Proc. ENLG. (2015) 166–174
6. Reiter, E., Sripada, S., Hunter, J., Davy, I.: Choosing words in computer-generated weather forecasts. Artificial Intelligence **167** (2005) 137–169
7. Tien, M., Portet, F., Labbé, C.: Hypertext Summarization for Hotel Review. hal-01153598 (March 2015)
8. Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., Sykes, C.: Automatic generation of textual summaries from neonatal intensive care data. Artificial Intelligence **173** (2009) 789–816
9. Labbé, C., Labbé, D., Portet, F.: Detection of computer generated papers in scientific literature. (March 2015)
10. Xiong, J., Huang, T.: An effective method to identify machine automatically generated paper. In: Knowledge Engineering and Software Engineering. (2009) 101–102
11. Lavoie, A., Krishnamoorthy, M.: Algorithmic detection of computer generated text. arXiv preprint arXiv:1008.0706 (2010)
12. Dalkilic, M.M., Clark, W.T., Costello, J.C., Radivojac, P.: Using compression to identify classes of inauthentic texts. In: Proc. of the 2006 SIAM Conf. on Data Mining. (2006)
13. Fahrenberg, U., Biondi, F., Corre, K., Jégourel, C., Kongshøj, S., Legay, A.: Measuring global similarity between texts. In: Second International Conference, SLSP. (2014) 220–232
14. Amancio, D.R.: Comparing the topological properties of real and artificially generated scientific manuscripts. Scientometrics **105**(3) (December 2015) 1763–1779

15. Williams, K., Giles, C.L.: On the use of similarity search to detect fake scientific papers. In: Similarity Search and Applications - 8th International Conference, SISAP 2015. 332–338
16. Singhal, A.: Modern Information Retrieval: A Brief Overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering **24**(4) (2001) 35–42
17. Feinerer, I., Hornik, K., Meyer, D.: Text mining infrastructure in r. Journal of Statistical Software **25**(5) (3 2008) 1–54
18. Labbé, C., Labbé, D.: How to measure the meanings of words? amour in corneille's work. Language Resources and Evaluation **39**(4) (2005) 335–351
19. Small, H.: Interpreting maps of science using citation context sentiments: A preliminary investigation. Scientometrics **87**(2) (May 2011) 373–388
20. Kollmer, J.E., Pöschel, T., Gallas, J.A.: Are physicists afraid of mathematics? New Journal of Physics **17**(1) (2015) 013036