Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2017"

Moscow, May 31—June 3, 2017

PARAPHRASED PLAGIARISM DETECTION USING SENTENCE SIMILARITY

Zubarev D. V. (dvzubarev@yandex.ru),
Sochenkov I. V. (isochenkov@sci.pfu.edu.ru)

RUDN University, Moscow, Russia

Federal Research Center "Computer Science and Control"
of Russian Academy of Sciences, Moscow, Russia

The paper describes an approach to plagiarism detection within PlagEvalRus-2017 competition. Our system leverages deep parsing techniques to be able to detect moderately disguised plagiarism. We participated in the two tracks of the competition: source retrieval (sources detection) and text alignment (paraphrased plagiarism detection). There are various cases of plagiarism presented in datasets of both tracks. They vary by the level of disguise that was used while reusing text. The results show that our method performed quite well for detecting moderately disguised forms of plagiarism.

Keywords: plagiarism detection, sentence similarity, plagiarism detection evaluation

МЕТОД ПОИСКА ПЕРЕФРАЗИРОВАННЫХ ТЕКСТОВЫХ ЗАИМСТВОВАНИЙ НА ОСНОВЕ ОЦЕНКИ СХОДСТВА ПРЕДЛОЖЕНИЙ

Зубарев Д. В. (dvzubarev@yandex.ru),
Соченков И. В. (isochenkov@sci.pfu.edu.ru)

Российский университет дружбы народов, Москва, Россия

Федеральный исследовательский центр «Информатика
и управление» Российской академии наук, Москва, Россия

1. Introduction

Plagiarism is a serious and well-known problem in education and science. With the rapid increase of amount of texts available in digital form, it is crucial to detect sources of plagiarism fast enough in the huge number of documents.

Since plagiarism detection systems (PDS) are very common now, authors who reuse text have learned to disguise the fact of plagiarism. Since "copy-paste" plagiarism is likely to be detected, authors use various techniques such as paraphrasing, words reordering, split/join of sentences and so on (Alzahrani et al., 2012). Therefore, it is important for the state-of-the-art PDS to detect such cases also. Paraphrased plagiarism detection on a large amount of potential sources is a challenging task with no "gold-standard" solution for now. In general it is about to find a balance between false positives and false negatives in results of plagiarism detection. Therefore, it is important to evaluate information retrieval methods for plagiarism detection to find the most promising solutions.

PlagEvalRus-2017 is the first Russian competition for evaluation of plagiarism detection methods. It addresses the two main tasks in this area: source retrieval and text alignment. PlagEvalRus-2017 is a playground with the open dataset for researchers dealing with two aforementioned tasks. This dataset contains about 7 millions documents and plagiarism cases that vary by the level of complexity: from copy-paste plagiarism to heavily disguised plagiarism. In source retrieval track, participating PDS should find for given suspicious documents all sources in the entire collection. In text alignment track, the participants should detect all reused text between documents in the given pairs.

In this paper, we describe our approach for detecting plagiarism, which uses deep linguistic parsing of texts. It includes PoS-tagging, syntactic parsing, semantic role labeling, and semantic relation extraction. We also employ our own indexing subsystem that provides an efficient storage for rich information about words and an effective data access for the fast candidates' selection. The evaluation of our approach on the PlagEvalRus-2017 is also presented.

2. Related work

A comprehensive overview of approaches used to detect plagiarism is given in (Meuschke et al., 2013). Another overview along with taxonomy of plagiarism is given in (Alzahrani et al., 2012). Classic approach for detecting plagiarism is to use N-word-grams or N-character-grams. Recent research focused on incorporating syntactic and semantic information into detection methods. (Lin et al., 2012) used six similarity scores to measure the degree of plagiarism between fragments. Although they showed that impact of semantic and syntactic aspects to the overall performance was quite small. (Osman et al., 2012) measured sentence similarity based on semantic role labeling and achieved an improvement of more than 35% for both, precision and recall in comparison with classical methods.

It is very important to have standardized dataset, on which researchers can evaluate all new methods. An overview on the evaluation of plagiarism is given in (Kraus, 2016). Actually, there are few open datasets for such evaluation and mostly used is PAN-PC-11 corpus (Potthast et al., 2010). This corpus was used in PAN competition that held yearly since 2009 until 2015 year. The corpus consists of documents that were created by borrowing text of books from Gutenberg collection. Reused text was modified automatically and manually. Since the text is borrowed randomly from any book, the suspicious documents do not belong to the same topic as sources. This is the main concern related with this corpus and it makes it suitable only for evaluation of the text alignment task. Those corpora comprise documents are mostly in English.

3. The proposed Plagiarism detection method

In this section, we describe our method for external plagiarism detection. Our method relies on a collection of documents in which sources of a potentially plagiarized document would be located. Therefore an indexing subsystem is crucial for our method.

3.1. Data Indexing

We use our own indexing subsystem designed for an efficient (in terms of space on hard drive) storage of various words characteristics (PoS-tags, semantic roles etc.). To provide this information for indexing we perform linguistic analysis of texts, which includes postagging, syntactic parsing, semantic role labeling, and semantic relation extraction (Osipov et al., 2013), (Shelmanov and Smirnov, 2014). This information is used when we measure similarity between sentences.

3.2. Search method

One approach of searching text in the large collection is to use a search engine with special operators such as quorum matching or proximity search. This approach becomes impractical with a large number of documents. Each query consists of ten or more words and takes considerable time to complete. Since a search engine loads a list of occurrences of each word and then merges these lists into one to perform search. The amount of queries for a document depends on its size, but it is typically in order of hundreds. Our current method performs 18 times faster than our previous algorithm (based on search engines) for large documents (PhD theses) and 12 times faster for documents like medium wiki article.

Search of plagiarism is divided into three stages.

First stage

We represent the suspicious document as a bag of terms: words and two-word noun phrases. Each word and two-word noun phrase is normalized. These terms are sorted by the TF-IDF weight (IDF weights are calculated based on word and phrase frequencies in all indexed documents) and the top N terms with the highest weight are sent as a request to the indexing subsystem for retrieving similar documents. N is dependent on the amount of unique terms in a document (e.g. we select 45% of all terms but maximum is 120).

Indexing subsystem contains pre-built inverted spectral index for the whole collection of documents. This index stores a mapping from terms to their TF-IDF weights and a document (as the modification of the inverted index described in (Elsayed, 2008)). The index is employed for quick loading of all other vectors that overlap with the query vector. Then we calculate the modified Hamming distance to estimate the similarity score between suspicious document and documents in index. The full description of this method is given in (Suvorov and Sochenkov, 2015). We use 600 most similar documents as candidates: documents that may contain plagiarism. All other documents from the collection of sources are not taken into consideration.

Second stage

In this stage, we consider sentences as a sequence of words. We weight all sentences from the suspicious document by TF-IDF. The least significant ones (weight < 0.01) are dropped. In addition, we discard sentences that contain less than *K* or more than *L* words. We also discard all duplicate sentences. The remaining sentences will be analyzed for plagiarism.

We represent each sentence as a vector of unique numbers (each number is a derived from the normal form of corresponding word occurrence in sentence). Next, we intersect each selected sentence from the suspicious document with all other sentences from the candidates found on the previous stage. The goal is to exclude irrelevant pairs of sentences that share small amount of words from further consideration. For that task we use fast set intersection algorithm (Takuma, 2013). It proved to be very efficient for this task, since it boils down to multiple bitwise operations for each pair of sentence. Pairs of sentences that share at least *M%* of words are passed to the next stage.

Third stage

The remaining pairs of sentences are scored using a sentence similarity measure. We described this measure in (Zubarev and Sochenkov, 2014). We will only briefly recap it here.

Given two arbitrary sentences *s*1 and *s*2, denote as *N*(*s*1, *s*2) a set of pairs of words with the same normal form, where the first element is taken from *s*1 and the second one from *s*2. We compare two sentences by considering words from the set *N*(*s*1, *s*2). For calculating overall similarity measure of two sentences we compute multiple similarities measures and then combine its values. Employed similarities are described below.

IDF overlap measure

We define IDF overlap as follows:

$$I_1(s1, s2) = \sum_{(w1,w2)\in N(s1,s2)} v(w1, s1)$$

where *v*(*w*1, *s*1) is IDF weight of word *w*1 in a sentence *s*1. Also there holds an equation

$$\sum_{w\in s1} v(w) = 1$$

TF-IDF measure

Let us define TF-IDF measure in the following way:

$$I_2(s1, s2) = \sum_{(w1,w2) \in N(s1,s2)} \upsilon(w1, s1) TF_{w2}$$

where $\upsilon(w1, s1)$ is IDF weight of the word $w1 \in s1$; TF_{w2} is TF weight of the word $w2 \in s2$.

Sentence syntactic similarity measure

To be able to measure this kind of similarity we need to use rich information stored in indexing subsystem for each word. We define $Syn(s1)$ as a set that contains triplets (w_h, σ, w_d), where w_h, w_d are normalized head and dependent word respectively, σ is type of syntactic relation. Then we define syntactic similarity in the following way:

$$I_3(s1, s2) = \frac{\sum_{(w_h,\sigma,w_d) \in (Syn(s1) \cap Syn(s2))} \upsilon(w_h, s1)}{\sum_{(w_h,\sigma,w_d) \in Syn(s1)} \upsilon(w_h, s1)}$$

Sentence semantic similarity measure

For semantic information representation in a sentence we need to define:

- A finite set of semantic values—*Roles*. Further in the text we will call them roles (Shelmanov and Smirnov, 2014).
- *SentRoles*(*s*) is a set which contains pairs (*w*, *ρ*), where *w* is a normalized word from a sentence with an assigned role. Each word can have one or more semantic roles in the sentence.

Then we define semantic similarity in the following way:

$$I_4(s1, s2) = \frac{|SentRoles(s1) \cap SentRoles(s2)|}{|SentRoles(s1)|}$$

The denominator of the previous formula can be equal to 0 when no roles were identified in the sentence. In this case the criterion is equal to 0.

Sentence semantic relations similarity measure

For semantic relations representation in a sentence we need to define:

- Set of types of relations *R* on the set of semantic roles (Osipov et al., 2013).
- *SentRels*(*s*) is a set which contains pairs *w*1, *w*2, which determine semantically related words in a sentence, *w*1 and *w*2 should have any role assigned.

We define *SemR*(*s*, *w*) as a set

$$\{a \in Roles | \exists w1 \in s: (w, w1) \in SentRels(s) \wedge (w1, a) \in SentRoles(s)\}$$

It is a set of roles which were assigned to words *w*1 that are linked with words *w* in this sentence *s* by any semantic links. Then we define semantic relations similarity in the following way:

$$I_5(s1, s2) = \frac{\sum_{(w1,w2) \in N(s1,s2)} |SemR(s1, w1) \cap SemR(s2, w2)|}{|SentRoles(s1)|}$$

Overall sentence similarity

The overall sentence similarity we define as a linear combination of described measures.

$$Sim(s1, s2) = \sum_{i=1}^{5} k_i I_i(s1, s2),$$

where k_i, $i = [1;5]$ determine relative contributions of each similarity.

Rationale for syntactic/semantic measures is to treat sentences not as a bag-of-words but as syntactically linked text with the meaning. Value of these measures will be low for sentences with the same words but with different usage of words.

Post-processing

There are two thresholds, which a pair of sentences must exceed to be considered as suspicious. First, a minimal value of IDF overlap measure and second, a minimal value of the overall sentence similarity.

Then all suspicious sentences are grouped by sources. Sources are sorted by the count of the sentences in them. We discard some sources: if they contain small number of sentences or if the percent of sentences from the total count is too small.

4. Tuning plagiarism detection method

There are many tunable parameters in the described method. We needed to tune 13 parameters each of them had from 10 to 20 values in general. It was not feasible to perform an exhaustive grid search for them. So we employed some kind of random search. At the beginning of search we initialize each parameter with a random value. Then we iterate over each parameter and tweak it by increasing/decreasing it slightly with respect of its bounds. On each iteration, we measure the performance of the detection method. The parameters from the best iteration are adopted as the current set of parameters and the search is repeated again. The search is interrupted, when the performance of the detection method is not changed for a while, and started again with new random values. We performed about 20 such restarts while optimizing parameters of the detection method. Mostly all searches converged to approximately one value with standard deviation 0.018. We optimized our method separately for text alignment and source retrieval tasks since these tasks use different performance measures.

5. Evaluation

5.1. Source retrieval task

We consider source retrieval as the first step of plagiarism detection, when all sources should be collected. Source retrieval occurs on the first stage in our plagiarism detection method. So we wanted to test how many sources we can find with our first stage.

Source retrieval training set includes plagiarism cases with various obfuscation types. **Academic** includes real world examples of plagiarism in academic environment (519 documents). This collection consists of PhD theses in which plagiarism was found. Texts from this collection contain copy-paste plagiarism in general.

Essays-1 (manually-paraphrased—name in the corpus) includes manually written essays on the given topic (118 documents). Authors of essays were asked to actively reuse other texts and change them. The texts from this collection may be described as being moderately disguised.

Essays-2 (manually-paraphrased2) the same as **Essays-1**, but they are heavily disguised in general (34 documents). **Generated plagiarism** includes suspicious documents generated automatically (1000 documents). We didn't evaluate this collection since the suspicious documents were filled with passages from the random sources and they are very likely on different topics. Hence it makes little sense trying to retrieve those sources. The results on the training dataset are presented in the following table.

Table 1. Results on the training data for source retrieval

	Recall	Mean average precision	Precision
Academic	0.97	0.359	0.001
Essays-1	0.983	0.149	0.009
Essays-2	0.969	0.118	0.009
Generated	—	—	—

The result shows that the first stage of our method is able to find most sources of plagiarism even when a search is performed against 7 millions documents. It means that most sources are in those 600 candidates that are left after the first stage and we still can find them in the next stages. Precision is low since we deliberately turn off any filtering of false candidates. We will show more balanced version of source retrieval when evaluating the whole method for detecting plagiarism in the next section.

Result on the test data were provided by the organizers. They are similar to results obtained on the training data, except that test data lacked Essays-1 collection.

Table 2. Results on the test data for source retrieval

	Recall	Mean average precision	Precision
Academic	0.978	0.61	0.003
Essays-2	0.989	0.39	0.009
Generated paraphrasing	0.75	0.2	0.005

5.2. Text alignment task

Text alignment is the crucial step of plagiarism detection, when reused text should be identified. Text alignment occurs on the second and third stage in our plagiarism detection method. For evaluating text alignment we use all stages except the first one, since a pair of documents is given in this task. Text alignment training set overlaps with the source retrieval training set. There is additional information in each corpus that is useful for text alignment task. In Essays-1 collection, authors annotated each pair of sentences with the type of obfuscation, which was used while modifying text. In **Essays-2**, authors were allowed to use more obfuscation types and each pair of sentences may be annotated with the multiple types. For example 'ADD,SYN' means that there were used addition of words and replacing some words with synonyms.

Standard metrics for text alignment were used to evaluate our approach:

- micro-averaged recall and precision;
- granularity is used to penalty multiple detections for a single plagiarism case (the higher the worse);
- plagdet—the overall score that is a combination of the previous three measures.

More information about these metrics can be found in (Potthast et al., 2010). Results obtained on the training data are shown in the next table.

Table 3. Results on the training data for text alignment

	Recall	Precision	Granularity	Plagdet
Essays-1	0.848	0.862	1.0011	0.854
Essays-2	0.463	0.824	1.0026	0.591
Generated copy/paste	0.756	0.977	1.41	0.672
Generated paraphrasing	0.706	0.982	1.53	0.614

We can see strong decrease of recall when difficulty of obfuscations is increased for both generated texts and manually written. Also it is clear that our method does not find all cases even for moderately disguised plagiarism. Low recall for generated plagiarism cases is rather surprising. The cause may be that the generated suspicious documents contain duplicate sentences taken multiple times from a single source file. We discard all duplicate sentences in the second stage of our method. Therefore, we can't find all of them.

Since the training data was annotated with the type of obfuscation used when modifying each fragment, we were able to identify the most difficult types of obfuscation for our method. The result for the collection **Essays-1** is presented below.

Table 4. Recall per obfuscation type

	Description	Recall
CCT	concatenation of sentences	0.41
HPR	paraphrasing	0.44
SSP	splitting of sentences	0.65
LPR	moderate modifications (replacing/reordering of words)	0.78
ADD	addition of words	0.85
DEL	deletion of words	0.85
CPY	copy/paste	0.87

This result shows that the most difficult type of obfuscation for our method is concatenation of sentences and paraphrasing. The latter is quite understandable but the former is the limitation of our sentence based approach. The most of such fragments are lost in the third stage, since sentences from a source failed to provide sufficient IDF-overlap. The distribution of recall is similar for the collection **Essays-2**.

Result on the test data were provided by the organizers. They also provided a comparison with the baseline on the same collections.

Table 5. Results on the test data for text alignment

	Recall	Precision	Granularity	Plagdet
Essays-2	0.531	0.82	1.0016	0.644
Baseline: Essays-2	0.076	0.896	1.141	0.128
Generated paraphrasing	0.865	0.981	1.483	0.7
Baseline: generated paraphrasing	0.833	0.97	3.464	0.416
Generated copy/paste	0.859	0.978	1.466	0.702
Baseline: generated copy/paste	0.994	0.961	1.004	0.9744

Our method is better than baseline for all collections except the generated copy/paste collection.

5.3. Evaluation of plagiarism detection method

For evaluating all stages of our method at once, we use collections **Essays-2** and **Essays-1**, since we evaluated on both subtasks. We performed optimization on **Essays-2** collection with the goal to maximize Mean Average Precision (MAP).

Results on the training data are shown in the next table.

Table 6. Results on the training data for the whole method

	Source Retrieval			Text Alignment			
	Recall	**Mean average precision**	**Precision**	**Recall**	**Precision**	**Granularity**	**Plagdet**
Essays-1	0.97	0.754	0.332	0.783	0.904	1.00089	0.839
Essays-2	0.82	0.709	0.652	0.316	0.883	1.00095	0.466

This result shows that our method returns most sources in the top of the search results, since MAP is high relative to precision. It detects about of 80% of moderately disguised text and only a third of the text that was heavily paraphrased.

Similar results were obtained for test data, provided by organizers.

Table 7. Results on the test data for the whole method

	Source Retrieval			Text Alignment			
	Recall	**Mean average precision**	**Precision**	**Recall**	**Precision**	**Granularity**	**Plagdet**
Essays-2	0.83	0.608	0.441	0.382	0.885	1.0015	0.533

6. Conclusion

In this paper, we described our method for plagiarism detection and evaluation of this method in two tracks of PlagEvalRus-2017. The method was performed quite well for various plagiarism cases. The best result was achieved for manually written essays with moderately disguised plagiarism. PlagEvalRus corpus helped to identify some weak points of our method, which we are going to address in future. We also plan to estimate current impact of semantic/syntactic similarity measures on recall, and explore more possibilities to leverage them for detecting heavily disguised plagiarism.

References

1. *Alzahrani S. M., Salim N., Abraham A.* (2012), Understanding plagiarism linguistic patterns, textual features, and detection methods, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 42(2), pp. 133–149.
2. *Elsayed T., Lin J., Oard D. W.* (2008), Pairwise document similarity in large collections with MapReduce, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, pp. 265–268.
3. *Kraus K.* (2016), Plagiarism Detection—State-of-the-art systems (2016) and evaluation methods, available at: http://arxiv.org/abs/1603.03014
4. *Lin W. Y., Peng N., Yen C. C., Lin, S. D.* (2012), Online plagiarism detection through exploiting lexical, syntactic, and semantic information, Proceedings of the ACL 2012 System Demonstrations, pp. 145–150.
5. *Meuschke N., Gipp B.* (2013), State-of-the-art in detecting academic plagiarism, International Journal for Educational Integrity, vol. 9(1), pp. 50–71
6. *Osipov G., Smirnov I., Tikhomirov I., Shelmanov A.* (2013), Relational-situational method for intelligent search and analysis of scientific publications, Proceedings of the Integrating IR Technologies for Professional Search Workshop, pp. 57–64.
7. *Osman A. H., Salim N., Binwahlan M. S., Alteeb R., Abuobieda A.* (2012), An improved plagiarism detection scheme based on semantic role labeling, Applied Soft Computing, vol. 12(5), pp. 1493–1502.
8. *Potthast M., Stein B., Barrón-Cedeño A., Rosso P.* (2010). An evaluation framework for plagiarism detection, Proceedings of the 23rd international conference on computational linguistics: Posters, Beijing, pp. 997–1005.
9. *Shelmanov A. O., Smirnov I. V.* (2014) Methods for semantic role labeling of Russian texts, Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference “Dialogue” No. 13, pp. 607–620.
10. *Suvorov R. E., Sochenkov I. V.* (2015). Establishing the similarity of scientific and technical documents based on thematic significance, Scientific and Technical Information Processing, vol. 42(5), pp. 321–327.
11. *Takuma D., Yanagisawa H.* (2013), Faster upper bounding of intersection sizes, Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp. 703–712
12. *Zubarev D., Sochenkov I.* (2014), Using Sentence Similarity Measure for Plagiarism Source Retrieval, In CLEF (Working Notes), pp. 1027–1034.