A Method for Plagiarism Detection over Academic Citation Networks

Sidik Soleman[1,a)] Atsushi Fujii

Abstract: The use of another person's words or ideas without giving credit, plagiarism, has become a crucial problem in academia. To alleviate this problem, a number of plagiarism detection methods have been proposed. Since the methods are formulated as finding a near-duplicate, measuring the document similarity is crucial. In this paper, we address this problem and propose a method by combining document content and citation behaviors. We also evaluate and discuss the effectiveness of our method.

1. Introduction

Reflecting the rapid progress of science, technology, and culture, an increasing number of academic publications have been recently available by means of digital libraries or general-purpose search engines on the Web. Whereas an academic publication should include the novel ideas proposed by the authors, most of the residue include known facts or knowledge in a large body of literature, for which citation provides a practical solution to indicate the source of each idea easily and also credit to the authors of each source.

This customary has resulted in a huge network in which each academic publication (i.e., document) and citation is represented as a node and a directed link between two nodes, which we shall call academic citation network (ACN). In practice, the entire ACN can be divided into more than one subnetwork, each of which roughly corresponds to a different discipline. However, we use only ACN to refer to both the entire and a partial ACN, without loss of generality.

Whereas in principle the authors who wish to borrow specific content from other documents are responsible for citing appropriate documents, in practice misconduct associated with missing or deceptive citations has of late become a crucial problem. Such conducts are generally termed "plagiarism" and is defined, for example, in the Merriam-Webster [*1] dictionary as "the act of using another persons words or ideas without giving credit to that person".

Plagiarism has a significantly negative impact on our society in terms of the following perspectives. First, it discourages the spirit of the invention and creativeness because the credit is not given to the right people. Second, the evaluation of each publication can purposefully be manipulated given that the frequency of a document being cited has been used to measure the achievement for a research project and intellectual contribution of individual researchers. Finally, plagiarism can decrease the trust of the academia. The above background has motivated us to explore plagiarism detection over the ACN.

A single case of plagiarism can generally be represented as "party X plagiarized document Q using one or more documents S_1,...,S_i,...,S_n, where X, Q, and S_i are variables representing a plagiarist, plagiarized document, and source document, respectively". Plagiarized documents, which refer to a resultant document, should not be confused with the source documents. In contrast, a task of plagiarism detection can be different depending on the purpose of a user. In the following, (1)-(2) are example scenarios for plagiarism detection associated with different resolution of analysis.

(1) To determine if a document in question is a plagiarized one, in which the input can potentially be non-plagiarized one.
(2) To find one or more source documents for each of the plagiarized ones as an evidence of plagiarism, in addition to (1).
(3) To identify how the fragment in a source document has been modified in the plagiarized one, in addition to (1) and (2).

Although it may also be important to determine whether a plagiarism is due to a deliberate intention or an innocent mistake, in this paper we focus only on intentional cases.

Because as in the general representation for plagiarism above, document Q usually consists of fragments of S_i $(1 \leq i \leq n)$ with optional modification, plagiarism detection has often been recast as detection for partial near-duplicate text in a document collection. Thus, a system for plagiarism detection can be realized with a straightforward application of information retrieval (IR), and more precisely the purpose is to search the document collection for one or more fragments resembling those in a document in question. Finally, the candidate documents whose similarity score or whose ranking in descending order of the similarity score is above a predetermined threshold are presented to the user. Systems for plagiarism detection that follows the IR approach gener-

[1] Graduate School of Information Science and Engineering, Department of Computer Science, Tokyo Institute of Technology, Ookayama 2-12-1, Meguro, Tokyo, Japan 152-8550
[a)] soleman.s.aa(at)mtitech.ac.jp

[*1] https://www.merriam-webster.com/dictionary/plagiarism

ally rely on the similarity between the plagiarized document and a candidate for its source document.

In this paper, we propose a method for plagiarism detection, focusing mainly on the computation for the similarity score between the contents of two documents. Our contribution is, unlike existing methods for plagiarism detection relying only on a single type of content similarity, to combine more than one type of document similarity between two documents in terms of the citation behaviors for those documents.

Section 2 surveys past research on plagiarism detection to clarify our focus and approach. Sections 3 and 4 elaborates on our method for plagiarism detection and evaluates its effectiveness experimentally, respectively. Finall, Section 5 concludes our work.

2. Related Work

In this section, we discuss the methods for plagiarism detection that formulate it as finding a near-duplicate. We categorize the methods into four types according to the compared aspects: (1) based on authorship attribution, (2) based on content, (3) based on citation, and (4) based on the combination of previously mentioned aspects.

Methods based on Authorship Attribution

Authorship attribution is a process for determining an author of a document. If a person plagiarizes a content from another document, the writing style of that content might be similar with the writing style of the source document.

When comparing two documents, the methods extract information related to writing styles of those documents to represent them. For example, Stamatatos [1] represented documents using n-gram based on stopwords such as *the, of, a, and, to*. They argued that stopwords could not easily be substituted because it is not likely to have synonym. The methods, however, may not be reliable if the length of plagiarized content is too short or the person who plagiarizes successfully modifies or changes the writing style of the plagiarized content.

Methods based on Content

In this category, the methods generally represent documents using bag-of-word [2], [3], [4], n-gram [5], [6], or fingerprinting [5]. Beside considering document as flat plain-text, some methods also consider content structure in documents when comparing them. Alzahrani et al. [2] considered word distribution in document sections for reweighting those words, while Soleman and Fujii [3] found that the disctinction between citing and non-citing sentence in documents is important during the document comparison.

Since content in plagiarized document may be modified, the methods in this category also utilize lexical dictionary to handle synonym, such as the method proposed by Chen et al. [4], and Chong and Specia [7].

Methods based on Citation

In the area of citation analysis, bibliographic coupling [8] is a well-known method to measure the similarity between documents with respect to citation link. Two documents are likely to be similar if they cite a lot of the same documents.

Motivated by the above method, Gipp and Meuschke [9] compared pattern of citation anchors between two documents for calculating their similarity score. Citation anchor is a symbol or character in text body that refers to a document in reference. While HaCohen-Kerner et al. [5] compared reference list in documents to measure the their similarity score.

Combination of the Existing Aspects

Recently, the methods for plagiarism detection measure the document similarity based on the combination of previously mentioned aspects. For example, Pertile et al. [10] combined citation and content-based method. While Sánchez-Vega et al. [11] combined authorship and content-based method. They found that combining more than one method with different aspect could improve the effectiveness of the plagiarism-detection methods.

3. Proposed Method

Since methods based on citation only consider citation anchor or reference to compute the similarity between two documents, in this paper, we propose a method that also considers the similarity between content around citation anchor. We argue that a plagiarized document is likely to have the same point of view with the source documents with respect to the cited document. It means that the plagiarized document may use the same content with the source document regarding to the same cited document.

Basically, our method computes the document similarity by measuring the content similarity in citing sentences that cite the same document. Later, we call this as *the similarity of citation behavior*. Here, citing sentence refers to sentence containing one or more citation anchors. Our method also considers the content similarity in non-citing part when computing the similarity of two documents because this part contains novel content. In following discussion, we refer this as *the similarity of novelty*. We assume that a plagiarized document would not provide a significant novel content in it.

Our proposed method is similar to the one proposed by Soleman and Fujii [3]. The difference is that their method does not consider whether citing sentences citing the same document or not.

Suppose we have two documents, e.g. x and y, our method linearly combines the similarity of citation behavior and the similarity of novelty to produce the similarity between those two documents (*simscore*) as shown in the following formula:

$$simscore(x, y, Z) = \alpha sim_{cb}(x, y, Z) + (1 - \alpha) sim_{novel}(x, y) \quad (1)$$

with

- Z: a set of documents cited by both x and y.
- α: a weighting value between 0 and 1. Thus, we could prioritize one of those similarity scores.

As we use linear combination, our similarity method produces score (*simscore*) between 0 and 1. The higher the score, the more likely x is a plagiarized document and y is the source one.

The similarity of citation behavior is calculated by summing the content similarity scores between citing sentences for the same documents cited by x and y, and then divided by the number of documents cited by x. If a cited document is associated with more than one citing sentence, those citing sentences are aggregated into a single text fragment. If a citing sentence contains more than one citation anchor, each document refered by those citation anchors is associated with that citing sentence.

The following equation describes the calculation of the similarity of citation behavior:

$$sim_{cb}(x, y, Z) = \frac{1}{|R_x|} \sum_{i=1}^{n} csim(cite(x, z_i), cite(y, z_i)) \quad (2)$$

with

- R_x: a set of documents cited by x.
- $csim$: textual content similarity between two text fragment.
- $cite(x, z_i)$: a function that aggregates a set of citing sentences by z_i in x into a single text fragment.
- $cite(y, z_i)$: a function that aggregates a set of citing sentences by z_i in y into a single text fragment.

The above formula produces score between 0 and 1. Additionally, the similarity score is not symmetrical. When the similarity score equals to 1, it means that all citing sentences in x are the same with the ones in y.

The next equation explains the calculation of the similarity of novelty:

$$sim_{novel}(x, y) = csim(ncite(x), ncite(y)) \quad (3)$$

with

- $ncite(x)$: a function that merges non-citing sentences in x into a single text fragment.
- $ncite(y)$: a function that merges non-citing sentences in y into a single text fragment.
- $csim$: textual content similarity between two text fragment.

To calculate $csim$, we use cosine similarity with bag-of-word model to represent both text fragments as vectors. The weight of each word is calculated as:

$$w_{t,c} = f_{t,c}\, log \frac{N}{n_t} \quad (4)$$

with

- $f_{t,c}$: total number of term t that appears in fragment c.
- N: total number of documents in document collection.
- n_t: total number of documents in document collection that contain term t.

Before transforming text fragments into their vector represensations, we also perform several pre-processing techniques. First, text fragment is lowercased, words that are considered as stopwords are removed [*2], and the remaining words are finally stemmed by using an English language stemmer [*3].

After transforming text fragments to their vector representations, e.g. u and v, the cosine similarity is calculated as:

$$csim(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||} \quad (5)$$

Table 1 Statistics of the datasets

Type	Detail
Topic	computation linguistics
Positive pair	41
Negative pair	52
Target document	4 685
Plagiarized document	40
Source/plagiarized document	1.025
Avg. word (target)	2 557.7
Avg. word (plagiarized)	2 797
Kappa	0.675 *(substantial)*
Agreement rate	84%

4. Experiments

4.1 Dataset

To evaluate our proposed method, we used datasets developed by Pertile et al. [10]. They investigated exhaustively two document collections, namely ACL [*4] and PubMed [*5] to create these datasets. For this experiment, however, we only used the dataset that is developed from ACL because these documents use more consistent reference and citation style than the ones in PubMed.

In the dataset creation, they performed pairwise comparisons between documents in the document collection by using some document similarity methods to select top-n pairs. Next, they asked 10 annotators to judge whether a pair is suspected as a plagiarism case because one of the documents in the pair reuses text from the other one. The annotation is based on the plagiarism level in IEEE [*6]. Thus, a pair of document consisting x and y is labeled as positive case if x is considered as having a significant text reuse from y. Thus, x is a plagiarized document and y is the source one. Although these positive pairs might be better to be addressed as suspected self-plagiarism, since both documents share some authors.

Since the document in the dataset is still in PDF format, we processed them and extracted citing sentences, references, and texts by using Grobid [12]. **Table 1** shows the complete information about this dataset.

4.2 Evaluation Scenario

As mentioned earlier, we have two options to use the similarity score ($SimScore$): (1) to rank candidate source documents and determine ranking cut-off before presenting the user, or (2) to select candidate source document if its similarity score exceeds a certain threshold.

In this paper, we address both issues. We call the first scenario as ranking task, while the second one as classification task. In the ranking task, we only use positive pairs and target documents. Thus, we can evaluate our method whether it can find the source documents or not.

In the classification task, we only use positive and negative pairs. Thus, we can evaluate whether our method could make distinction between a pair of plagiarized and source document and the one that is not as good as human does.

[*2] http://www.nltk.org/nltk_data/
[*3] http://www.nltk.org/howto/stem.html
[*4] http://aclanthology.info/
[*5] https://www.ncbi.nlm.nih.gov/pubmed/
[*6] https://www.ieee.org/documents/Level_description.pdf

4.3 Evaluation Methods

To evaluate the performance of our method, we use precision (P), recall (R), and $F1$. They are calculated by using these equations:

$$P = \frac{tp}{tp + fp} \tag{6}$$

$$R = \frac{tp}{tp + fn} \tag{7}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{8}$$

with

- tp: true positive, i.e. the number of retrieved source documents in the ranking task, or the number of correctly predicted positive pairs in the classification task.
- fp: false positive, i.e. the number of retrieved non-source documents in the ranking task, or the number of negative pairs predicted as positive ones in the classification task.
- fn: false negative, i.e. the number of source documents that are not retrieved in the ranking task, or the number of positive pairs predicted as negative ones in the classification task.

In ranking task, we also use Mean Average Precision (MAP) that is calculated by this following equation:

$$MAP(n) = \frac{1}{|D|} \sum_{d=1}^{|D|} \frac{1}{|sr_d|} \sum_{i=1}^{n} \frac{|\{sr_d \cap L_{d,i}\}|}{i} \tag{9}$$

with

- n = cut-off for a ranked list in the ranking task.
- sr_d = a set of source documents of a plagiarized one d.
- D = a set of plagiarized documents.
- $L_{d,i}$ = top-i documents in an output list of a plagiarized document d.

All above methods produce score between [0, 1]. The higher the score, the better the performance of the method is.

4.4 Baseline Method

In the experiment, we compare the proposed method with the baseline that compares two documents as bag-of-word. Thus, this method only use a single type of content similarity and does not consider the similarity of the citation behavior or novelty for those documents.

To compare two documents, the baseline applies the same preprocessing as the proposed method, i.e. lowercasing, stopword removal, and stemming. Then, those documents are transformed into document vectors using TFxIDF weighting as explained in the Equation 4. Lastly, those document vectors are compared using cosing similarity as described in Equation 5.

4.5 Results

4.5.1 Preliminary Investigation

Before conducting the full experiment, we investigated whether there is a significant different between the similarity of citing sentences for the same cited document in the positive and negative pair in the dataset. Thus, we calculate the content similarity of citing sentences by using Equation 5.

We found 195 and 220 total citing sentences for the same cited document from all positive and negative pairs, respectively. We found that the average similarity score of citing sentences for the same cited document in positive pair is higher than the one in negative pair. The average scores are 0.7306 and 0.5899 for positive and negative pair, respectively. We also conducted 2-tailed t test between these average scores and found that their different is significant at level 1%. This investigation result suggests that it is likely that plagiarized document maintains the content in citing sentences for the same cited documents with the source document.

4.5.2 Ranking Task

Table 2 shows the performance of the baseline and the proposed method in terms of P, R, and F1 for various cut-off from 10 to 1000. According to these results, the baseline and the proposed method have equal performance. At cuf-off equals to 10, the R almost achieves 1. It suggests that the source documents could be retrieved by only retrieving the top-10 documents.

Table 3 shows the MAP scores of the baseline and the proposed method at various cut-off. The baseline's performance is quite good. The MAP score closes to 1 means that most of the source documents are located at top of the document lists. In fact, there are only 4 out of 40 plagiarized documents, which the source document is not located at the top of document list. Thus, improving the baseline might not be easy. This happens because during the dataset creation, Pertile et al. [10] only pooled the document pairs that have high textual content overlap.

Our method for $\alpha \in [0.7, 0.8]$ slightly outperforms the baseline. Its MAP scores are 0.0062 higher than the baseline's score at all cut off. These results also suggest that the best α in the ranking task is [0.7, 0.8]. It means that we should prioritize the similarity of citation behavior. Although we should not ignore the similarity of novelty as well.

Additionally, our method improves the MAP scores for two plagiarized documents. In our method, the source documents for each plagiarized document are located at 1 and 2 in the document lists, while in the baseline, they are at 2 and 4, respectively.

However, our method also worsens the MAP scores for the other two plagiarized documents. In our method, the source documents for each plagiarized document are located at 2 and 42, while in the baseline, they are at 1 and 38.

4.5.3 Classification Task

In the detailed analysis, the methods are evaluated by means of leave-one-out cross-validation. During the training, we used F1 score as the cost function when finding the optimum *simscore* threshold. **Table 4** shows the P, R, and F1 of the baseline and the proposed method using various $\alpha \in [0, 1]$.

Our method with $\alpha = 0.4$ outperforms the baseline. It performs 0.0298 higher than the baseline. These results suggest that the best α in the classification task is 0.4. It means that we should slightly prioritize the similarity of novelty between two documents. This α is different with the one in the ranking task. Therefore, we should also tune the value of α depending on the tasks.

In this task, our method produces 5 more true positives and 4 more false positives than the baseline. This result still indicates

Table 2 P, R, and F1 of the baseline and the proposed method in the ranking task

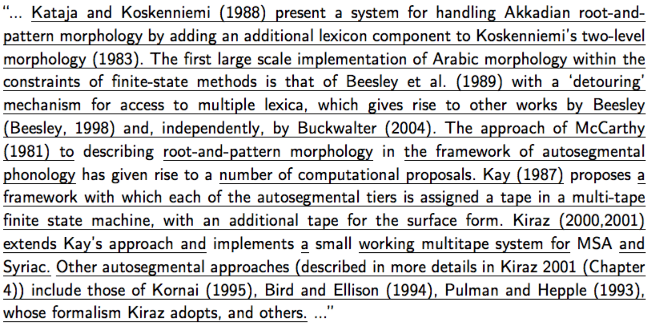
	P			R			F1		
Cut-off	Baseline	[0, 0.9]	1	Baseline	[0, 0.9]	1	Baseline	[0, 0.9]	1
10	0.1000	0.1000	0.0775	0.9750	0.9750	0.7500	0.1814	0.1814	0.1405
30	0.0333	0.0333	0.0267	0.9750	0.9750	0.7750	0.0644	0.0644	0.0516
100	0.0103	0.0103	0.0080	1.0000	1.0000	0.7750	0.0204	0.0204	0.0158
1000	0.0010	0.0010	0.0009	1.0000	1.0000	0.8750	0.0020	0.0020	0.0018

Table 3 MAP scores of the baseline and the proposed method in the ranking task

		The proposed method for α between 0 and 1					
Cut-off	Baseline	[0, 0.4]	0.5	0.6	[0.7, 0.8]	0.9	1
[10, 30]	0.9313	0.9167	0.9313	0.9333	**0.9375**	0.8833	0.6779
≥ 100	0.9319	0.9173	0.9319	0.9339	**0.9381**	0.8839	0.6801

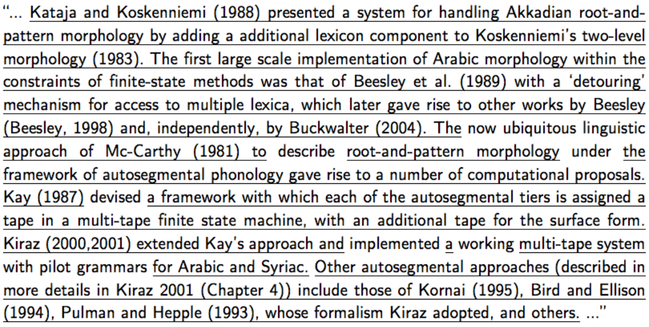
Table 4 P, R, and F1 of the baseline and the proposed method in the classification task

Method	P	R	F1
Baseline	0.7292	0.8537	0.7865
0	0.6724	0.9512	0.7879
0.1	0.7059	0.8780	0.7826
0.2	0.6610	0.9512	0.7800
0.3	0.7292	0.8537	0.7865
0.4	**0.7018**	**0.9756**	**0.8163**
0.5	0.7674	0.8049	0.7857
0.6	0.6667	0.9268	0.7755
0.7	0.5882	0.9756	0.7339
0.8	0.4118	0.6829	0.5138
0.9	0.5246	0.7805	0.6275
1	0.4409	1.0000	0.6119



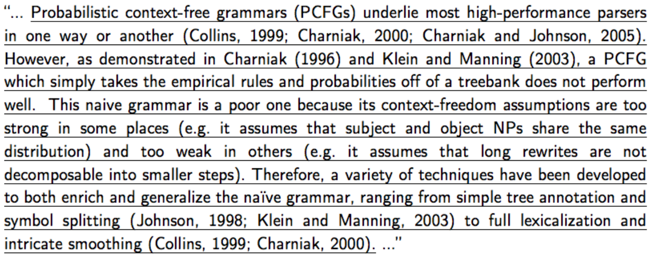
"... Kataja and Koskenniemi (1988) present a system for handling Akkadian root-and-pattern morphology by adding an additional lexicon component to Koskenniemi's two-level morphology (1983). The first large scale implementation of Arabic morphology within the constraints of finite-state methods is that of Beesley et al. (1989) with a 'detouring' mechanism for access to multiple lexica, which gives rise to other works by Beesley (Beesley, 1998) and, independently, by Buckwalter (2004). The approach of McCarthy (1981) to describing root-and-pattern morphology in the framework of autosegmental phonology has given rise to a number of computational proposals. Kay (1987) proposes a framework with which each of the autosegmental tiers is assigned a tape in a multi-tape finite state machine, with an additional tape for the surface form. Kiraz (2000,2001) extends Kay's approach and implements a small working multitape system for MSA and Syriac. Other autosegmental approaches (described in more details in Kiraz 2001 (Chapter 4)) include those of Kornai (1995), Bird and Ellison (1994), Pulman and Hepple (1993), whose formalism Kiraz adopts, and others. ..."

Fig. 1 A document excerpt quoted from [13] in page 4–5



"... Kataja and Koskenniemi (1988) presented a system for handling Akkadian root-and-pattern morphology by adding a additional lexicon component to Koskenniemi's two-level morphology (1983). The first large scale implementation of Arabic morphology within the constraints of finite-state methods was that of Beesley et al. (1989) with a 'detouring' mechanism for access to multiple lexica, which later gave rise to other works by Beesley (Beesley, 1998) and, independently, by Buckwalter (2004). The now ubiquitous linguistic approach of Mc-Carthy (1981) to describe root-and-pattern morphology under the framework of autosegmental phonology gave rise to a number of computational proposals. Kay (1987) devised a framework with which each of the autosegmental tiers is assigned a tape in a multi-tape finite state machine, with an additional tape for the surface form. Kiraz (2000,2001) extended Kay's approach and implemented a working multi-tape system with pilot grammars for Arabic and Syriac. Other autosegmental approaches (described in more details in Kiraz 2001 (Chapter 4)) include those of Kornai (1995), Bird and Ellison (1994), Pulman and Hepple (1993), whose formalism Kiraz adopted, and others. ..."

Fig. 2 A document excerpt quoted from [14] in page 2–3



"... Probabilistic context-free grammars (PCFGs) underlie most high-performance parsers in one way or another (Collins, 1999; Charniak, 2000; Charniak and Johnson, 2005). However, as demonstrated in Charniak (1996) and Klein and Manning (2003), a PCFG which simply takes the empirical rules and probabilities off of a treebank does not perform well. This naive grammar is a poor one because its context-freedom assumptions are too strong in some places (e.g. it assumes that subject and object NPs share the same distribution) and too weak in others (e.g. it assumes that long rewrites are not decomposable into smaller steps). Therefore, a variety of techniques have been developed to both enrich and generalize the naïve grammar, ranging from simple tree annotation and symbol splitting (Johnson, 1998; Klein and Manning, 2003) to full lexicalization and intricate smoothing (Collins, 1999; Charniak, 2000). ..."

Fig. 3 A document excerpt quoted from [15] in page 1

that our method is better than the baseline.

4.6 Error and Successful Cases

In the experiment, we conducted error investigation for the results from the classification task. Our method for $\alpha = 0.4$ could detect more true positives than the baseline is because the plagiarized document in a positive pair reuses a significant content of citing sentences from the source document. For example, **Fig. 1** and **Fig. 2** show text excerpts from a positive pair detected only by our method. By reading those texts, we could identify several citing sentences that are similar or identical between those documents. There are 4 of 5 true positives associate with this reason. While the remaining positive pair is detected by our method because one of the document in the pair reuses a significant content from methodology and evaluation section. In other words, it reuses a significant non-citing sentences, which is related to novelty from the other document.

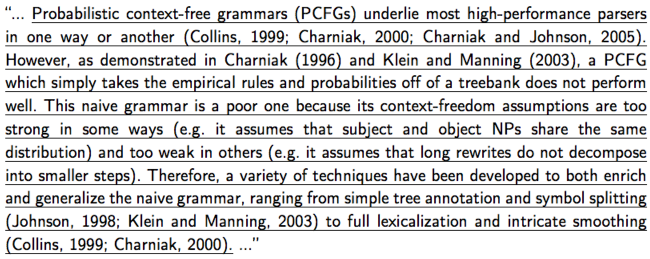
In the case of false positive, these errors happen because a single citing sentences are calculated multiple times because it has more than one citation anchors by our method. Thus, the similarity of citation behavior becomes quite significant. There are 3 false positives associated with this error type. For example, **Fig. 3** and **Fig. 4** show text excerpts from a negative document pair, which is predicted by our method as positive one. For the last false positive, we could not find the error reason for it.

Lastly, our method and the baseline also make the same false negative for a positive pair. The documents in this pair propose a similar method for different purpose or goal. As a result, the experiment results, key findings, or citing sentences are different. Thus, the documents has relatively less significant text overlap compared to other positive pairs. This indicates that it is still challenging to detect a plagiarism case for the one that has less text overlap.

5. Conclusion

In this paper, we propose a method to measure the similarity score of two documents for plagiarism detection. Unlike the existing methods, our method linearly combines two types of similarity: the similarity of citation behavior and the similarity of novelty.

For plagiarism detection, we use this similarity score to rank candidate source document or to decide whether a document is



"... Probabilistic context-free grammars (PCFGs) underlie most high-performance parsers in one way or another (Collins, 1999; Charniak, 2000; Charniak and Johnson, 2005). However, as demonstrated in Charniak (1996) and Klein and Manning (2003), a PCFG which simply takes the empirical rules and probabilities off of a treebank does not perform well. This naive grammar is a poor one because its context-freedom assumptions are too strong in some ways (e.g. it assumes that subject and object NPs share the same distribution) and too weak in others (e.g. it assumes that long rewrites do not decompose into smaller steps). Therefore, a variety of techniques have been developed to both enrich and generalize the naive grammar, ranging from simple tree annotation and symbol splitting (Johnson, 1998; Klein and Manning, 2003) to full lexicalization and intricate smoothing (Collins, 1999; Charniak, 2000). ..."

Fig. 4 A document excerpt quoted from [16] in page 1

a source document or not. According to our experiment, our method outperforms the baseline, which computes the similarity score between two documents by means of bag-of-word model to represent their textual contents. Our experiment also suggests that we should adjust the weighting parameter in our method depending on the usage of the similarity score. In this paper, we also discuss the error analysis and the examples of error and successful cases.

As for future work, it is important to evaluate our method using the actual or verified plagiarism case if this type of dataset is available. Additionally, it is also important to evaluate our method using the dataset with various type of plagiarism not only verbatim copy, which generally has a significant content overlap.

References

[1] Stamatatos, E.: Plagiarism Detection Based on Structural Information, *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, Glasgow, Scotland, UK, pp.1221–1230 (2011).

[2] Alzahrani, S., Palade, V., Salim, N., and Abraham, A.: Using Structural Information and Citation Evidence to Detect Significant Plagiarism Cases in Scientific Publications, *Journal of the American Society for Information Science and Technology. Wiley Subscription Services, Inc., A Wiley Company*, Vol.63, No.2, pp.286–312 (2012).

[3] Soleman, S., Fujii, A.: Plagiarism Detection Based on Citing Sentences. *Proceeding of 21st International Conference on Theory and Practice of Digital Libraries*, Thessaloniki, Greece, pp.485–497 (2017).

[4] Chen, C. Y., Yeh, J. Y., and Ke, H. R.: Plagiarism Detection Using ROUGE and WordNet. *Journal of Computing* Vol.2, No.3, pp.34–44 (2010).

[5] HaCohen-Kerner, Y., Tayeb, A., and Ben-Dror, N.: Detection of Simple Plagiarism in Computer Science Papers, *Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics*, Beijing, China, pp.421–429 (2010).

[6] Barrón-Cedeño, A. and Rosso, P.: On Automatic Plagiarism Detection based on N-grams Comparison, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* Vol.5478 LNCS, pp.696-700 (2009).

[7] Chong, M. and Specia, L.: Lexical Generalisation for Word-level Matching in Plagiarism Detection, *International Conference Recent Advances in Natural Language Processing*, Hissar, Bulgaria, pp.704–709 (2011).

[8] Kessler, M.M.: Bibliographic Coupling between Scientific Papers, *American Documentation* Vol.14, No.1, pp.10–25 (1963).

[9] Gipp, B. and Meuschke, N.: Citation Pattern Matching Algorithms for Citation-based Plagiarism Detection: Greedy Citation Tiling, Citation Chunking and Longest Common Citation Sequence, *Proceedings of the 11th ACM Symposium on Document Engineering (DocEng'11). ACM*, Mountain View, California, USA, pp.249–258 (2011).

[10] Pertile, S.D.L., Moreira, V.P., and Rosso, P: Comparing and Combining Content-and Citation-based Approaches for Plagiarism Detection, *Journal of the Association for Information Science and Technology. Association for Information Science and Technology*, Vol.67, No.10, pp.2511–2526 (2016).

[11] Sánchez-Vega, F., Villatoro-Tello, E., Montes-y-Gómez, M., Rosso, P., Stamatatos, E., and Villaseñor-Pineda, L.: Paraphrase Plagiarism Identification with Character-level Features, *Pattern Analysis and Applications*, pp.1-13 (2017)

[12] Lopez, Patrice: GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications, *Proceeding of 13th European Conference on Digital Libraries*, Corfu, Greece, pp.473–474 (2009).

[13] Habash, N., Rambow, O., and Kiraz, G.: Morphological Analysis and Generation for Arabic Dialects, *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, Michigan, USA, pp.17–24 (2005)

[14] Habash, N. and Rambow, O.: MAGEAD: A Morphological Analyzer and Generator for the Arabic Dialects, *Proceedings of the 21st International Conference on Computational Linguistics*, Sydney, Australia, pp.681–688 (2006)

[15] Petrov, S., Barrett, L., Thibaux, R., and Klein, D.: Learning Accurate, Compact, and Interpretable Tree Annotation, *Proceedings of the 21st International Conference on Computational Linguistics* Sydney, Australia, pp.433–440 (2006).

[16] Petrov, S., and Klein, D.: Parsing German with Latent Variable Grammars, *Proceedings of the ACL-08: HLT Workshop on Parsing German (PaGe-08)*, Columbus, Ohio, USA, pp.33–39 (2008).