VNU Journal of Science, Mathematics - Physics 27 (2011) 55-62

A unified plagiarism detection framework

Nguyen Xuan Toi[1,*] , Nguyen Viet Hung[2], Pham Bao Son[3,4]

[1]*People Security Academy*
[2]*Ministry of Public Security*
[3]*Information Technology Institute, Vietnam National University, Hanoi*
[4]*University of Engineering and Technology, Vietnam National University, Hanoi*

Received 20 January 2011

Abstract. With the rapid growth of information technology, Internet and digital libraries have been developing so fast that illegal copying of documents is becoming easier and more popular. A challenging question is how to identify documents with similar content which are candidate of plagiarism. There are several approaches for estimating the similarity between two documents and each has its own advantages and disadvantages. An approach may be effective in one domain but may not work in others. In this paper, we propose a unified plagiarism detection framework that can identify which approach works most effectively in a new domain. Experimental results on three different corpora for different languages have demonstrated the effectiveness of our approach.

Keywords: plagiarism detection, copied documents, similarity.

1. Introduction

Recently, with advances in technology, Internet and Digital libraries have provided users with easier access to online digitized news, articles, magazines, books and other information of interest. Word processors also become more sophisticated and faster. In this environment, users may cut and paste, modify existing documents from a lot of different sources and redistribute the information without permission much easier.

A challenging question is how to give users access to a lot of digital libraries and different sources and to protect our original documents at the same time. In this paper, we attempt to address this problem by providing a mechanism to identify documents with similar content, which are candidates of plagiarism.

A number of methods have been proposed to address this problem. However, it is very difficult to decide which is the best algorithm or the best tool as each has its own advantages and disadvantages.

[*] Corresponding author: Tel.: +84913.373.118
E-mail: talongc500@yahoo.com

One approach with a particular set of parameter values may be effective in one domain but may not be effective in another domain. So how do we automatically identify the most effective method in a new domain?

In this paper, we propose a unified plagiarism detection framework that can automatically identify which approach with corresponding parameter values are the most effective in a given domain. Three popular methods are used in the framework, which are Overlap, Cosine and Greedy String tiling (GST) methods.

Furthermore, we would like to apply the framework to Vietnamese in particular. As word segmentation in Vietnamese is different to English we will investigate the impact of word segmentation in detecting plagiarism for Vietnamese documents.

The rest of the paper is organized as follows. In Section 2, we present some related works. The design and implementation of our system will be described in Section 3. The result of our experiments will be discussed in Section 4. Finally we will conclude with pointers to future work in Section 5.

2. Related works

In this section, we are going to discuss the existing related literature and background research on methods for measuring the similarity between documents as well as plagiarism detection systems.

2.1 Comparison methods

There are several methods that have been used for similarity measure in plagiarism detection. In this section, we discuss and describe three of such popular methods namely: Overlap, Cosine and GST.

2.1.1 Chunk

In the Overlap and Cosine methods, each document is split into chunks before it is compared with other documents. In English, a word is a syllable but in Vietnamese, a word may contain one or more successive syllables. So a chunk may be an n-gram of syllables or n-grams of words. For example, in this following sentence: "Xử lý ngôn ngữ tự nhiên là một lĩnh vực rất khó" if a chunk is a word in Vietnamese then "xử lý", "ngôn ngữ”,“tự nhiên", "là", "một", "lĩnh vực", "rắt", "khó" are chunks of the above sentence. A list of chunks is a potential representation for the content of the document and we can measure the similarity between documents by comparing their corresponding chunks.

2.1.2 Overlap method

Overlap method is one of the methods to measure the similarity of documents. This measure is used popularly in IR system. When the user gives a query to the system, the system will search in its database to look for documents that are most similar to the query.

Overlap measure between documents A with B is the quotient of the number of chunks that appear in both documents and the smaller value of the total numbers of chunks in A and B.

$$S(A,B)=\frac{A\cap B}{\min(\|A\|,\|B\|)}$$

This measure has its value range between 0 and 1. It indicates the proportion of the number of shared chunks in the shorter document.

2.1.3 Cosine similarity measure

Another popular method to measure the similarity is Cosine measure. Given two vectors of attributes, A and B, the cosine similarity, θ, is computed as:

$$S(A,B)=\frac{A\square B}{\|A\|\|B\|}$$

The attribute vectors A and B are usually the term (chunk) frequency vectors of the documents.

Essentially, S(A,B) is the cosine value of the angle which is created between two chunk vectors in the k dimension space. If the angle is small, this means that the value of S(A,B) will be large and the similarity of the two documents is high.

2.1.4 Greedy String tiling

This algorithm aims at identifying the longest possible common strings between two documents. We can measure the similarity of two document based on these longest common strings [1].

While the two above algorithms (Overlap and Cosine) use fixed-length chunk this algorithm uses variable-length chunk. We conjecture that using variable-length chunk may be better than the fixed length chunk in some particular domains.

2.2 Some Plagiarism Detection Systems

There are several plagiarism detection systems such as CHECK [2], COPS [3], SCAM [4], YAP3 [5]. In these systems, they often use only one predetermined kind of chunk (sentence chunk or word chunk) and one comparison method. As different systems report their best results on different domains, the comparison methods and the type of chunk used in each system are different. It indicates that different domain may require different methods with different chunk types. This is the goal of our framework to automatically identify the most effective method and chunk type for a given domain.

3. System Architecture

We have built a unified plagiarism detection framework with its architecture shown in Figure 1.

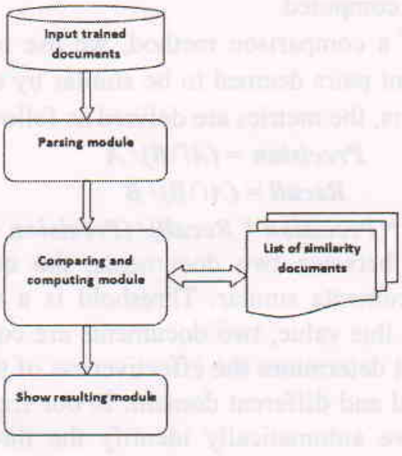

Fig. 1. Architecture of system.

As shown in the diagram, the system is composed of three main modules: Parsing module, Comparing module and Resulting showing module.

The inputs of the system are a set of documents representing the target domain and a list of similar documents pairs. The definition of plagiarism is implicitly encoded in the list of similar documents pairs. Outputs of the system are the method with corresponding parameter values that are deemed most effective in identifying plagiarism in the given domain.

3.1 Parsing module

The current version of the system works with Unicode plain text. The parsing module consists of two steps:

Step 1: This is the preprocessing step before a document is split into chunks. All punctuations such as commas, semi-colons are removed and all characters are converted into lower characters.

Step 2: In this step, all chunks with its frequency of each document and the set of tiles of each document pair are generated.

The Overlap and Cosine method will use the set of chunks with their occurrence frequency as a representation for a document while GST method uses the set of tiles of each document pair.

3.2 Comparing module

This module compares and computes the degree of similarity between all of document pairs in the input document set. This module uses the set of chunks or tiles that are generated by the previous module to compute the degree of similarity. The process of this module is divided into two steps as follow:

Step 1: After all documents are parsed into lists of chunks (or sets of tiles), similarity measure are computed for all document pairs. For example, in the trained documents set, we have *N* documents D_1, D_2,…, D_N then document D_1 is compared with *N-1* remaining documents D_2, D_3…, D_N and document D_2 is compared with *N-2* documents D_3, D_4,…, D_N and so on. It means that for each comparison method and with each specific parameter value, there are (N*(N-1))/2 comparison pairs.

Step 2: After the similarity measures of all of document pairs have been computed, a document pair is considered to be similar if its similarity measure is above a certain threshold. Comparing against the given list of similar document pairs, the effectiveness of each combination of comparison method and parameter values will be computed.

To evaluate the effectiveness of a comparison method, we use recall, precision and F-measure metrics. Let A be the list of document pairs deemed to be similar by a comparison method and B be the given list of similar document pairs, the metrics are defined as follows:

$$\textbf{Precision} = (A \cap B)/A$$

$$\textbf{Recall} = (A \cap B)/B$$

$$\textbf{F-measure} = 2*(\textbf{Precision} * \textbf{Recall})/(\textbf{Precision} + \textbf{Recall})$$

After calculating the similarity between two documents, the question is how to choose the threshold that would deem two documents similar. Threshold is a value that when the similarity measure of a document pair exceeds this value, two documents are considered similar. Choosing the threshold is very important because it determines the effectiveness of the system. The threshold value may be different for different method and different domain. In our framework, for each combination of method and parameter values, we automatically identify the threshold that maximizes the F-measure in the given domain.

3.3 Resulting showing module

The output of the *Comparing module* is the F-measure and the corresponding threshold of each method and parameter values combination. To visualize the result, we use ZedGraph* for creating 2D line and bar graphs of arbitrary datasets. Zedgraph is a very good open source C sharp graph plotting library and distributed under the GNU lesser general public license.

4. Experimental results

In this section, we present and discuss experiments to investigate the effectiveness of our framework. We use three different corpora in our experiment. Different dataset have different definition of what plagiarism is and it is implicitly encoded in the list of similar document pairs of the corpus.

4.1 Experiment with Vietnamese corpus

4.1.1 Data collection

In this experiment, we use over 800 netnews as the testing document set. These netnews are published on 14 consecutive days in some popular Vietnamese websites such as *vnexpress.net, dantri.com.vn, laodong.com.vn, tienphong.vn, tuoitre.vn, hanoimoi.com.vn, etc*... During this period, a large number of netnews of one website are copied from or have overlaps with those in other websites. Therefore, the chosen document set is good to test the system.

The documents set are divided into five groups which are economics, sport, law, medicine and mixed netnews. With each netnews group, we would like to test which method is the most effective. Our other purpose is to test the impact of word segmentation in Vietnamese documents plagiarism detection. In this experiment, plagiarized documents are highly related ones. Two documents are considered as related if they mention about one same event or problem. The creation of the list of plagiarized documents is done manually.

4.1.2 Result

Table 1. The most effective method in each group

Group	**Method**	**Chunk**	**F**	**P**	**R**	**Th**
Economic	Cosine	2-gram	**83,8%**	84,1%	83,5%	0,24
Law	GST	MML=2	**97,0%**	97,6%	95,5%	0,26
Sport	Cosine	2-gram	**81,7%**	86,8%	77,1%	0,22
Medicine	Cosine	2-gram	**91,7%**	93,3%	90,1%	0,22
Mix	Cosine	2-gram	**89,6%**	89,3%	89,9%	0,22

* http://zedgraph.org/wiki/index.php?title=Main-Page

Table 1 shows the most effective methods and parameters in each group. In this table, we find that in law netnews group, the most effective method is GST method with MML value of two and in the other netnews groups the most effective method is Cosine method with 2-gram chunk. Vietnam word segmentation does not perform well with the three comparison methods used in our framework.

4.2 Experiment with PAN corpus

4.2.1 Data collection

In the second experiment, we collected over 320 documents randomly from 2009 PAN Plagiarism Detection Competition[*]. The corpus of this competition has been created by a computer program. A suspicious document in this corpus could be inserted with one or more text passages which are given from the source documents. Before inserting into suspicious document, some words of the text passage could be replaced by one of its synonyms, antonyms and some word could be inserted or removed or parts of speech of this passage are reordered and so on.

4.2.2 Result

Table 2. The best result of each method in PAN corpus

Method	Chunk	F	P	R	Th
Cosine	7-gram	**98,0%**	98,7%	97,5%	0,002
GST	MML=7	**98,1%**	100%	96,3%	0,006
Overlap	5-gram	**98,2%**	96,5%	100%	0,004

Table 2 shows the best results of each method in this corpus. We find that all of F-measure values are very high. The highest value is 98,2% when we use Overlap method with 5-gram chunk. Although the difference between the highest F-measure values of three methods is not much but we find that as a whole Overlap method and GST method are more effective than Cosine method in this corpus.

In this experiment, the results illustrate the effectiveness of our framework. The framework can automatically find which methods and parameters are most effective for a given domain.

4.3 Experiment with corpus of P. Clough and M. Stevenson

4.3.1 Data collection

In this experiment, we used a corpus which is created and published by Paul Clough and Mark Stevenson[†] - Department of Information Studies at the University of Sheffield. They required their students to answer a set of five short questions. For each of these questions they obtained a lot of answers. Some answers are plagiarized and some are not plagiarized. Clough and Stevenson used a suitable Wikipedia website to show what, how and why the answer is plagiarized or not.

This corpus is useful to test our framework because the documents in this corpus are plagiarized by humans and these cases are real plagiarism. In this experiment we divided plagiarism into two levels of plagiarism:

[*] http://www.uni-weimar.de/cms-medien/webis/research/corpora/pan-pc-09.html

[†] http://ir.shef.ac.uk/cloughie/ resources/corpus-final09.zip

Near copy: At this level, the answers are created by performing cut-and-paste actions from Wikipedia article.

Heavy revision: At this level, the answers are created by rephrasing the source text and using different words and structures.

With each kind of plagiarism we want to test which methods and parameters are most effective.

4.3.2 Result .

Table 3 and Table 4 show the best results of each method in ***Near copy*** plagiarism level and ***Heavy revision*** plagiarism level.

Table 3. The best result of each method in Near copy plagiarism level

Method	Chunk	F	P	R	Th
Overlap	22-gram	**97,4%**	95,0%	100 %	0,16
Cosine	18-gram	**95,0%**	90,5%	100 %	0,14
GST	MML=29	**94,7%**	94,7%	94,7%	0,18

Table 4. The best result of each method in heavy revision plagiarism level

Method	Chunk	F	P	R	Th
Overlap	2-gram	**89,7%**	81,4%	100 %	0,26
Cosine	5-gram	**85,3%**	80,0%	91,4%	0,21
GST	MML=2	**82,4%**	70,0%	100 %	0,28

In this experiment, we find that in both plagiarism levels the most effective method is Overlap method. When we want to find the plagiarized documents that is cut and pasted from other documents, using large chunk is more helpful. However in the heavy-revision plagiarism level, using small chunk is more helpful. In both plagiarism cases, the GST method shows that it is not an effective method in this corpus.

In this section, we report the results of three experiments. Clearly, different domains may need different methods and corresponding parameter values. The result of all three experiments illustrates to some extent the feasibility of our framework. The framework can identify which methods and parameter values are most effective in a new domain automatically. Both Cosine and GST method are more effective than Overlap method in the experiment with Vietnamese corpus, but in both corpora of experiment with PAN corpus and experiment with corpus of P. Clough and M. Stevenson, Overlap method is the most effective one. It means that one method may be effective in this domain but may not be in others. Our framework can help users to choose the method that works the best in new domains. Users can test several kinds of chunks to find out the best chunk type. With the result of the Vietnamese corpus experiment, we conclude that word segmentation is not effective in this Vietnamese corpus.

5. Conclusion

In this paper, we have built a Unified Plagiarism Detection Framework to automatically identify which comparison method and corresponding values is most effective in a domain. The comparison methods namely: Overlap, Cosine and GST have been incorporated in the framework. Experimental results on three different corpora on different languages have demonstrated the effectiveness our approach.

Although in the Vietnamese corpus, word segmentation is not effective but it may be useful in other Vietnamese corpora. Further experiments for Vietnamese documents are needed to fully evaluate the effectiveness of word segmentation.

In the future, we will implement additional comparison methods to incorporate into the framework. We expect to evaluate our framework on larger text documents such as essays, theses and so on.

Acknowledgment. This work is partly supported by the research project No. QG.10.39 granted by Vietnam National University, Hanoi.

6. References

[1] Michael J. Wise, String similarity via greedy string tiling and running-karp-rabin matching, *Technical report, University of Sydney: Department of Computer Science,* 1993.

[2] A. Si, H.V. Leong, R.W.H. Lau. Check, A document plagiarism detection system, *Proceeding of ACM symposium for Applied Computing,* Vol 72 (12) (1997) 70.

[3] S. Brin, J. Davis, H. Garcia Molina, Copy detection mechanisms for digital documents, *In Procesding of the ACM SIGMOD international conference on management of date*, San Jose, California, 1995.

[4] N. Shivakumar, H. Garcia Molina, Scam: A copy detection mechanism for digital documents, *In Procesding International Conference on Theory and Practice of Digital Libraries,* Austin, Texas, 1995.

[5] Michael J. Wise, Yap3: improved detection of similarities in computer program and other texts, *SIGCSE Bull.,* Vol 28 (1) (1996) 130. ISSN 0097-8418. doi: http://doi.acm.org/10.1145/236462.236525.