Semantic Text Alignment based on Topic Modeling

Huong T. Le, Lam N. Pham, Duy D. Nguyen
School of Information and Computer Science Technology
Hanoi University of Science and Technology
Hanoi, Vietnam

Son V. Nguyen, An N. Nguyen
Institute of Military Science and Technology
Ministry of Defence
Hanoi, Vietnam

***Abstract*— The development of Internet makes plagiarism problem more and more serious. Plagiarism can be in different types, ranging from copying texts to adopting ideas, without giving credit to the original author. Most research in plagiarism checking concentrate on string matching. This method cannot deal with intelligent plagiarism in which the same content can be expressed by different ways. To deal with this problem, this paper proposes an approach to semantic text alignment based on sentence-level topic modeling. Experiments with PAN corpora gave us much higher recalls and approximate plagdets compared to the winning system in PAN2014. It shows that topic modeling is a potential solution for detecting intelligent plagiarism.**

Keywords*— *text alignment; topic modeling; Latent-Dirichlet Allocation; Apriori

I. INTRODUCTION

The availability of Internet makes the access to electronic texts more and more easy. However, it also gives convenient environment for plagiarism. Plagiarism can vary from literal plagiarism (i.e., reusing partial text of a document) to intelligent one (i.e., expressing the original work in a different way). This is a serious problem in many sectors such as academic, journalism, business, etc.

Over past two decades, automatic plagiarism detection has received significant attention from research community. Two main tasks of automatic plagiarism detection are source retrieval and text alignment. In the source retrieval task, given a suspicious document and a web search engine, the task is to retrieve all source documents from which text has been reused. In the text alignment subtask, given a pair of documents (suspicious and source), the task is to identify all contiguous maximal-length passages of reused text between them.

Most research works on text alignment are character-based methods (e.g., [1],[2],[3]). They apply exact string matching or approximate string matching with measures such as hamming or levenshtein distances to compute the similarity between two text spans. Instead of comparing strings as in character-based methods, vector-based methods (e.g., [4],[5]) represent the input texts are as vectors of tokens and compute the distance between these vectors by using similarity coefficients such as Jaccard, Cosine, Euclidean, or Manhattan distances.

Based on the intuition that similar documents would have similar syntactical structure, some research works (e.g., [6], [7],[8]) use syntactic information at the first stage of measuring text similarity. After that other string similarity measures are applied.

This work was supported by the Hanoi University of Science and Technology project, under Grant T2016-PC-052

978-1-5090-4134-3/16/$31.00 ©2016 IEEE

A disadvantage of the above mentioned methods is that they cannot deal with intelligent plagiarism in which the same content can be expressed by different words and in different orders. Semantic-based methods are solutions for this problem. However, there are not much research following this direction. This could be due to the difficulties in representing the semantic meaning of sentences and measuring theirs similarity.

Some researches in this direction use linguistic knowledge bases such as WordNet thesaurus and/or Wikipedia encyclopedia to measure the conceptual similarity between words (e.g., [9],[10],[11],[12]). Other methods use statistical information of words in a corpus to compute the semantic similarities between texts (e.g., [13]). The system in [13] uses vector space model for document modeling and Latent Semantic Indexing technique for measuring the semantic similarity between two paragraphs. Most existing approaches, including the approaches mentioned above, can only deal with simple intelligent plagiarism by rewording or modifying structure of original text. Such approaches cannot deal with higher levels of intelligent plagiarism such as story retelling or idea adoption.

To deal with this problem, this paper proposes an approach to group sentences with similar meaning by using topic modeling. Semantically-related text fragments are aligned based on pairs of similar sentences in the two documents. These sentences are then extended to find longer semantic-related paragraphs.

This paper is organized as follows. Our method of applying topic modeling in detecting semantically-related sentences is introduced in Section II. Section III describes our algorithm to extend pairs of similar sentences into pairs of larger text fragments that are still similar. Our experimental results are represented and analyzed in Section IV. Finally, Section V concludes our paper and proposes some directions for future work.

II. DETECTING SEMANTICALLY-RELATED SENTENCES

Sentences with semantically-related meaning will be put in the same cluster by using topic modeling. Based on this idea, we applied sentence-based topic modeling to detect similar sentences from suspicious and source documents. The basic concepts of topic modeling and then the way of applying it in our task will be introduced next.

Topic models, such as probabilistic Latent-Semantic Analysis [14] and Latent-Dirichlet Allocation [15], are

algorithms for discovering the abstract topics in a collection of documents. Given the assumption that a document is a bag of topics, words related to these topics appear in the document more frequently. The graphical model representation of smoothed LDA [15] is shown in Fig.1. In this figure, M, N, k denotes the number of documents, the number of words in a document, and the number of hidden topic, respectively. α, β are parameters of the Dirichlet priors on the per-document topic distributions and the per-topic word distribution. θ_i denotes the topic probability distribution for document i. φ_k denotes the word distribution for topic k. z_{ij} is the topic for the j^{th} word in document i. w_{ij} is the specific word.

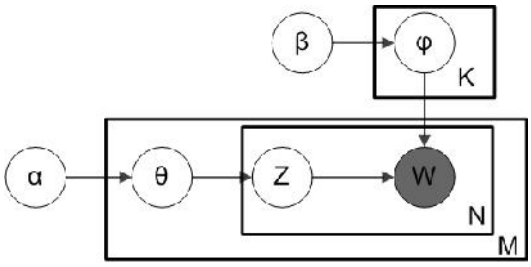

Fig. 1. The graphical model representation of smoothed LDA [15]

The inference process of LDA learns the various distributions (the set of topics, their associated word probabilities, the topic of each word, and the particular topic mixture of each document) by using variational Bayes approximation of the posterior distribution or by using Gibbs sampling.

To apply LDA in clustering sentences, each sentence is treated as a whole document. jLDADMM [16] - implementations of the LDA topic model [15] and the one-topic-per-document Dirichlet Multinomial Mixture (DMM) model on normal or short texts - is used in our paper. jLDADMM uses the collapsed Gibbs sampling algorithms for inference as described in [17] and [18]. It assumes that each line in the input file represents a document. The output contains a text file for topic assignments in which each word is assigned with a topic.

The problem with sentence-level topic modeling is that a sentence is too short with a limited context and it does not account for the context in which the sentence appear. To deal with this problem, instead of evaluating individual sentences from suspicious and source documents, a window of size n (n sentences) is used to slide from the begin to the end of both documents; sentences in these windows are used as the input for jLDADMM. Since we do not know where is content-change points in the document, we cannot simply split documents into continuously passages of n-sentences.

The process of applying jLDADMM to cluster sentences is described below.

A. Preprocessing

First, suspicious and source documents are merged into one text file D, in which suspicious document is at the beginning of the file and source one is at the end of the file. After splitting text in D into sentences, each passage of n continuous sentences is picked by the sliding window and put in one line of a new text file D1. The file D1 is processed further by tokenizing and removing stop words so that each line contains only content words from the original text.

B. Finding semantically-related passages

After the preprocessing step, the file D1 is used as the input for jLDADMM to discover topics from each passage of n sentences. jLDADMM outputs a text file D2 in which each line contains topics that have been assigned for words in this line. In our experiment, the topic of a passage is the highest frequent topic of words in that passage. Passages with the same topic are then put in the same cluster. Based on this information, a file with clustering information is generated with each line containing indexes of passages that are in the same cluster. Here index of a passage is the index of the line containing this passage in the file D2. This is also the index of the first sentence of the passage in the file D2.

In our experiment, jLDADMM was tested with different numbers of Gibbs sampling iterations from 1000, 2000, 5000, 10000 to 100000. The conclusion withdraws from our experiments is that the iteration = 2000 is good enough for our task. More iterations do not improve the system's accuracy. However, since the input of jLDADMM in our case are a set of n continuous sentences, whose length may be not long enough for clustering task, jLDADMM may return slightly different results at different runs.

To solve this problem, jLDADMM was run 10 times for each input file to assure the convergence of results. Each time running jLDADMM, the system generates a file containing clustering information called Clusters_k (k=1÷10). Passages that are in the same cluster frequent enough are considered as semantically-related meaning ones. A modifying version of Apriori algorithm is applied to get frequent sets from clusters returned by 10 times running jLDADMM. Here, each cluster is seen as a set of items (an itemset). The itemsets that appear more than 3 times are considered as frequent ones.

Since our purpose is to find pairs of passages that one belongs to suspicious document and one belongs to source document, all clusters that contain passages from only one document (suspicious or source) are removed from Clusters_1 to Clusters_10 before applying the modifying version of Apriori algorithm.

The original idea of Apriori algorithm and our modifying version of Apriori algorithm is shown below.

Apriori algorithm

Apriori [19] is an algorithm for mining frequent itemsets from database's transactions (for example, collections of items bought by customers). The main operator of Apriori is to count up the number of occurrences, called the support, of each item or itemset in database's transactions. A frequent itemset (denoted L_k, where *k* is the set's size) is an itemset whose support is greater than an user-specified minimum support (called *minsup*). The Apriori algorithm bases on the idea: if {AB} is a frequent itemset, both {A} and {B} should be a

frequent itemset. The original Apriori algorithm can be found in [19].

In our task, a cluster plays the role of a database's transaction. Passages' indexes in a cluster are considered as items in a transaction.

Let us called C_k and F_k are candidate itemsets and frequent itemsets with the size k, respectively. Our modified Apriori algorithm is as follows.

Input:

- ClusterSet CS including Clusters_1 to Clusters_10.
- A minimum support *minsup*. *minsup* is set to 3 by default.

Output: Frequent itemsets in which each itemset has at least two items.

Algorithm:

1. C_1 = CS
2. Filter items of C_1 that appear more than *minsup* times, put them in frequent itemsets F_1
3. k = 2
4. While F_{k-1} is not empty
   1. Generate the candidate itemsets C_k from the frequent itemsets F_{k-1}:
   if k = 2
   $C_2 \leftarrow \{\{a\}\cup\{b\} \mid \{a\},\{b\} \in F_1 \wedge b \neq a \wedge doc(a) \neq doc(b)\}$
   //doc(i) is the document that i belongs to; doc(i) can either be suspicious or source document
   else
   $C_k \leftarrow \{p\cup\{b\} \mid p \in F_{k-1} \wedge b \notin p \wedge b \in q \wedge q \in F_{k-1}\} - \{r \mid \{s \mid s \subseteq r \wedge |s|=k-1\} \not\subset F_{k-1}\}$
   in which p and q are frequent itemsets in F_{k-1}; b is an item in q but not in p; s is an itemset with size k-1, s is a subset of an itemset r, s is not a frequent itemset in F_{k-1}
   2. For each cluster C $\in$ CS
   Get all candidate itemset t $\in$ C_k and is an item subset in cluster C, save them in Ccluster:
   Ccluster $\leftarrow$ $\{t \mid t \in C_k \wedge t \subseteq C\}$
   For each candidate itemset t $\in$ Ccluster
   count[t] $\leftarrow$ count[t] +1
   3. Get all candidates in C_k that have the number of appearance is not less than minsup:
   $F_k \leftarrow \{t \mid t \in C_k \wedge count[t] \geq minsup\}$
   4. k $\leftarrow$ k + 1
5. Return frequent itemsets F = $\bigcup_{i=2 \div k} F_i$

Our modified version of Apriori algorithm is illustrated by the following example:

Suppose that the jLDADMM is ran three times. After filtering all clusters that contain passages from only one document (suspicious or source), three clusters are as follows:

Clusters_1:
topic 4: 12 100 108
topic 7: 13 25 52 67 94
topic 10: 1 3 13 38
topic 13: 7 11 95 99
topic 17: 7 8 86 95 96
Clusters_2:
topic 7: 13 86 101 107
topic 8: 11 99 106
topic 11: 8 96
topic 17: 1 3 5 13 14 16 38 66 69
Clusters_3:
topic 2: 13 21 25 67 94
topic 4: 1 3 13 14 14 16 20 28 38 69 69 76 107
topic 6: 3 18 39 53 76 115
topic 8: 11 58 60 106
topic 12: 7 22 29 95
topic 15: 12 78 100

In the above clusters, a number is the sentence index in the text. ClusterSet CS is a combination of all these clusters. The appearances of each number

The frequent itemsets F are then sorted by their appearance time in CS. The itemsets that appear more frequently contain indexes of passages that are more similar. These itemsets are used as materials for the next stage of text alignment - extension stage - which will be described next.

III. EXTENSION STAGE

Given the frequent itemsets F, each itemset contains indexes of passages from both suspicious and source documents, the purpose of the extension stage is to align similar text fragments between these documents. This is done by aligning passages from the highest frequent itemsets, then extending these passages by merging them with their neighbor pairs, so that those larger passages still be similar.

There are two merging cases: (i) merging passages of a document in one cluster; and (ii) merging passages of a document from two clusters. The first case is carried out when the distance between two passages in a frequent itemset and in one document is smaller than maximum gap for merging text fragments with the same meaning (called *maxGapSameMeaning*). The distance between two passages is calculated as the nearest distance (based on sentence indexes) among sentences in the two passages. This process returns pairs of passages' indexes $[(p_i,p_j),(p_k,p_l)]$ that appear most frequent in all clusters. (p_i,p_j) are indexes of the passage that expands from passage with index p_i to passage with index p_j of the suspicious document. (p_k,p_l) represents for the passage that expands from passage with index pk to passage with index pl of the source document. (p_i,p_j) and (p_k,p_l) are from the same cluster.

The second case is carried out between each highest frequent pair $[(p_i,p_j),(p_k,p_l)]$ and a pair $[(p_{i'},p_{j'}), (p_{k'},p_{l'})]$ returned by the first case's process when the distance between two pairs of passages is smaller than maximum gap for merging text fragments with different meaning (called

maxGapDifferMeaning). (p_i,p_j) and $(p_{i'},p_{j'})$ are passages from the suspicious document; (p_k,p_l) and $(p_{k'},p_{l'})$ are from the source document.

Our extension algorithm is as follows:

Input: frequent itemsets F, size of the sliding window *n*

Output: pairs of similar passages in suspicious and source documents

Algorithm:

1. Set weight for each itemset equals to its count returned by the Apriori algorithm
2. For each itemset IS ∈ F:

   Create a new set IS1= NULL

   Foreach item $p_i \in$ IS, insert pair (p_i, p_i) to IS1

   Repeat

   If there are pairs (p_i,p_j) and $(p_k,p_l) \in$ IS1 and p_i, p_j, p_k, p_l are indexes from one document (Suspicious or Source), and $p_k-(p_j+n-1)<$ *maxGapSameMeaning*:

   Replace (p_i,p_j) and (p_k,p_l) by (p_i,p_l) in IS1

   Until each pair in IS1 and in the same document cannot be expanded anymore.
3. At the end of step 2, each set IS1 contains indexes of passages from suspicious and source documents with maximum lengths. Generate pairs $[(p_i,p_j),(p_k,p_l)]$ from IS1, with $(p_i,p_j) \in$ Suspicious and $(p_k,p_l) \in$ Source. All of these pairs and their weights are put in a new set IS2.
4. For each pair $[(p_i,p_j),(p_k,p_l)] \in$ IS2 with the highest weight:

   Repeat

   1. Create lower bounds and upper bounds:

      lowbSusp = p_i – *maxGapDifferMeaning*

      upbSusp = p_j + n - 1 + *maxGapDifferMeaning*

      lowbSour = p_k – *maxGapDifferMeaning*

      upbSour = p_l + n - 1 + *maxGapDifferMeaning*

      in which *lowbSusp*, *upbSusp* are lower bound and upper bound for passage's indexes in the suspicious document; *lowbSour*, *upbSour* are lower bound and upper bound for the passage's index in the source document, respectively.
   2. If there is a pair $[(p_{i'},p_{j'}), (p_{k'},p_{l'})] \in$ IS2 satisfying the following conditions:

      lowbSusp < $p_{i'} < p_{j'}$ < upbSusp

      lowbSour < $p_{k'} < p_{l'}$ < upbSour

      then

      a. Merge (p_i,p_j) with $(p_{i'},p_{j'})$ to (p_m, p_n) in which $p_m = \min(p_i,p_{i'})$, $p_n = \max(p_j,p_{j'})$
      b. Merge (p_k,p_l) with $(p_{k'},p_{l'})$ to (p_r,p_s) in which $p_r = \min(p_k,p_{k'})$, $p_s = \max(p_l,p_{l'})$
      c. Replace $[(p_i,p_j), (p_k,p_l)]$ and $[(p_{i'},p_{j'}), (p_{k'},p_{l'})]$ by $[(p_m,p_n),(p_r,p_s)]$ in IS2 with the weight equal to the highest weight
      d. Assign i=m, j=n, k=r, l=s

   until no further pairs can be merged
5. Return pairs $[(p_x,p_y), (p_z,p_t)] \in$ IS2 with the highest weight

---

Results of the extension stage are pairs of passages that are aligned between suspicious and source documents with the highest weight. They are used as our system's output.

IV. Experimental Results

A. Data Sets

Since this research concerns about detecting intelligent plagiarism, corpora containing information about intelligent copies are selected to evaluate our system. They are text alignment corpora provided in PAN series including "summary" data sets in text alignment training and testing corpora of PAN2013 [20]; Cheema [21] and Alvi data set [22] of PAN2015. PAN2013 training corpus includes 3230 and 1827 files in source and in suspicious folders, respectively. PAN2013 test corpus has 3169 source files and 1826 suspicious ones. Materials of PAN 2013 corpora are documents from various sources. Cheema corpus contains documents from students' work, with 248 files in each source and suspicious folder. Alvi corpus contains 70 source files and 90 suspicious files taken from various translations of Grimms fairy tales. Some documents in Alvi data set do not use Unicode font; these files were not used in our experiments.

B. Evaluation Measures

Our system was evaluated by using a tool to measure performance provided by PAN [24]. Four measures used in PAN to evaluate system performance [20] are macro-averaged *Precision* and *Recall*, *Plagdet*, and *Granularity*.

Given S, R, s, r are a set of all plagiarism cases, a set of all plagiarism system-detection cases, a plagiarism case, and a plagiarism system-detection case, respectively. The macro-averaged precision and recall are defined as follows:

$$prec(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{\left|\cup_{s\in S}(s\cap r)\right|}{|r|} \quad (1)$$

$$rec(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{\left|\cup_{r\in R}(s\cap r)\right|}{|s|} \quad (2)$$

The detection granularity of R under S indicates whether each plagiarism case $s \in S$ is detected as a whole or in several pieces. It is calculated as:

$$gran(S,R) = \frac{1}{|S_R|}\sum_{s \in S_R} |R_s| \quad (3)$$

where $S_R \subseteq S$ are cases detected by detections in R, and $R_S \subseteq R$ are the detections of a given s.

Plagdet is the overall score of the system, which is calculated as:

$$plag\det(S,R) = \frac{2 * prec * rec}{prec + rec} * \frac{1}{\log_2(1 + gran(S,R))} \quad (4)$$

C. Experimental Results

The configuration of jLDADMM in our experiments as follows: α = 0.1; β = 0.1; the topic model *model* = LDA. Since each data domain has a different way of using vocabulary, the number of topics (*ntopics*) and the minimum support (*minsup*) of the Apriori algorithm are varied by domain. ntopics was set to 10 and minsup was set to 2 for "*summary*" data set and Cheema dataset. These values were 30 and 3 for Alvi one. The window size was set to 3 by default.

Our experimental results are shown in Table I below.

TABLE I. OUR RESULTS ON DIFFERENT PAN CORPORA

Corpus		Prec	Rec	Gran	Plagdet
PAN 2013	"summary" - Training corpus	0.8015	**0.7722**	1.0	**0.7866**
	"summary" - Testing corpus	0.8344	**0.7701**	1.0	**0.8010**
Cheema	02-undergrad-in-progress	0.4630	**0.8702**	1.0	0.6044
	03-undergrad	0.4407	**0.7427**	1.0	**0.5530**
	04-masters	0.5787	**0.8**	1.0	**0.6716**
	05-phd	0.3696	**0.8872**	1.0	**0.5218**
	Entire	0.4457	**0.8281**	1.0	**0.5795**
Alvi	02-human-retelling	0.5769	**0.8446**	1.0	0.6856
	03-synonym-replacement	0.5513	**0.8304**	1.0	0.6627
	Entire	0.6017	**0.8194**	1.0	0.6939

In the above table, numbers in the "Entire" line are results when running the system with all folders (four data sets in Cheema corpus and two data sets in Alvi corpus).

In order to evaluate the efficiency of our approach, our system was compared with existing approaches in this field. Since PAN2015 and PAN2016 do not have any research about text alignment for English language, we compared our system with the winning approach to text alignment at PAN2014, created by Sanchez-Perez [23]. Sanchez-Perez's method relies on a sentence similarity measure based on tf-idf. Based on the seed set S of sentence pairs, the system extends them to form larger text fragments that are similar between two documents. Finally, it resolves overlapping fragments and removes short fragments from the result.

Sanchez-Perez's system[1] was tested with the same data sets mentioned above. The results are shown in Table II below.

[1] The source code is at http://www.gelbukh.com/plagiarism-detection/PAN-2014/.

TABLE II. SANCHEZ-PEREZ'S RESULTS ON DIFFERENT PAN CORPORA

Corpus		Prec	Rec	Gran	Plagdet
PAN 2013	"summary" - Training corpus	**0.9941**	0.4235	1.0435	0.5761
	"summary" - Testing corpus	**0.9990**	0.4158	1.0585	0.5638
Cheema	02-undergrad-in-progress	**0.8440**	0.6491	1.0	**0.7338**
	03-undergrad	**0.8633**	0.2976	1.0	0.4426
	04-masters	**0.9961**	0.2595	1.0	0.4117
	05-phd	**0.8934**	0.1638	1.0	0.2769
	Entire	**0.8644**	0.3348	1.0	0.4826
Alvi	02-human-retelling	**0.9499**	0.5961	1.0	**0.7325**
	03-synonym-replacement	**0.9686**	0.8595	1.0	**0.9108**
	Entire	**0.9607**	0.7278	1.0	**0.8282**

The result tables show that our system's recall is much higher than that of Sanchez-Perez's. It indicates that our system can detect most plagiarism cases comparing to Sanchez-Perez's one. This confirms our assumption that topic modeling is a good solution for detecting intelligent plagiarism in which the same content can be expressed in different ways and by different words.

Our Plagdet scores are higher than Sanchez-Perez's one in six out of ten cases. However, our precision is lower than Sanchez-Perez's system. In other words, the accuracy of our system's prediction is lower. By checking the system's output, we found that although some alignments are correct, they are either larger or smaller than the actual pairs. This is the main reason for the low precision of our system. This problem will be solved in our future work, at the post processing stage after the extension step.

Some typical errrors in our system's output are:

In Chemma corpus, plagiarised document pairs are generated by inserting manually-created plagiarised paragraphs into documents. The inserted paragraphs may break a sentence into two fragments, such as the paragraph below:

In suspicious document:

Volunteers and financial support to provide volunteers with the assistance they need*Vampires are magical and imaginary beings who are dependent upon feeding on human beings especially their blood. Folklores tell us stories about vampires visiting their loved ones and also causing misery and destruction in the neighborhood they used to live in when they were alive.*, is critical to reaching Project Gutenberg-tm's goals and ensuring that the Project Gutenberg-tm collection will remain freely available for generations to come.

In source document:

I painted the unquestionable result of being taken after such r*A vampire is a mythical being who subsists by feeding on the life essence (generally in the form of blood) of living creatures. In folkloric tales, undead vampires often visited loved ones and caused mischief or deaths in the neighborhoods they inhabited when they were alive*.esistance as had already been made.

In such cases, the first half of the broken sentence is considered as in the same sentence with the first sentence of the inserted paragraph. The second half of the broken sentence is considered as in the same sentence with the last sentence of the inserted paragraph. Therefore, when the inserted paragraphs are detected, the first and second halfs of the broken sentence are also included in the plagiarised fragments. These redundances reduce the system's recall.

After generating source-plagiarised fragment pairs, To create artificial intở corpus khi chèn text ngẫu nhiên vào để làm giả sao chép thì có nhiều trường hợp chèn giữa 1 từ làm từ bị chia làm 2 phần → hậu xử lý kết quả đầu ra của chương trìn

Chỗ chèn text vào thường giữa 1 từ và phần đầu ở dạng character(s)Word và kết thúc ở dạng word.character(s). Xử lý = cách tìm trong câu đầu tiên xâu dạng aaaBbb và cắt đoạn từ đầu đến aaa khỏi kết quả.

V. Conclusions and Future Works

This paper introducces our approach to semantic text alignment based on LDA topic modeling. Because of the unstable of sentence-level topic modeling, several solutions have been proposed. First, a sliding window was used to run from the beginning to the end of documents. Sentences within this window were used as materials for topic modeling, instead of single sentence ones. Second, the topic modeling tool was run ten times to assure the reliability of the results. Third, a modified version of the Apriori algorithm has been proposed to get frequent item sets after ten time running topic modeling tool. Finally, an extension procedure was used to extend the alignment passages to larger ones. Experimental results show that our propose approach is potential in improving the performance of a text alignment system. Future work includes investigating methods for post processing outputs of the extension stage, in order to solve overlapping cases and to refine borders of alignment passages.

References

[1] C. Grozea, C. Gehl, and M. Popescu, “ENCOPLOT: Pairwise sequence matching in linear time applied to plagiarism detection,” in *Proc. SEPLN*, Donostia, Spain, 2012, pp. 10–18.

[2] M. Elhadi and A. Al-Tobi, “Duplicate detection in documents and webpages using improved longest common subsequence and documents syntactical structures,” in Proc. 4th Int. Conf. Comput. Sci. Converg. Inf. *Technol.*, Seoul, Korea, Nov. 2009, pp. 679–684.

[3] J.Kasprzak, M.Brandejs, and M.Kripac, “Finding plagiarism by evaluating document similarities,” in *Proc. SEPLN*, Donostia, Spain, 2009, pp. 24–28.

[4] A. Barron-Cedeno, C. Basile, M. Degli Esposti, and P. Rosso, “Word length n-Grams for text re-use detection”, in Computational Linguistics and Intelligent Text Processing, 2010, pp. 687–699.

[5] M. Murugesan, W. Jiang, C. Clifton, L. Si, and J. Vaidya, “Efficient privacy-preserving similar document detection,” *VLDB J.*, vol. 19, no. 4, pp. 457–475, 2010.

[6] M. Elhadi and A. Al-Tobi, “Duplicate detection in documents and webpages using improved longest common subsequence and documents syntactical structures,” in Proc. 4th Int. Conf. Comput. Sci. Converg. Inf. *Technol.*, Seoul, Korea, Nov. 2009, pp. 679–684.

[7] M. Elhadi and A. Al-Tobi, “Use of text syntactical structures in detection of document duplicates,” in Proc. 3rd Int. Conf. Digital Inf. Manage., London, U.K., 2008, pp. 520–525.

[8] K. Koroutchev and M. Cebri´an, “Detecting translations of the same text and data with common source,” *J. Stat. Mech.: Theor. Exp.*, p. P10009, 2006.

[9] Y. Li,D.McLean, Z.A. Bandar, J. D. O’Shea, andK. Crockett, “Sentence similarity based on semantic nets and corpus statistics,” *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1138–1150, Aug. 2006.

[10] Tsatsaronis G, Varlamis I, Giannakoulopoulos A, Kanellopoulos N. "Identifying free text plagiarism based on semantic similarity", Proceedings of the 4th International Plagiarism Conference, 2010.

[11] Yurii Palkovskii, Alexei Belov, Iryna Muzyka, "Using WordNet-based semantic similarity measurement in External Plagiarism Detection", Notebook for PAN at CLEF 2011

[12] Eman Salih Al-Shamery and Hadeel Qasem Gheni. Plagiarism Detection using Semantic Analysis. Indian Journal of Science and Technology, Vol 9(1), DOI: 10.17485/ijst/2016/v9i1/84235, January 2016

[13] De Cao Tran, Tri Cao Tran, Copy Detection Using Latent Semantic Similarity, IEEE International Conference on Research, Innovation & Vision for the Future, RIVF 2008, July 13-17, 2008, Ho Chi Minh City, Vietnam.

[14] T. Hofmann, "Probabilistic latent semantic indexing", in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999, pages 50–57.

[15] Blei, D. M., Ng, A. Y., Jordan, M. I., "Latent Dirichlet Allocation", Journal of Machine Learning Research 3, 2003, pp.993–1022.

[16] Dat Quoc Nguyen. 2015, "jLDADMM: A Java package for the LDA and DMM topic models", http://jldadmm.sourceforge.net/.

[17] Dat Quoc Nguyen, Richard Billingsley, Lan Du and Mark Johnson. 2015, Improving Topic Models with Latent Feature Word Representations," Transactions of the Association for Computational Linguistics, vol. 3, pp. 299-313.

[18] Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. 2011. Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. Information Retrieval, 14:178–203.

[19] Rakesh Agrawal and Ramakrishnan Srikant Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, 1994, pp. 487-499.

[20] Martin Potthast , Benno Stein , Alberto Barrón-cedeño , Paolo Rosso , Bauhaus-universität Weimar, "An evaluation framework for plagiarism detection", In Proceedings of the 23rd International Conference on Computational Linguistics, 2010.

[21] Waqas Arshad Cheema, Fahad Najib, Shakil Ahmed, Syed Husnain Bukhari, Abdul Sittar, and Rao Muhammad Adeel Nawab, "A Corpus for Analyzing Text Reuse by People of Different Groups" Notebook for PAN at CLEF 2015.

[22] Faisal Alvi, Mark Stevenson, and Paul Clough. " The Short Stories Corpus", Notebook for PAN at CLEF 2015.

[23] Miguel A. Sanchez-Perez, Grigori Sidorov, and Alexander Gelbukh, "A Winning Approach to Text Alignment for Text Reuse Detection", Notebook for PAN at CLEF 2014.

[24] Tool for text alignment performance measures, http://www.uni-weimar.de/medien/webis/events/pan-14/pan14-web/plagiarism-detection.html, last visited July 2016