EVALUATION OF THE SHAPD2 ALGORITHM EFFICIENCY IN PLAGIARISM DETECTION TASK USING PAN PLAGIARISM CORPUS

Dariusz Ceglarek

Department of Applied Informatics, Poznan School of Banking, Poznan, Poland
dariusz.ceglarek@wsb.poznan.pl

ABSTRACT

This work presents results of the ongoing novel research in the area of natural language processing focusing on plagiarism detection, semantic networks and semantic compression. The results demonstrate that the semantic compression is a valuable addition to the existing methods used in plagiary detection. The application of the semantic compression boosts the efficiency of Sentence Hashing Algorithm for Plagiarism Detection 2 (SHAPD2) and authors' implementation of the w - shingling algorithm. Experiments were performed on Clough & Stephenson corpus as well as *an available PAN–PC-10 plagiarism corpus used to evaluate plagiarism detection methods, so the results can be compared with other research teams.*

KEYWORDS

Plagiarism detection, Longest common subsequence, Semantic compression, Sentence hashing, w-shingling, Intellectual property protection

1. INTRODUCTION

The main objective of this work is to present recent findings obtained in the course of research on more efficient algorithms used in matching longest common subsequences, as well as semantic compression. As signaled in the previous publication introducing Sentence Hashing Algorithm for Plagiarism Detection 2 (*SHAPD2*) [9], this algorithm is capable of providing better results than the most known alternative algorithms operating on hash structures representing fragments (usually *n-grams*) of a text document. By better, author understand a certain set of features that the alternative algorithms cannot deliver along with performance characteristics surpassing known competitors. More details about this characteristics will be provided in next sections of the article.

One of important domains of the longest sequence matching is plagiarism detection. It was observed that a solution that uses text hashing to detect similar documents can benefit greatly from the inclusion of a mechanism that makes it resilient to a number of techniques used by those inclined to commit an act of plagiarism. The existing solutions and systems protecting intellectual property are usually limited to searching for borrowings or plagiarisms in specified (suspicious) documents in relation to documents stored in internal repositories of text documents (document's corpora) and, thus, their effectiveness is greatly limited by the size of their repositories.

Obtained System Protecting of Intellectual Property (*SeiPro2S*) has been designed to protect resources from the unauthorized use of intellectual property, including its appropriation. The most commonly known example of such treatment is plagiarism, which, as is well known, infringes

Sundarapandian et al. (Eds) : CSE, CICS, DBDM, AIFL, SCOM - 2013
pp. 119–133, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3312

intellectual property and is based on misappropriation of another person's work or a portion thereof (other people's creative elements) by hiding the origins of that work. It is difficult to give a precise and universal definition of plagiarism, since the intellectual and artistic achievements of mankind arise from the processing and development of the achievements of predecessors, thus providing information on even the smallest use of one's achievements would lead to the absurd. We are dealing with open plagiarism, which consists of the acquisition of all or part of another person's work by putting one's own name on it. We also have to deal with hidden plagiarism, which takes place when passages from someone else's work are included in one's own work without providing information on the original author and source, and then weaving these passages into one's own arguments. Such a situation may arise as a result of borrowings or literal copying, but can also be based on a mapping of the original document by using different words or concepts yet representing exactly the same semantic content and structure of thought.

There is no universally valid definition of plagiarism - e.g. Polish law does not specify the notion of plagiarism, nevertheless it enumerates a list of offenses directed at intellectual property, such as the following:

- claiming authorship of some piece of work as a whole or its parts
- reusing someone else's work without changes to the narration, examples and the order of arguments
- claiming authorship of a research project
- claiming authorship of an invention to be used in an evaluation of research achievements that is not protected by intellectual property law or industry property ownership law.

Text plagiarism is perhaps one of the oldest forms of plagiarism, which, to this day, remains difficult to be identified in practice. Therefore, a lot of research has been conducted to detect plagiarism automatically. The most popular existing systems using for plagiarism detection in textual documents use knowledge representation structures and methods that are characteristic of Information Retrieval systems processing information for classification purposes at the level of words or strings, without extracting concepts [2, 4]. Moreover, these systems use, as similarity indicators, criteria such as the fact that a coefficient of similarity between documents understood as a set of words appearing in compared documents exceeds several tens of percent and/or the system has detected long identical text passages in a given work. Furthermore, a suspicious document which is analyzed for borrowings might have undergone various stylistic transformations, which cause it to be regarded by such systems as largely or completely different from other documents as far as long and seemingly unique sentences are concerned [3]. This stylistic transformations include such operation as shuffling, removing, inserting, or replacing words or short phrases or even semantic word variation created by replacing words by one of its synonyms, hyponyms, hypernyms, or even antonyms.

For the above-mentioned reasons there is a need to construct such an IT system protecting intellectual property contained in text information resources that would use conceptual knowledge representation as well as methods and mechanisms causing parts of the text of documents which are expressed in a different way but which have the same information value in semantic terms to be understood as identical or at least very similar. In the *SeiPro2S* system the author used a semantic network as a knowledge representation structure because of its ability to accumulate all the knowledge about the semantics of concepts, which makes it usable in systems that process natural language.

Obtained *SeiPro2S* system architecture was first described in [7]. The system has been designed to detect plagiarism (system working in single mode) and to be able to perform tasks such as the protection of intellectual property in documents assigned to the system (monitoring mode). The

system periodically monitors documents listed in the available repositories of documents (e.g. the Internet) and detects documents that violate intellectual property. In such cases a list of permissible locations of document copies is prepared for a protected document.

When an input document δ is presented to the system, responsible component starts a procedure of Internet sampling. Samples are obtained by submitting a number of fragments obtained from the input document. A list of potentially viable documents is prepared basing on the occurrence of necessarily long identical passages. Documents are downloaded and subjected to text-refining. After completing these steps every downloaded document is indexed. For every document an abstract is created and stored in local repository. A key for the whole process procedure follows. At first, the most relevant of previously sampled, indexed and abstracted documents are selected by comparing the abstracts. Then, input document δ is subjected to comparison with every relevant document. As a result of this procedure, a similarity report is constructed.

It conveys following information on the overall similarity of input document δ to every checked document:

- length of the longest identical phrase obtained from the documents from the sampling phase
- similarity indicator, which using percent ratio demonstrates how much of the submitted text is identical with documents coming both from the sampling phase and from local repository coming from earlier checks
- checked text along with markup showing which text fragments are identical to those coming from local repository and current Internet sample.

When *SeiPro2S* acts in monitoring mode some enhancements are introduced. First of all, each analyzed document along with first batch of samples obtained from the Internet is stored in local repository. Thus, when the system is monitoring the Internet it can omit all previously checked documents and focus only on new ones.

Current architecture of the Semantically Enhanced Intellectual Property Protection System *SeiPro2S* is described in details in [12].

The common plagiarism techniques include changing the word order in a plagiarized text, paraphrasing passages of a targeted work and interchanging original words with their synonyms, hyponyms and hypernyms [19]. Not all of the above can be addressed at the moment, but the technique based of synonym usage can be easily detected when one is equipped with sufficiently large semantic network that is a key requirement for the semantic compression.

Throughout the research activities author crafted a majority of the necessary tools and methods vital in order to establish how well an application of the semantic compression should level up the results of plagiarism detection efforts.

In order to provide a reasonable study three approaches were tested in detail:

- Sentence Hashing Algorithm for Plagiarism Detection 2 (*SHAPD2*),
- *w-shingling* using 4-grams,
- *w-shingling* using 6-grams.

Each of them was used to gather results on detection of plagiarized documents. The experiment was twofold: firstly, test data was run across the unmodified implementations of the enlisted approaches, secondly the semantic compression was introduced in several steps gauging at each step the strength of the compression. PAN plagiarism evaluation corpus [16] has been utilized in

order to run a benchmark. PAN–PC corpora have been used since 2009 in plagiarism uncovering contests, gathering researchers from this domain to evaluate their methods in comparison with others. The possibility to use the same data sets and measures allows to perceive the results as reliable. In the Pan-PC corpus for each plagiarized documents there is additional information: set of source documents plagiarized, annotations of passages plagiarized from the sources and logs of queries posed by the writer while writing the text.

The *SHAPD2* algorithm has been already evaluated using Clough & Stephenson corpus [15] for plagiarism detection and the results published in [10]. The results proved that the idea of combining sentence hashing and semantic compression improves method's performance in terms of both efficiency and results. Documents in the Clough & Stephenson corpus are, unfortunately, small (text lengths range only from 200 to 300 words, which is hardly more than 2-3 paragraphs). The choice of topics in theClough & Stephenson corpus is very narrow. Also, the sources to plagiarize were given up front so that there is no data on retrieving them.

The first evaluation positions *SHAPD2* as effective as *w-shingling*, or even more for specific cases, while overrunning *w-shingling* when considering run time of comparisons of bigger document sets. In the course of experiments, Vector Space Model has been checked, too, and combined with semantic compression turned out to be ineffective when considering task precision - too many documents appeared similar when processed with semantic compression enabled. There should be understand, that not all the alterations are possible to be discovered by the semantic compression. In the current form, used semantic networks do not store data on whole phrases that can be synonym or hyponym of any other given phrase. Such an extension should provide even better results, yet the amount of work needed to craft such a resource is at the moment beyond the grasp of the research team.

The results obtained throughout the experiments show a great improvement of the *SHAPD2* algorithm augmented with semantic compression over w-shingling. What is more, time wise performance of the *SHAPD2* is far better than the *w-shingling*.

Article is structured as follows: related work section is given where the most important algorithms concerned with a longest common sequence finding are reviewed briefly. The discussion is accompanied by description of plagiarism detection methods and some of the most important initiatives addressing the issue of similarity of documents to one another. Following that, there is a brief presentation of the *SHAPD2* and the semantic compression. The next section is devoted to the experiments, test corpus used, implementations of benchmarked algorithms and the obtained results. All is summarized in the final section extended with plans of future research work.

2. RELATED WORK

The core of the work presented in this article is dependent on the *SHAPD2* algorithm that allows for robust and resilient computation of a longest common subsequence shared by one or many input documents. The task of matching a longest common subsequence is an important one in many subdomains of Computer Science. Its most naive implementation was deemed to have a time complexity of $O(m_1 \cdot m_2)$ (where m_1 and m_2 are the numbers of concepts in compared documents). The question whether it is possible to achieve significantly better results was stated first by Knuth in [14]. First affirmative answer was given in [13] with time complexity $O((m_1 \cdot m_2)/\log(m_2))$ for case when $m < n$ and they pertain to a limited sequence. Another special case with an affirmative answer was given in [20] with time complexity of $O((m_1 + m_2) \cdot \log(m_1 + m_2))$. As to be detailed later, the presented algorithm is another example of a special case where the time complexity is lower than quadratic.

One of the most important implementations of a search for the longest common subsequence is to be found in [21]. This work presents an application of Smith-Waterman algorithm for matching a longest common subsequence in textual data using dynamic programming idea for solving complex problems by breaking them down into simpler subproblems. This is a top achievement of algorithms that do not operate with text frames and their hashes. Other works such as [18] or [23] prove that better efficiency is yielded rather by careful engineering strategies than a fundamental change in time complexity.

All of the above cited works use algorithms which time complexity is near quadratic which results in drastic drop of efficiency when dealing with documents of considerable length.

It was first observed in [24] that introduction of a special structure that was later known as a shingling (a continuous sequence of tokens in a document) can substantially improve the efficiency of deciding on the level of similarity of two documents by observing a number of common shinglings. Following works such as [1, 5] introduce further extensions to the original idea. A number of works represented by publications such as [13] provided plausible methods to further boost measuring of the similarity between entities.

The importance of plagiarism detection is recognized in many publications. One might argue that, it is an essential task in times, where access to information is nearly unrestricted and culture for sharing without attribution is a recognized problem (see [6] and [28]).

With respect to plagiarism obfuscation further explanations are necessary. Plagiarists often paraphrase or summarize the text they plagiarize in order to obfuscate it, i.e., to hide their offense. In the PAN plagiarism corpus a synthesizer, that simulates the obfuscation of a section of text sx in order to generate a different text section sq to be inserted into dq, has been designed on the basis of the following basic operations [29]:

- Random Text Operations. Given *sx*, *sq* is created by shuffling, removing, inserting, or replacing words or short phrases at random.
- Semantic Word Variation. Given *sx*, *sq* is created by replacing each word by one of its synonyms, hyponyms, hypernyms, or even antonyms.
- POS-preserving Word Shuffling. *sq* is created by shuffling words while maintaining the original sequence of parts of speech in *sx*.

It is obvious that these operations do not guarantee the generation of human-readable text. However, automatic text generation is still a largely unsolved problem which is why we have approached the task from the basic understanding of content similarity in information retrieval, namely the bag-of-words model.

3. SHAPD2 AND SEMANTIC COMPRESSION

3.1 Sentence Hashing Algorithm for Plagiarism Detection - SHAPD2

The *SHAPD2* algorithm focuses on whole sentence sequences, calculating hash-sums for them. It also utilizes a new mechanism to organize the hash-index as well as to search through the index. It uses additional data structures such as correspondence list CL to aid in the process. The detailed description of subsequent steps of the *SHAPD2* algorithm formulated in pseudocode can be found in [9].

The *SHAPD2* algorithm works with two corpora of documents comprise the algorithm's input: a corpus of source documents (originals) $D = \{d_1, d_2, ..., d_n\}$, and a corpus of suspicious documents to be verified regarding possible plagiaries, $P = \{p_1, p_2, ..., p_r\}$.

First of all, each document has to undergo a set of transformations in the text-refinement process what is standard procedure in Natural Language Processing (NLP) or Information Retrieval (IR) task. The process of text refinement in the documents starts from extracting lexical units (tokenization), and further text refinement operations include elimination of words from the so-called information stop-list, identification of multiword concepts, and bringing concepts to the main form by lemmatization or stemming. The whole process of text refinement is an especially difficult task for highly flexible languages, such as Polish, Russian or French (with multiple noun declination forms and verb conjugation forms). The last step in this procedure is the concept disambiguation (i.e. choosing right meaning of a polysemic concept). As an output of the text-refinement process the system produces a vector containing ordered concept descriptors coming from documents.

Then, all documents need to be split into text frames of comparable length – preferably sentences, or in the case of longer sentences – split into shorter phrases (long passages of text without a full-stop mark such as different types of enumerations, tables, listings, etc.). A coefficient α is a user-defined value which allows to set the expected number of frames that a longer sentence is split into. The coefficient ranges from 6 to 12 concepts. In the next step, a hash table T is created for all documents from corpus D, where for each key the following tuple of values is stored: $T[k_{i,j}] =< i, j >$, (document number, frame number). A correspondence list CL is declared, with elements of the following structure: n_d – document number, m_l – local maximum, and n_l – frame number for local sequence match. For documents from the corpus P are also created indexes containing hash values for all frames coming from suspicious documents. As a result, every document from original corpus, as well as all suspicious documents, is represented by index as a list of sentence hashes.

Another data structure is the maxima array TM for all r documents in corpus P, containing records with structure as follows: m_g – global maximum, n_g – frame number with global sequence match.

The comparison between corpus D and corpus P is performed sequentially in phase 2 for all documents from corpus P. In phase 2 for all documents d_i from corpus P (containing suspicious documents), the correspondence list CL and maxima array TM are cleared. For each frame, set of tuples is retrieved from hash index table T. If there are any entries existing, it is then checked whether they point to the same source document and to the previous frame. If the condition is true, local correspondence maximum is increased by one. Otherwise, the local maximum is decreased. After all of the frames are checked, table TM storing the correspondence maxima is searched for records whose correspondence maxima are greater than a threshold set e (the number of matching frames to be reported as a potential plagiarism). Frame and document number is returned in these cases.

The important distinction between solution mentioned in the related works section and *SHAPD2* is the emphasis on a sentence as the basic structure for a comparison of documents and a starting point of a procedure determining a longest common subsequence. Thanks to such an assumption, *SHAPD2* provides better results in terms of time needed to compute the results. Moreover, its merits does not end at the stage of establishing that two or more documents overlap. It readily delivers data on which sequences overlap, the length of the overlapping and it does so even when the sequences are locally discontinued. The capability to perform these, makes it a method that can be naturally chosen in the plagiarism detection.

In addition, it implements the construction of hashes representing the sentence in an additive manner, thus word order is not an issue while comparing documents.

It is worth mentioning, that *SHAPD2* is more time efficient than *w-shingling*, while giving highly comparable results. Quality of the similarity measures is discussed in the experiments' part, and execution times for both algorithms are presented in Table 1.

3.2 Semantic compression

The idea of the global semantic compression has been introduced by the author in 2010 in a paper [8] as a method of improving text document matching techniques both in terms of effectiveness and efficiency. Compression of text is achieved by employing a semantic network and data on concepts frequencies (in form of frequency dictionaries). The least frequent concepts are treated as unnecessary and they are replaced with more general concepts (their hypernyms stored in semantic network). As a result, a reduced number of concepts can be used to represent a text document without significant information loss, which is important from a perspective of processing resources (especially when one would like to apply a Vector Space Model - very popular approach in Information Retrieval and text mining systems [17] or [25]). Another feature of the emphasized concept level allows for capturing of common meaning expressed with differently worded sentences [22].

n	1000	2000	3000	4000	5000	6000
w- shingling	5.680	8.581	11.967	16.899	23.200	50.586
SHAPD2	4.608	5.820	7.125	7.527	8.437	8.742

Table 1. Processing time [s] for comparing 3000 documents with a corpus of n documents.

Source: own elaboration

The semantic compression combines data from two sources: concepts frequencies from frequency dictionaries, and concept hierarchy from a semantic network. Usually, one extensive semantic network is used for a given language (e.g. WiSENet [11] semantic network converted from *WordNet* for English [27], SenecaNet for Polish [7]) and thus it is able to include linguistic knowledge covering multiple domains.

To give an illustrative example, consider a document originating from nature studies where a life of common rodents is discussed. On the other hand, let us consider document from Information Technology focused on Human Computer Interfaces. Both documents have passages where a concept 'mouse' is to be found. The semantic compression allow for generalizing 'mouse' into two different less specific concepts, different for every domain. In nature studies one may generalize mouse as a rodent and while dealing with it in Information Technology one would use pointing or electronic device. It should be remembered that semantic compression is a lossy one. Yet, the loss of information is minimal by selecting the least frequent words and replacing them by more general concepts, so their meaning remain as similar to the original as possible. The compression ratio can be tuned easily, by setting a number of concepts to be used to describe text documents. Experiments, that were conducted to measure quality of the method in NLP tasks showed, that the number of concepts can be reduced to about 4,000 or even 3,000 without significant deterioration of classification results.

4. EXPERIMENTS

The algorithm's performance has been evaluated using the PAN–PC plagiarism corpus [16], which is designed for such applications. Available test data sets, source and suspicious documents, have been downloaded and made available for the implementation of the algorithm.
A series of program executions have been run in the course of the experiment with different level of semantic compression set. This enabled to adjust the compression strength to gain optimum results.

In order to reliably compare the results with other methods submitted to the PAN-PC competitions in plagiarism detection, the same measures needed to be employed in the evaluation. PAN-PC competitions use the following indicators to determine methods' performance: precision, recall, granularity and *plagdet* score.

—precision and recall are standard measures used in Information Retrieval [2, 25] and calculated here in a respective way. Precision is determined as a ratio of correct plagiarism detections to a total number of reported potential plagiarized text fragments. Recall is a ratio of correctly reported plagiarism cases to a total number of plagiaries, existing in the corpus. An aim is to achieve both measures as close to 1.00 as possible.

$$— \ precision = \frac{r_s}{|R|} \qquad (1)$$

$$recall = \frac{r_s}{|S|} \qquad (2)$$

where: r_s is a number of correct plagiarism detections, R is a set of reported suspicious plagiarism cases, S is a set of plagiarism cases.

— as it's possible, that some methods can report one plagiarized text passage as multiple plagiarism detections, a measure of granularity is introduced. It can be quantified as an average number of reported plagiarisms per one plagiarized text passage. The *SHAPD2* algorithm is always reporting subsequent or overlapping detections as a single plagiarism detection, hence achieving granularity of 1.00.

— in order to balance precision, recall and granularity in one synthetic indicator, a *plagdet* score has been introduced by PAN-PC authors. *Plagdet* is calculated from the following formula:

$$— \ \text{plagdet} = \frac{\text{H}}{\log_2(\text{granulari ty})} = \frac{2 \cdot precision \cdot recall}{(precision + recall) \cdot \log_2(\text{granulari ty})} \qquad (3)$$

—

where H is a harmonic mean of precision and recall.

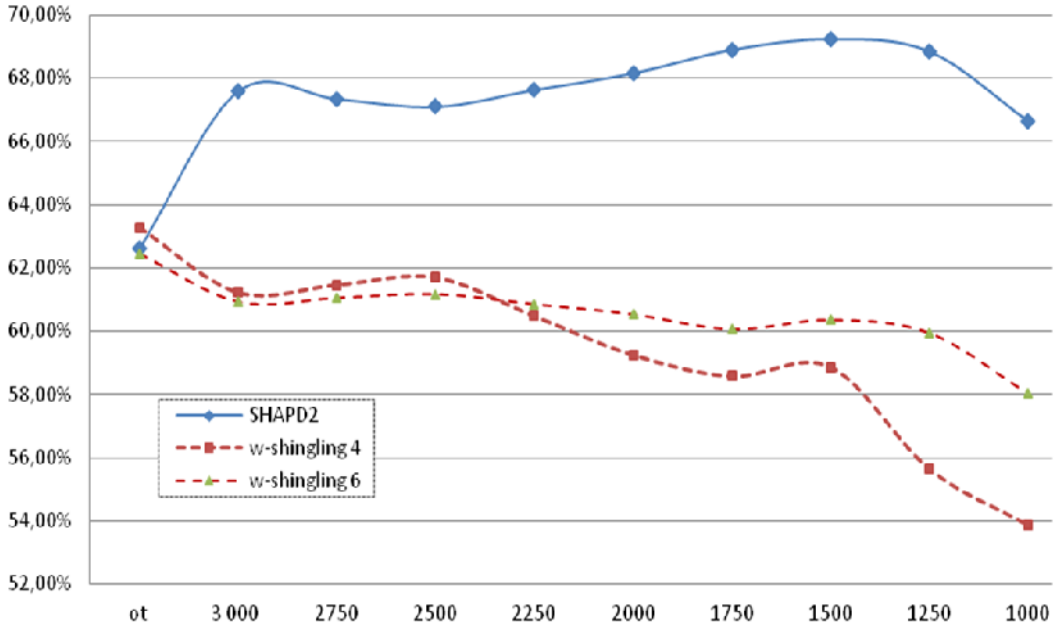

Figure 1. Comparison of synthetic *plagdet* indicator: SHAPD2 versus *w-shingling* 4-grams and 6-grams - for all cases of plagiarism in the PAN-PC corpus. Source: own elaboration

In order to verify whether the semantic compression is a valuable addition to the already available technologies a set of test runs was performed on the mentioned corpus. The results are given in Tables 1, and 2. As one can easily see, the results prove that *SHAPD2* cope better with the task than *w-shingling*. Figure 1 shows the difference in *plagdet* indicator value between the *SHAPD2* and *w-shingling* method using 4-grams and 6-grams for all cases. Figure 2 shows measure of recall for simulated plagiarism cases using *SHAPD2* algorithm and *w-shingling* with 4-grams and 6-grams. On Figure 2 it easily can be seen that value of precision of plagiarism detection was better for SHAPD2 then for *w-shingling* using 4-grams and 6-grams in each cases.

Method	PlagDet	Precision	Recall	Granularity
SHAPD2 (original text)	0.623	0.979	0.401	1.00
SHAPD2 (semantic compression 1500)	0.692	0.506	0.945	1.00
w-shingling (4-grams)	0.633	0.955	0.419	1.00
w-shingling (6-grams)	0.624	0.964	0.404	1.00

Source: own elaboration

Another step in the experiments was an introduction of the semantic compression of varying strength (understood as a number of concepts from the semantic network that were allowed to appear in final data). An example of textual data from this step are provided in Table 3.

Semantic compression	PlagDet	Precision	Recall	Granularity
none	0.626	0.979	0.401	1.00
3000	0.676	0.974	0.469	1.00
2750	0.673	0.970	0.468	1.00
2500	0.671	0.966	0.466	1.00
2250	0.676	0.961	0.476	1.00
2000	0.682	0.957	0.486	1.00
1750	0.689	0.949	0.500	1.00
1500	0.692	0.948	0.506	1.00
1250	0.689	0.933	0.508	1.00
1000	0.667	0.917	0.485	1.00
750	0.650	0.877	0.482	1.00

Table 2. SHAPD2 performance with different thresholds of semantic compression.

Source: own elaboration

Source document (fragment)
ce of change in the character of the country within twenty or thirty miles was visible, and we had provisions left (not having expected the stream to extend so far), and the camp at sixty miles distar d to leave the farther examination of the river to some future explorers; but we regretted it the less of the gravel and sand brought down by the stream, there seemed great probability that it takes its marshes similar to those known to exist 100 miles east of the Irwin.
Subject document (fragment)
As no appearance of change in the character of the gravel sand brought down by the stream, there probability that it takes its rise in large salt marshes similar to those known to exist 100 miles eas Resumed our commute; passed two parties of natives; a few of them constituted us some distance, and having overcome their first surprise, commenced talking in own language explorers; but we regretted it the less that, from the nature of the country within twenty or thirty miles was visible, and we had actually two days' provisions left (not having expected the stream to extend merely so), and the camp of seven hours we arrived at the river.
Comparison between original and subject document after semantic Compression
appearance change character country twenty thirty mile visible food left expected stream run camp adjective mile duty-bound leave adjective investigation river future someone regret nature cover courage bring stream seem suitable probability take rise large salt land similar known exist mile east Irvin.

Table 3. Example for a sample document (suspicious-document02234.txt) compared to original article (source-document02313.txt) before and after semantic compression.

Source: own elaboration

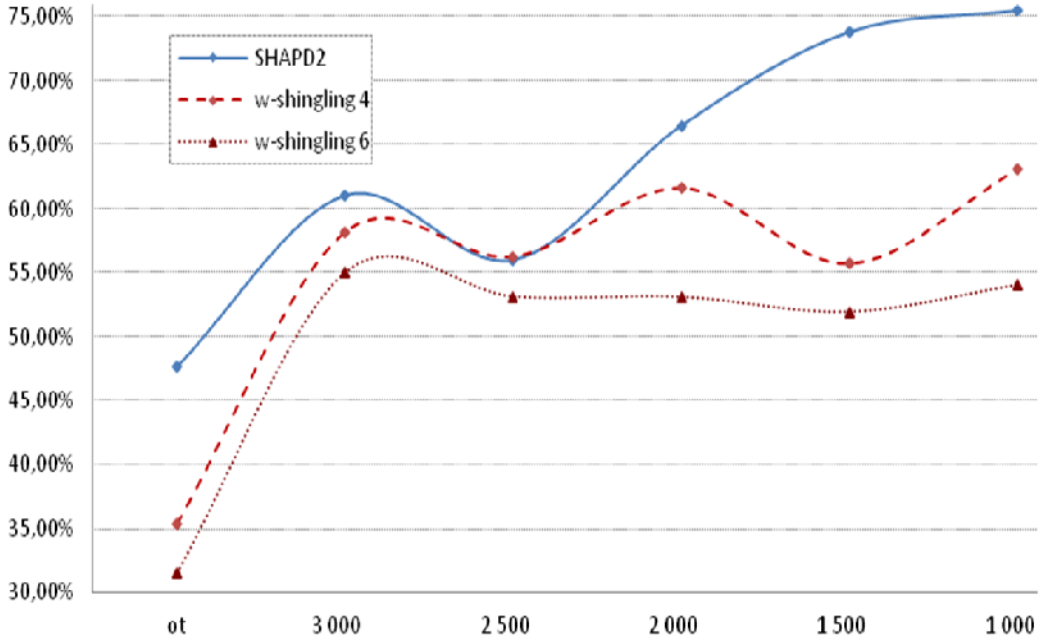

Figure 2. Comparison: *SHAPD2* algorithm versus *w-shingling* algorithm using 4-grams and using 6-grams - recall for simulated plagiarism cases in the PAN-PC corpus. Source: own elaboration

Figure 3 shows the difference in the value of recall in the PAN-PC corpus of with the use of *SHAPD2* and *w-shingling* algorithms. Is easy to see on figures 1, 2 and 3 that the algorithm *SHAPD2* produces both better results of recall and a synthetic indicator *plagdet* than *w-shingling*.

The results of corpus matching using the *SHAPD2* method employing the semantic compression with a compression force threshold set to 1000 are especially worth noticing. Clough & Stevenson test corpus [15] contains a subset of non-plagiarism documents (i.e. participants were to read and rewrite the article using their knowledge, without any possibility to copy or modify the original), which stands for 40% of the corpus. The remaining documents were created by copying original articles and revising them to a certain extent: exact copy, light revision, heavy revision - 20% of documents each. The results presented in Table 4 show, that the method employing the semantic compression allows for achieving the results structure very similar to the original one.

Table 3 shows the distribution of similarity calculated for 3 algorithms: commonly known Vector Space Model (VSM), *w-shingling* and *SHAPD2*. The distribution was evaluated for these algorithms when they are not used semantic compression, and when semantic compression was enabled - using 1000 and 2000 concepts as descriptors.
In VSM method it easily can be seen, that too many documents have been evaluated as a light revision or near copy (even without using semantic compression mechanism). We expected there are 20% and 40% of hits, while VSM method estimated to this group almost all documents.

W-shingling algorithm is more careful, but when compression is enabled. It easily can be seen that for the plagiarism is considered more and more documents – in compression level of 1000 concepts more than the expected 60% (too many false-positive) - because it is only 30% as non-plagiarism.

The *SHAPD* algorithm – as *w-shingling* - carefully evaluates documents, but when semantic compression is enabled, we get a close similarity to the expected distribution.

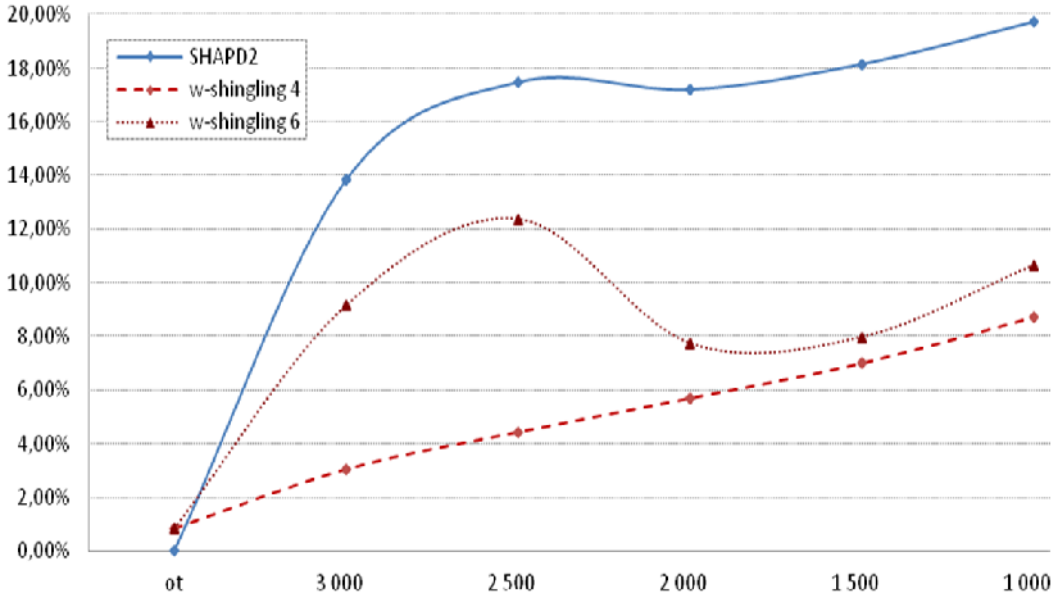

Figure 3. Comparison: *SHAPD2* algorithm versus *w-shingling* algorithm using 4-grams and using 6-grams - recall for translated cases of plagiarism. Source: own eleboration

One of the test cases, which reveals a major improvement in recognizing an evident plagiarism is demonstrated in Table 4. The semantic compression enabled to calculate similarity measures at around 0.6, while both methods (*w-shingling* and *SHAPD2*) without semantic compression returned relatively low similarity measures (about 0.3).

Method	Similarity	Expected division	without semantic compression	Semantic compression level 2000	Semantic compression level 1000
VSM	non plagiarism	0.40	0.00	0.00	0.02
	heavy revision	0.20	0.02	0.01	0.01
	light revision	0.20	0.30	0.36	0.46
	near copy	0.20	0.68	0.63	0.51
w-shingling	non plagiarism	0.40	0.53	0.36	0.30
	heavy revision	0.20	0.17	0.20	0.23
	light revision	0.20	0.16	0.21	0.23
	near copy	0.20	0.14	0.23	0.24
SHAPD	non plagiarism	0.40	0.59	0.46	0.40

	heavy revision	0.20	0.13	0.16	0.19
	light revision	0.20	0.14	0.17	0.18
	near copy	0.20	0.15	0.21	0.23

Table 4. Distribution of similarity calculated for 3 algorithms: VSM, *w-shingling* and *SHAPD2.*

Source: own elaboration

Method	without semantic compression	Semantic compression	
		level 2000	level 1000
w-shingling	0.271	0.567	0.585
SHAPD2	0.229	0.433	0.508

Table 5. Evident plagiarism detection calculated for *w-shingling* and *SHAPD2* algorithms.

Source: own elaboration

Generalization, which is a key operation in semantic compression, entails minimal information loss, which cumulates together with reducing target's lexicon size. It does not appear to deteriorate results of the experiments employing *SHAPD2* or *w-shingling* method (see Table 5). It is especially visible in cases where the strong semantic compression was enabled (target lexicon size below 1500 concepts), causing some non-plagiarized articles on the same subject to be marked as highly similar and recognized as possible plagiaries (cf. Table 5). This suggest, that for certain use cases the compression threshold needs to be set carefully. Establishing a proper level of compression threshold is especially important in classification tasks of text documents. Further research is necessary to identify the optimum compression threshold for individual information retrieval tasks.

5 SUMMARY

The conducted research can be summarized as follows:

– The *SHAPD2* algorithm can be successfully employed in plagiarism detection systems, giving results of competitive quality when compared to *w-shingling*, while performing substantially better when taking into account time efficiency of the algorithms.

– Enabling semantic compression in *w-shingling* and *SHAPD2* methods improves the quality of plagiarism detection significantly. A combination of *SHAPD2* and semantic compression used for plagiarism detection in Clough & Stephenson corpus returns results which structure is very close to experts' assessment.

– The *SHAPD2* algorithm produces both better results of recall and a synthetic indicator plagdet than w-shingling - the results prove that *SHAPD2* cope better with the task than *w-shingling*. However, the results of the first attempt of using *SHAPD2* to detect plagiarism in the PAN-PC-10 corpus are still below the best system in the original PAN-PC-10 task (achieved results of 0.797 of the *plagdet* indicator).

– Vector Space Model is commonly regarded as a good solution for a detection of duplicates or near duplicates (very similar documents), but it returns a significant number of false-positives, so this method is not recommended for plagiary detection.
– For certain information retrieval tasks, semantic compression strength needs to be adjusted carefully in order not to lose too much information, which may lead to deterioration of retrieval precision. Semantic compression threshold set to a level 1500–1250 words in the target lexicon seems to be a safe value when used for plagiarism detection.

In the near future, author plans to further develop various algorithms and available knowledge representation structures so that new applications for the semantic compression and *SHAPD2* can be devised.

What is more, there are plans for reorganization of available assets so that the semantic compression can be applied in a automated manner to text passages without introduction of hypernyms disrupting user's experience. this might be achieved by introduction of information on concept relevance in current culture and prohibition on archaic concepts.

REFERENCES

[1] Andoni, A., Indyk, P. (2008), *Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions*, Commun. ACM, 51(1). pp. 117–122
[2] Baeza-Yates, R, Ribeiro-Neto, B. (1999), *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., New York
[3] Blackburn, P., Bos, J. (2005), *Representation and Inference for Natural Language : A First Course in Computational Semantics*, CSLI Publications
[4] Brachman, R. J., Levesque, H. J. (2004), *Knowledge Representation and Reasoning*, Elsevier Press
[5] Broder, A. Z., (1997), *Syntactic clustering of the web*, Computer Networking ISDN Systems, vol. 29 (8-13), pp. 1157–1166
[6] Burrows, S., Tahaghoghi, S.M., Zobel J. (2007), *Efficient plagiarism detection for large code repositories*, In: Software: Practice and Experience, vol. 37 (2), pp. 151–175
[7] Ceglarek, D., Haniewicz, K., Rutkowski (2009), *Semantically Enhanced Intellectual Property Protection System – SEIPro2S*, In: 1st International Conference on Computational Collective Intelligence, in: Lecture Notes on Computer Science, pp. 449–59, Springer Verlag, Berlin Heidelberg
[8] Ceglarek, D., Haniewicz, K., Rutkowski, W. (2010), *Semantic compression for specialised information retrieval systems*, In: Nguyen N. Th., Katarzyniak R., and Chen S.M., editors, Advances in Intelligent Information and Database Systems, vol. 283 of Studies in Computational Intelligence, pp. 111–121, Springer Verlag, Berlin Heidelberg
[9] Ceglarek, D. (2013), *Linearithmic Corpus to Corpus Comparison by Sentence Hashing*, In: The 5th International Conference on Advanced Cognitive Technologies and Applications - COGNITIVE 2013, pp. 12, Xpert Publishing Services, Valencia (Spain)
[10] Ceglarek, D., Haniewicz, K., Rutkowski, W. (2012), *Robust Plagiary Detection Using Semantic Compression Augmented SHAPD*, In: Nguyen N.T., Hoang K., Jedrzejowicz P. (Eds.), Computational Collective Intelligence. Technologies and Applications , Lecture Notes in Artificial Intelligence, vol. 7653, pp. 308–317, Springer Verlag, Berlin Heidelberg
[11] Ceglarek, D., Haniewicz, K., Rutkowski, W. (2011), *Towards knowledge acquisition with WiseNet*, In: Nguyen N. Th., Trawinski B., and Jung J.J., editors, ACIIDS Posters, volume 351 of Studies in Computational Intelligence, pp. 75–84. Springer Verlag, Berlin Heidelberg
[12] Ceglarek, D. (2013), *Architecture the Semantically Enhanced Intellectual Property Protection System* ,In: Lecture Notes in Artificial Intelligence - Computer Recognition System 5, pp. 10, Springer Verlag, Berlin Heidelber
[13] Charikar, M. S. (2002), *Similarity estimation techniques from rounding algorithms*, In Proceedings of the 34th annual ACM symposium - STOC'02. pp. 380–388
[14] Chvatal, V., Klarner, D. A., Knuth, D. E. (1972), *Selected combinatorial research problems*, Technical report, Stanford, CA, USA
[15] Clough, P., Stevenson, M. (2009), *A Corpus of Plagiarized Short Answers*. University of Sheffield [http://ir.shef.ac.uk/cloughie/resources/plagiarism corpus.html], access January 1 2012

[16] Corpus PAN-PC on line: http://pan.webis.de/, access March 11 2013
[17] Erk, K., Pad, S. (2008), *A Structured Vector Space Model for Word Meaning in Context* , pp. 897-906, ACL
[18] Grozea, C., Gehl, Ch., Popescu, M.(2009), *Encoplot : Pairwise sequence matching in linear time applied to plagiarism detection*, Time, pp. 10–18
[19] Hoad, T., Zobel, J. (2003), *Methods for Identifying Versioned and Plagiarised Documents*, *Journal of the American Society for Information Science and Technology*. *vol.* 54 (3), pp. 203–215
[20] Hunt, J., Szymanski, T. (1977), *A fast algorithm for computing longest common subsequences*, Communications of the ACM 20 (5), pp. 350–353
[21] Irving, R. W. (2004), *Plagiarism and collusion detection using the Smith-Waterman algorithm*, Technical report, University of Glasgow
[22] Keikha, M., Razavian, N. S., Oroumchian, F., Razi, H.S. (2008), *Document Representation and Quality of Text: An Analysis*, red. M. W. Berry & M. Castellanos [w]: Survey of Text Mining II: Clustering, Classification, and Retrieval, pp. 219-232, Springer Verlag, Berlin Heidelberg
[23] Lukashenko, R., Graudina, V., Grundspenkis, J. (2007), *Computer-based plagiarism detection methods and tools: an overview*. In: Proceedings of the 2007 international conference on Computer Systems and Technologies, CompSysTech '07, pp. 40:1–40:6, ACM, New York, NY, USA
[24] Manber, U. (1994), *Finding similar files in a large file system*, Proceedings of the USENIX Winter 1994 Technical Conference on USENIX, WTEC'94
[25] Manning, C. D., Raghavan, P., Schutze, H. (2008), *Introduction to Information Retrieval*, Cambridge University Press
[26] Masek, W.J., Paterson, M.S.: *A faster algorithm computing string edit distances*, In: Journal of Computer and System Sciences, vol. 20, pp. 18–31, Elsevier (1980)
[27] Miller, G. A. (1995), *WordNet: a lexical database for English*, Communication of the ACM, Volume 38 (11), pp. 39-41
[28] Ota, T., Masuyama, S. (2009), *Automatic plagiarism detection among term papers*, In: Proceedings of the 3rd International Universal Communication Symposium, IUCS '09, pp. 395–399, ACM, New York, NY, USA
[29] Potthast, M., Stein, B., Barrn-Cedeo, A., Rosso, P. (2010), *An Evaluation Framework for Plagiarism Detection*, In: Proceedings of 23rd International Conference on Computational Linguistics - COLING, pp. 997–1005, Beijing
[30] Potthast, M., Gollub, T., Hagen, M., Kiesel, J., Michel, M., Oberlander, A., Tippmann, M., Barrn-Cedeor, A., Gupta, P., Rosso, P., Stein, B. (2012), *Overview of the 4th International Competition on Plagiarism Detection*, In: P. Forner, J. Karlgren, and Ch. Womser-Hacker, editors, CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers. p. 28

Author

Dr. D. Ceglarek is a Assistant Professor of Information Technology in Poznan School of Banking, Poznan, Poland. He earned his Ph.D degree in Clustering of information from Department of Information Systems, Poznan University of Economics, Poznan, Poland in 1997. He has published more than thirty of research papers in the area of classification of information, plagiarism detection, natural language processing and semantic technology. He has served as Assistant Professor in Department of Information Systems, Poznan University of Economics during 1997 to 2007.