Plagiarism Detection Based on a Novel Trie-based Approach

Alireza Talebpour
Department of Computer Engineer
Faculty of Computer Science and Engineering
Shahid Beheshti University
Tehran, Iran
Talebpour@sbu.ac.ir

Mohammad Shirzadi Laskoukelayeh
CyberSpace Research Institute
Shahid Beheshti University
Tehran, Iran
m.shirzadi@email.kntu.ac.ir

Zahra Aminolroaya
CyberSpace Research Institute
Shahid Beheshti University
Tehran, Iran
z.aminolroaya@Mail.sbu.ac.ir

ABSTRACT

Nowadays, plagiarism detection becomes as one of major problems in the text mining field. New coming technologies have made plagiarisation easy and more feasible. Therefore, it is vital to develop automatic system to detect plagiarisation in different contents.

In this paper, we propose a trie to compare source and suspicious text documents. We use PersianPlagDet text documents as a case study. Both character-based and knowledge-based techniques for detection purposes have improved our method. Besides, our fast algorithm for insertion and retrieval has made possible to compare long documents with high speed.

CCS Concepts

•Computing methodologies → Natural language processing; •Information systems → *Information systems applications;*

Keywords

Plagiarism detection; Trie-based method, Text mining

1. INTRODUCTION

Plagiarism means trying to pass off somebody else's words as your own [14]. Plagiarism detection is the process of locating text reuse within a suspicious document [5]. Nowadays, with the advent of technologies like the internet and the growth of digital content creation, plagiarism, especially in the format of text from existed contents, becomes a growing problem and one of the major problems in the text mining field. For example, plagiarism as a way to release the pressure to publish papers pushes down the quality of scientific papers. In [7], Lesk declares that, in some countries, 15% of submissions to arXiv contain duplicated materials and are plagiarized. Due to these problems, it is urgent to provide a system to automatically detect plagiarism and validate them.

There have been many approaches proposed based on lexical and semantic methods. On the one hand, the plagiarisation problem could be reduced to the problem of finding exact matched phrases, and, on the other hand, it could be as hard as finding restated phrases. Due to what a problem asked, different knowledge-based or character-based techniques could be applied. One of the lexical database for knowledge-based approach is the wordnet database. In this database different words are grouped together based on their cognitive synonyms[4]. This database could be used to find restated phrases. Words in different locations in sentences may have different applications, so knowing syntactic category (POS) of the words *i.e.* noun, verb, etc. could simplify the problem of plagiarism detection.

Plagiarized documents can be in any languages which need different policies to be detected due to different semantics and grammars. In this paper, we have proposed a novel approach for the PAN FIRE Shared Task of Persian plagiarism detection in the international contest PersianPlagDet 2016. We have used a hybrid method considering both character-based and knowledge-based approaches. A Persian wordnet database, Farsnet, is considered as our knowledge database [13]. Besides, we have applied POS tagging by using HAZM package [1]. By finding nouns and their synsets from the Farsnet, we could more precisely save and retrieve suspicious words from our proposed tree structure. In our plagiarism detection methodology, we have applied a novel extended prefix tree *i.e.* trie to store and retrieve documents. We not only consider the task of text plagiarism detection but also the algorithm computation time as an important factors.

1.1 Related works

There are many studies to find solutions for the problems of plagiarism detection and document matching. In the Nineties, studies on copy detection mechanisms of digitalized documents have led to computerized detecting plagiarism [16]. By the growth of generated data, the speed of plagiarism detectors has become an important criterion. in [8], a parameterized backward trie matching is considered as a fast method for the problem of source and suspicious documents alignment.

The plagiarism detection problem is also studied in different languages. For Persian language plagiarism detection, In [9], after preprocessing the source and suspicious documents, different similarity measurements like "Jaccard similarity coefficient", "Clough & Stevenson metric" and "LCS" are used for similarity comparisons between source and suspicious documents. Also, by applying FarsNet, Rakian *et.al* propose an approach, "Persian Fuzzy Plagiarism Detection (PFPD)", to detect plagiarized cases [17].

Our fast trie-based approach is proposed for the problem of the persian language plagiarisation detection. We describe the problem data and data preprocessing applied to

documents in section 2. Then, in section 3, the novel approach for plagiarism detection is described, and, in section 4, algorithm evaluation measurement is described, and our approach is evaluated. Finally, the results are concluded in section 5.

2. DATA

The data is a set of suspicious and source text documents released by PersianPlagDet competition [2]. In PersianPlagDet data, the document plagiarisms could happen in different ways: parts of a source text document could exactly being copied into a suspicious text, parts of a source text document with some random changes could being copied into a suspicious text, and parts of a restated source text document could be seen in a suspicious text.

2.1 Data preparation

Before applying plagiarism detection method, the source and suspicious text documents should be prepared. We explain the processes needed before plagiarism detection step by step:

Text tokenization and POS tagging

We tokenize text documents into words. Tokenization is the procedure of splitting a text into words, phrases, or other meaningful parts, namely tokens [3]. In addition to tokenization, the exact position of tokens, word offsets, are stored. A token offset represents the token character-based distance from the beginning of the document. By applying the Hazm POS tagger, we also specify part-of-speech of each word. The nouns are important for us, and help us to compare phrases for plagiarism detection purpose. Thus, nouns are flagged for the next stages of processing.

Text cleansing and normalization

First, we normalize text documents. Normalization is the task of transforming text characters into a unique and normal form of a language. For example, we convert all Arabic "yaa" and "Kaaf" to Persian "ye" and "Kaaf" for preprocessing Persian text documents, and we unify all numbers with different Persian and English unicodes. Punctuations are also removed from text documents.

Removing stop words and frequent words

Stop words are also removed from text data. Stop words are words which are moved out from text data in processing steps because they do not contain significant information. First, a group of stop words has been selected which an expert has proposed. Then, frequent words are also chosen and removed with considering a frequency threshold value.

Stemming words

The next step is to specify words stems. There are many kinds of words inflections and derivations. The suffices "haa", "aan", "yaan", "aat", "ien" and somtimes "gaan" could make a single word plural. We remove these suffices from nouns. Also, Arabic broken plurals are the most challenging kinds of noun pluralization which cannot be distinguished by removing some suffices. An expert has provided the words stems by the help of Dehkhoda and Moein dictionaries which could help us to convert Arabic broken plural nouns to singular ones.

Acquiring words synsets

After defining words part-of-speech, we search through the Farsnet to find the nouns cognitive synonyms, synsets. We find synsets because words may have be used instead of their synonyms in different positions. For example, "computers" may be used as "estimators" or "data processors". Like the words offsets, the synsets offsets are stored. Notice that the synset offsets are equal to the original words offsets.

To solve plagiarism problem, offsets specification of word tokens and also collecting noun words synsets and words stems are basic satellite data to be used in our proposed tree model, trie, which is explained in the next section.

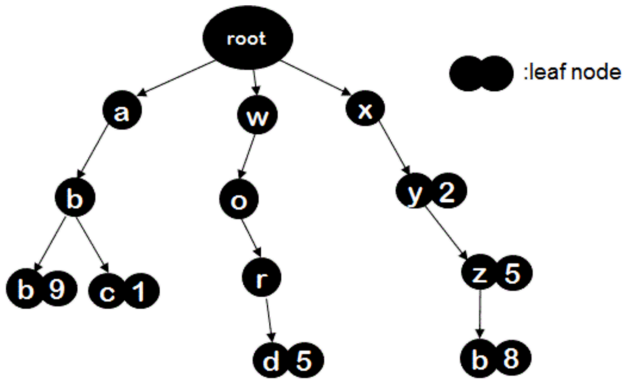
3. METHODOLOGY

After source and suspicious documents have been preprocessed, we use a method to find similar fragments and their exact offsets in both suspicious and source files. Before source and suspicious documents being compared, documents are saved to and retrieved from a trie data structure. In the next subsections, there would be a brief survey of trie trees and an explanation of our new proposed trie.

3.1 Brief survey of trie trees

A tokenized document is a set of words which can be stored in a dictionary. A trie data structure can be used to insert and find words in a dictionary in $O(n)$, n represents a single word length. The word "***trie***" is actually comes from the "re***trie***val" which is its usage. In the trie tree, the prefix tree, each node is a word or a prefix. All prefix characters of a word are inserted as a node, and the last letter is flagged as the word end in trie. Trie trees could have $< key, value >$ data structure. Words with similar prefixes may have similar subpaths. As an example brought in Figure 1, the word "xy" value is "2". Besides, "xy" and "xyzb" words have similar subpath. The node values are defined based on the problem. In the following, we describe the proposed trie and different key values.

Figure 1: An example of trie data structure[15].



3.2 Proposed trie trees

In this paper, we use trie data structure to insert and retrieve documents words due to trie properties *i.e.* fast insertion and searching and its high adjustment to our problem

solving *i.e.* defining the offsets of plagiarized strings. Our method for plagiarism detection is divided into two different processes:

Inserting documents to data structures

After preprocessing both source and suspicious documents, all the words with their exact positions in the source document are inserted into trie, and the suspicious words are added into an ordered list based on their position in the document. According to the trie definition, each trie node is a part of preprocessed words. In the proposed trie, each word has a "word positions" list which includes the word occurance positions in the documents. Notice that the words may have occurred in different positions in the document, but they are only inserted once in the trie, and their occurance positions are added in the words positions lists. Also, the words positions lists are only considered for the nodes which represent the last character of words. The more repeated words include in the suspicious document, the faster the trie can be constructed.

Due to enhancing searching speed, It is better to save the longer document in the trie, however we always save the source doument in the trie for simplicity.

Finding the longest plagiarized fragments

To report plagiarized sections, it is important to find similar words based on their sequential occurrences in the source and suspicious documents. The contiguity of words in suspicious documents could simply be kept based on the applied data structure *i.e.* ordered list. For the source documents, the words positions lists added in the trie will help us to find the order of words in the source plagiarized sections.

After constructing documents data structure, the longest plagiarized fragments should be found in both source and suspicious documents. Thus, we iterate over the suspicious document words one by one and find the corresponding words in source trie. It is obvious that finding words in trie contribute to obtaining the word positions in the source document. Also, the detected plagiarized positions of the suspicious document are added to the corresponding word "suspicious positions" list in the trie.

When a similar word is found in both documents the information of the word front and rear words in source document in also kept in the trie:

consider $W_p = \{w_{p_1}, w_{p_2}, ..., w_{p_n}\}$ is the list of suspicious words in ordered list and $W_s = \{w_{s_1}, w_{s_2}, ..., w_{s_m}\}$ is the ordered list of source words inserted in the trie. Where n is the number of words in the suspicious document and m is the number of distinct words in the source one. If $w_{p_1} = w_{s_1}$ and $w_{p_2} = w_{s_2}$, then "w_{s_2}", "w_{s_2} position in the source" and "w_{p_2} position in the suspicious" are added as the w_{s_1} front node "value", "words position list" and "suspicious position list". Moreover, w_{s_1} is added as the first of a sentence into the "sentence list". The process is also correct for rear nodes.

Traversing the suspicious document thoroughly leads to generating a set of linked lists helping to find the plagiarized fragments. The "sentence list" includes the first of plagiarized sections. By looking at the first of sentences in the sentence list and finding them in the trie, all the plagiarized fragments could be found.

Adding both the exact word and its synonyms (with the help of Farsnet) to the trie would cause to find the potential similar sections which are plagiarized by restatement. For example, for a potential copied phrase $P = \{w_1, w_2, ..., w_5\}$ in source and suspicious documents, if the synonym of w_2 were used instead of w_2, both w_2 and its synonym are added to the trie.

Furthermore, if $w2$ were deliberately added or deleted from the suspicious document, our plagiarism detector system could detect the plagiarized section P correctly. This feature is achieved because of the nature of the linked lists which we could trace the front and rear of words with. According to different POS in sentence, the words can be weighted differently for being added to removed intelligently.

Table 1: The evaluation of our approach on the test data released by PersianPlagDet 2016 contest [6].

Measure	Plagdet	Granularity	Precision	Recall
Value	0.775	1.228	0.964	0.837

4. EVALUATION

We use macro-averaged precision and recall, granularity measurements, and the plagdet score described in [11]. The precision and recall measurements evaluate the performance of detection in character level, while granularity considers the contiguity of text plagriasied phrases detected in source and suspicious documents. The granularity of detections R under true plagiarisms S is described as below [11];

$$gran(S, R) = \frac{1}{|S_R|} \sum_{s \in S_R} |R_S| \tag{1}$$

Where $S_R \subseteq S$ are the cases which are detected by detections and $R_S \subseteq R$ are the detections by considering s:

$$S_R = \{s | s \in S \wedge \exists r \in R : r \quad detects \quad s\},$$
$$R_S = \{r | r \in R \wedge r \quad detects \quad s\}.$$

Plagdet score is an overall score which considers the other mentioned measurements. The Plagdet score overall score is as follows [11];

$$plagdet(S, R) = \frac{F_\alpha}{\log_2(1 + gran(S, R))} \tag{2}$$

In which S and R are detections and true cases of a plagiarism and F_α is $F_\alpha - Measure$, the weighted harmonic mean of precision and recall which can be defined as bellow;

$$F_\alpha = (1 + \alpha^2).\frac{precision.recall}{(\alpha^2.precision) + recall} \tag{3}$$

If α is not predefined, we consider $\alpha = 1$.

Table 1 shows the evaluation of our approach on the test data released by PersianPlagDet 2016 competition which is based on TIRA and the PAN evaluation setup [10, 18, 12]. Our approach high precision, recall and acceptable granularity values contribute to admissible plagdet score.

5. CONCLUSIONS

The advents of digitalization and technology have simplified the act of plagiarizing. Thus, it is crucial to develop an automatic systems to detect plagiarisation in different contents.

We first prepared the text data released by international PersianPlagDet 2016 contest. We made the data ready by preprocessing, tokenization and Morphological analysis (e.g. POS tagging) before documents comparison. In this paper, we have proposed a novel trie-based approach to save and retrieve source and suspicious preparation documents for solving the plagiarism detection problem. Fast inserting and retrieval long sentences were our reasons to exploit trie trees structures for the detection problem. Both finding noun words and their synsets with saving them to our extended trie have helped us to improve our text comparison especially in the case of restatement phrase matching.

To evaluate our algorithm, we used macro-averaged precision and recall, granularity measurements, and the plagdet score which were proposed by the PersianPlagDet competition. High precision, recall and acceptable granularity made the overall plagdet score for our algorithm admissible. Besides, thanks to the help of our proposed trie, large documents can be compared for the purpose of plagiarism detection.

In the next study, we will work on the contiguity of text plagriasied phrase for better granularity results. Besides, we will consider other part-of-speech synsets like verb synsets to improve our algorithm performance.

References

[1] 2013. Hazm. https://github.com/sobhe/hazm. (2013).

[2] 2016. PersianPlagDet 2016. http://www.ictrc.ac.ir/plagdet. (2016).

[3] Uysal A. K. and Gunal S. 2014. The impact of preprocessing on text classification. *Information Processing & Management* 50, 1 (2014), 104–112.

[4] Miller G. A. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[5] Asghari H., Khoshnava K., Fatemi O., and Faili H. 2015. Developing Bilingual Plagiarism Detection Corpus Using Sentence Aligned Parallel Corpus. *Notebook for PAN at CLEF* (2015).

[6] Asghari H., Mohtaj S., Fatemi O., Faili H., Rosso P., and Potthast M. 2016. Algorithms and Corpora for Persian Plagiarism Detection: Overview of PAN at FIRE 2016. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation (CEUR Workshop Proceedings)*. CEUR-WS.org.

[7] Lesk M. 2015. How many scientific papers are not original? *Proceedings of the National Academy of Sciences* 112, 1 (2015), 6–7.

[8] Mozgovoy M. 2007. *Enhancing computer-aided plagiarism detection.* University Of Joensuu.

[9] Mahmoodi M. and Varnamkhasti M. M. 2014. Design a Persian Automated Plagiarism Detector (AMZPPD). *arXiv preprint arXiv:1403.1618* (2014).

[10] Potthast M., Stein B., Barrón-Cedeño A., and Rosso P. 2010. An Evaluation Framework for Plagiarism Detection. In *23rd International Conference on Computational Linguistics (COLING 10)*, Chu-Ren Huang and Dan Jurafsky (Eds.). Association for Computational Linguistics, Stroudsburg, Pennsylvania, 997–1005.

[11] Potthast M., Stein B., Barrón-Cedeño A., and Rosso P. 2010. An evaluation framework for plagiarism detection. In *Proceedings of the 23rd international conference on computational linguistics: Posters*. Association for Computational Linguistics, 997–1005.

[12] Potthast M., Gollub T., Rangel F., Rosso P., Stamatatos E., and Stein B. 2014. Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)*, Evangelos Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms (Eds.). Springer, Berlin Heidelberg New York, 268–299. DOI:http://dx.doi.org/10.1007/978-3-319-11382-1_22

[13] Shamsfard M., Hesabi A., Fadaei H., Mansoory N., Famian A., Bagherbeigi S., Fekri E., Monshizadeh M., and Assi S. M. 2010. Semi automatic development of farsnet; the persian wordnet. In *Proceedings of 5th Global WordNet Conference, Mumbai, India*, Vol. 29.

[14] Lea M. R. and Street B. 2014. understanding textual practices in higher education. *Writing: Texts, processes and practices* (2014), 62.

[15] More N. 2015. Trie Data Structure. http://www.ideserve.co.in/learn/trie-insert-and-search. (2015).

[16] Brin S., Davis J., and Garcia-Molina H. 1995. Copy detection mechanisms for digital documents. In *ACM SIGMOD Record*, Vol. 24. ACM, 398–409.

[17] Rakian Sh., Esfahani F. S., and Rastegari H. 2015. A Persian Fuzzy Plagiarism Detection Approach. *Journal of Information Systems and Telecommunication (JIST)* 3, 3 (2015), 182–190.

[18] Gollub T., Stein B., and Burrows S. 2012. Ousting Ivory Tower Research: Towards a Web Framework for Providing Experiments as a Service. In *35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12)*, Bill Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson (Eds.). ACM, 1125–1126. DOI:http://dx.doi.org/10.1145/2348283.2348501