A Deep Learning Approach to Persian Plagiarism Detection

Erfaneh Gharavi
University of Tehran
Faculty of new Science and Technology
Data & Signal processing Lab
e.gharavi@ut.ac.ir

Kayvan Bijari
University of Tehran
Faculty of new Science and Technology
kayvan.bijari@ut.ac.ir

Kiarash Zahirnia
University of Tehran
Faculty of new Science and Technology
zahirnia.kia@ut.ac.ir

Hadi Veisi
University of Tehran
Faculty of new Science and Technology
Data & Signal processing Lab
h.veisi@ut.ac.ir

ABSTRACT

Plagiarism detection is defined as automatic identification of reused text materials. General availability of the internet and easy access to textual information enhances the need for automated plagiarism detection. In this regard, different algorithms have been proposed to perform the task of plagiarism detection in text documents. Due to drawbacks and inefficiency of traditional methods and lack of proper algorithms for Persian plagiarism detection, in this paper, we propose a deep learning based method to detect plagiarism. In the proposed method, words are represented as multi-dimensional vectors, and simple aggregation methods are used to combine the word vectors for sentence representation. By comparing representations of source and suspicious sentences, pair sentences with the highest similarity are considered as the candidates for plagiarism. The decision on being plagiarism is performed using a two level evaluation method. Our method has been used in PAN2016 Persian plagiarism detection contest and results in %90.6 plagdet, %85.8 recall, and % 95.9 precision on the provided data sets.

CCS Concepts

• Information systems → Near-duplicate and plagiarism detection • Information systems → Evaluation of retrieval results.

Keywords

Deep Learning; Word Vector Representation; Persian Plagiarism Detection.

1. INTRODUCTION

Due to the growth and expansion of the global networks and the increasing volume of unstructured data by both men and machine, an automated intelligent processing and knowledge extraction system is required. The primary goal of language processing methods is to achieve direct human computer interaction as the main purpose of artificial intelligent [26]. Natural language processing (NLP) encompasses wide variety of tasks and applications including: part of speech tagging (POS), text classification, machine translation, text similarity detection, and etc. One well-known application of text similarity detection is to identify plagiarism especially for scientific documents. Plagiarism is defined as the act of taking someone else's works or ideas and presenting them as one's own without explicitly acknowledging the original source which is considered immoral and illegal [9]. In this regard, detection and prevention such duplications has vital importance.

In order to be processed in natural language processing algorithms, textual data should be numerically described. In traditional approaches, list of the words are considered as distinct features for the textual data. In such methods, the similarity between the synonym words is not taken into account. Furthermore, due to the sparseness of new feature space and time complexity of feature extraction, these approaches are not efficient [5]. To overcome deficiencies of the traditional feature extraction methods, deep learning techniques are used which have resulted in promising performance in many application such as NLP [11]. The essential goal of deep learning [19] is to improve the processing, and pre-processing methods of NLP in an automatic, efficient, and fast way. In text mining applications, deep learning methods represent words as a vector of numerical values [9]. This new representation contains a major part of synthetic as well as semantic rules of the text data. In applications such as similarity detection and text classification, much larger units such as phrases, sentences and documents should be described as a vector. For this purpose, there are a number of methods ranging from simple mathematical approaches [30] to neural networks-base combination functions [36]. Vectorized representation of text data makes it easy to compare words and sentences as well as minimizing the need to use lexicons. In this paper, deep learning approach is used for Persian plagiarism detection in PAN plagiarism detection contest. This method results in %90.6 plagdet, %85.8 recall, %95.9 precision on the PAN provided data sets.

Rest of this paper is organized as follow: in Section 2 we described plagiarism and the act of plagiarism detection, followed by presenting related works in Section 3.Section 4 is devoted to illustrate deep learning and the approach of using it in NLP applications. Section 5 defines proposed method and Section 6 demonstrates the experimental results. Finally we explain privileges of our methods in Section 7.

2. PLAGIARISM DETECTION

Plagiarism is an attempt to use the other's idea and present it as your personal work, which is considered both illegal and immoral. The era of the internet and quick access to wide range of

information, exacerbates acts such as plagiarism. Plagiarism is being done in various ways, and often it is difficult to prove whether a text is plagiarized or not. Previously, the plagiarism was detected only manually and based on the reviewer's knowledge. But nowadays, due to the difference between human cognition and vast amount of information, the process of plagiarism detection is very challenging to be performed manually. Therefore, automated plagiarism detection gets wide attention in the recent years [8, 9].

In 2000, only 5 systems have been developed for the purpose of plagiarism detection, four of which was used to detect plagiarism in text and one system was used to detect copied programming codes [22]. This number growth to 47 in 2010 which indicates an increase in demand of such systems as well as the need to improve speed and efficiency. It should be noted that previous approaches often benefit from string matching scheme in order to detect copied texts. The inadequacy of existing systems leads the research direction to new approaches for plagiarism detection. The main drawback in this area is system's inability to recognize the syntactic and semantic changes in the text data. Although it seems very simple for human beings, but the computer is facing many difficulties in this detection, especially when the detection is dependent on exact text matching. Plagiarism detection steps is outlined in the below algorithm.

Algorithm: Plagiarism Detection steps

- Data pre-processing: preparation of the input data including original and plagiarized text.
- Similarity comparison: In this step, texts from original and plagiarized source are compared based on a similarity measure. The output of this step is a rate which indicates the similarity of the input texts.
- Filtering: based on a predefined threshold, the generated rates in the previous step are used to identify candidate pairs.
- Further processing: at this point, pairs are evaluated base on other similarity measures.
- Classification: The final step is to assign a label indicating whether the texts are plagiarized or not. This can be done using the calculated rate resulted from the 4-th step.

Scientific plagiarized text comprises of word sequences including n-grams which are exactly the same or paraphrased form of the original text. This sequence of words can be in different lengths to include whole or a part of the original documents. Examples of rules that show how the plagiarism in scientific fields is occurred, are provided in the following [27].

- Inadequate referencing
- Direct copy from one or more sources of text
- Displacement of words in a sentence
- Paraphrase and rewrite the texts, present other's ideas with different words
- Translation, expression of an idea in one language into another one

Plagiarism can include changes in the vocabulary, or syntactic, and semantic representation of the text. These types will be discussed further in the following:

Vocabulary changes: Including the addition, deletion or replacement of words in a given text. Such changes would be indistinguishable by string matching approach.

Synthetic changes: Changes in the structure includes rearranging words and expressions, and turning sentences from active to passive and vice versa.

Semantic changes: This kind of plagiarism is more fundamental and usually includes paraphrase as well as semantic and vocabulary changes. Detecting such changes requires semantic analysis of the information in the text data to see whether or not the texts imply a same sense.

Plagiarism detection can also be divided into two main categories: external plagiarism detection, and intrinsic plagiarism detection. External plagiarism detection tries to extract plagiarism in a text by checking all given source documents. Intrinsic plagiarism detection analyzes the given suspicious document, and tries to discover parts of the input document which are not written by the same author. In this study we propose a new method to detect external plagiarism for Persian documents using deep learning approach [21].

3. RELATED WORK

In this section some plagiarism detection methods are reviewed. These methods categorized based on features that are used to determine the similarity between two documents which address different kind of plagiarism:

- Lexical methods: These methods consider text as a sequence of characters or terms. In this methods the assumption is that the more terms both documents have in common, the more similar they are. Methods that use features such as longest common subsequence, n-grams and fingerprint are considered as this kind of methods. These methods usually end up with a great outcome when the words are not changed by their synonyms [2, 7, 13, 14, 17, 21, 31, 38 and 40].
- Syntactical methods: Some methods use text's syntactical units for comparing the similarity between documents. This is a realization of the intuition that similar documents would have similar syntactical structure. This methods make use of characteristics such as POS tag to compare the similarity between different documents. [24,25]
- Semantic methods: These methods use semantic similarity for comparing documents. Methods that use synonyms, antonyms, hypernyms, and hyponyms are placed in this category [7, 39].

To the best of our knowledge, due to lack of Persian corpus (Persian tagged data) [16], there exist only few studies on Persian plagiarism detection. Mahdavi et al., [24] introduce Persian plagiarism detector based on bag of word model. Their approach has two steps: at first, most relevant source documents are retrieved by using cosine similarity, then, using the overlap coefficient and tri-gram model, plagiarism is identified. Mahmoodi et al., [25] use different combination of n-grams, Clough metric [9] and Jaccard similarity coefficient for automatic Persian plagiarism detection.

Most of conducted studies in Persian plagiarism detection are placed among lexical methods. As it is mentioned earlier, this kind of methods does not acts well when the words are changed and rewritten. Applying semantic similarity in Persian language has some limitations due to the constraints of the Persian WordNets.

Socher et al propose a deep method for paraphrase detection based on recursive autoencoder networks [37]. In this article a deep learning approach is introduced which uses semantic and lexical

features to detect plagiarism in Persian documents. To the best of our knowledge there is no reported study that uses deep learning for Persian plagiarism detection.

4. DEEP LEARNING FOR FEATURE EXTRACTION

Deep learning is a branch of machine learning which tries to find more abstract features using deep multiple layer graph. Each layer has linear or non-linear function to transform data into more abstract ones [3]. One of the reasons that the deep learning helps to improve NLP is the hierarchical nature of concepts. Concepts exist in natural world are generally hierarchical. For example a cat is a domestic animal which itself is a branch of animals. In most, not all, cases the word "cat" can be replaced by "dog" in any sentence with no change in resulting sentence. So abstract concepts in higher level are less sensitive to changes [4].

Recently, three factors contributed to the better performance of deep architecture: large datasets, faster computers and parallel processing in addition to the increasing number of machine learning methods for normalization and improvement of algorithms [12].

Due to the large amount of textual data and mentioned problems for natural language processing tasks, using automatic methods like deep learning seem mandatory. Advantages of using deep methods for NLP task are listed below:

- No hand crafted feature engineering is required
- Fewer number of features
- No labeled data is required

Multi-layer networks in deep learning, called deep belief network, can also lead to analogous set of features for all natural language processing tasks [10]. Using these representations reduces the number of features and the text can be described by far fewer features through combination functions.

4.1 Word Vector Representation

Most of language processing algorithms consider words as single symbols. This kind of representation suffers from sparsity since the length of vector corresponds to the size of word glossary. This vector has zero in all elements except one. This approach, called One-On, is unable to distinguish similarity between two synonym words. To address this challenge, an idea of representing a word by its neighbors was introduced by Firth [15].

In application of deep learning in natural language processing, each word is described by the surrounding context. The vector generated automatically by a deep neural networks and contain semantic and syntactic information about the word. Distributed word representation, generally known as word-embedding, is used to solve the aforementioned problems of high dimensionality and sparsity in language model. Here the similar words have the similar vectors [36].

Distributed representation learning introduced by Hinton for the first time [20] and developed in language modeling concept by Bengio [6]. Collobert [11] shows that distributed representation of words with almost no engineered features can be shared by several NLP tasks resulting the equal or more accuracy than the state of the art methods. Finally, authors in [29] indicate that this kind of presentation not only encompass a huge part of syntactic and semantic rules, but also the relationship between words can be modeled by vectors' offset. This offset can also presents the plurality, syntactic label (noun, verb, etc.), semantic feature (pet, animal, car, etc.) of a word.

This representation is used in all NLP tasks like Name-Entity-Recognition (NER), word-sense-disambiguation, parsing, and machine translation [10].

There are two approaches to learning word vector representation: 1) General matrix decomposition methods such as Latent Semantic Analysis (LSA) and 2) context-base methods such as skip-grams, continuous bag of words [28, 32].

Skip-grams and continuous bag of words, which are employed by this study, are two-layer neural networks that are trained for language modeling task. Skip-gram used one-on representation of words in a limited window size as an input and try to predict the middle word of the context. Another version of this network, continuous bag of words, is used to predict the context considering a middle word. The resulted vectors, which are the weights of the neural network, are the same for semantically similar words.

4.2 Text Document Vector Representation

There are so many algorithms which are used as the composition function for combining word vectors to generate a representation for text document.

Paragraph Vector is an unsupervised algorithm that learns representation for variable-length pieces of texts, such as sentences, paragraphs, and documents. The algorithm used the idea of word vector training and considered a matrix for each piece of text. This matrix also update during language modeling task. Paragraph vector outperform other methods such as bag-of-words models for many applications [23].

Socher [36] introduce Recursive Deep Learning methods which are variations and extensions of unsupervised and supervised recursive neural networks (RNNs). This method uses the idea of hierarchical structure of the text and encodes two word vectors into one vector by auto-encoder networks. Socher also presents many variation of these deep combination functions such as Recurrent Neural Network (RNN) and Matrix-Vector Recursive Neural Networks (MV-RNN).

There are also some simple mathematical methods which applied as a composition function generally used as benchmarks [30].

5. PROPOSED METHOD

In this study, in order to detect plagiarism, a sentence by sentence comparison is carried out in two phases. We first extract word vectors by word2vec algorithm [28], then remove Persian stop words while text pre-processing. After that, for each sentence an average of all word vectors is calculated as in equation 1.

$$S_i = \frac{\sum_{i=1}^{n} w_i}{n} \quad (1)$$

Where S is the vector representation for sentences and w_i is the word vector for i^{th} word of the sentences and n is the number of words in that sentence.

After feature extraction, in phase 1, each sentence in a suspicious document is compared with all the sentences in the source

documents. Cosine similarity is used as a comparison metric, which is described in equation 2.

$$Cosine\ Similairy = \frac{S1.S2}{\|S1\|\|S2\|} = \frac{\sum_{i=1}^{K} S1_i S2_i}{\sqrt{\sum_{i=1}^{K} {S1_i}^2}\sqrt{\sum_{i=1}^{K} {S2_i}^2}} \quad (2)$$

Where *S1* is the sentence vector of the sentence from suspicious documents and *S2* is the sentence vector of the sentence from source documents and *K* denoted the dimension of the vectors.

After this step which helps us to find the most nearest sentences in real time, in phase 2, lexical similarity of two sentences is evaluated by the Jaccard similarity measure. Jaccard similarity score is calculated as in equation 3.

$$Jaccard\ (S_1, S_2) = \frac{S_1 \cap S_2}{S_1 \cup S_2} \quad (3)$$

Where S_1 is the set of unique words in the first sentence and S_2 is the set of unique words in the second sentence.

Two sentences which pass Jaccard similarity threshold considered as plagiarism at final step. We used training corpus to fine-tune the thresholds. The workflow of our method is represented in figure 1.

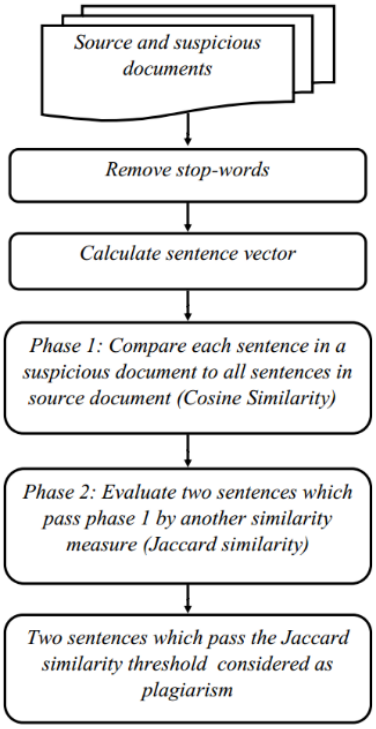

Figure 1: Steps of our plagiarism detection method

6. EXPERIMENTS

6.1 Dataset

We train our learning parameters on Persian PAN2016 dataset, since PAN2016 dataset has not been released yet, detailed information cannot be described. More detail in [1].

6.2 Parameter Definition

In this paper there are two parameters to be optimized. The task is to answer the following questions.

- What is the optimized threshold for the cosine similarity measure?
- What is the optimized threshold for the Jaccard similarity measure?

Two sentences are considered as plagiarism if they pass the cosine similarity threshold (α). The second threshold (β) filters the selected sentences to assure lexical similarity. These thresholds were fine-tuned by several trial on the training corpus. The results achieved when α=0.3 and β=0.2.

6.3 Evaluation Metrics

Evaluation measures on this text alignment task include: Precision, recall, and granularity, which are combined into the plagdet score [34].

$$Prec(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|\cup_{s\in S}(S \sqcap r)|}{|r|} \quad (4)$$

$$rec(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{|\cup_{r\in R}(S \sqcap r)|}{|s|} \quad (5)$$

$$\text{Where } S \sqcap r = \begin{cases} s \cap r \ if\ r\ detects\ s, \\ \emptyset \quad otherwise. \end{cases}$$

Where *S* is the set of plagiarism cases in the corpus and *R* is the set of detected plagiarism.
Granularity is defined to address overlapping or multiple detection for one plagiarism case and is defined as bellow.

$$gran(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|R_s| \quad (6)$$

All these measure combined into a single score, palgdet, as follows:

$$plagdet(S,R) = \frac{F_1}{\log_2(1 + gran(S,R))} \quad (7)$$

Where F_1 is the harmonic mean of precision and recall.

6.4 RESULTS

The results of applying this method to Persian PAN2016 corpus is presented in table 1, Rank 2, which is also reported in [1]. Persian plagiarism detection contest, PAN2016, was hosted on Tira [18, 33], a framework for shared tasks, and evaluated based on evaluation framework presented in [34].

7. CONCLUSION

In this paper, we used deep representation of words for plagiarism detection task. Sentence-by-sentence comparison is used to find text similarities. Advantages of this method among others are its simplicity and its fast sentence comparison. This methods has resulted in %90.6 plagdet, %85.8 recall, %95.9 precision on the PAN2016 provided data sets.

Why our method works? Since our comparison transformed from word-by-word or n-gram-by-n-gram representation of text to numerical one, the calculation of similarity execute in a much faster and more convenient way. Our method could easily and immediately address plagiarism with no obfuscation since the average of two same sentences word vectors are exactly the same. This methods also detect plagiarism with synthetic changes, include change of word's order, which have the same average vectors, as well. Vocabulary change, include adding or omitting words, which would be indistinguishable by string matching approach, could be identify by the proposed method. The reason is that the average vector is insensitive to few number of changes in a sentence vocabulary. On semantic changes, which is our main privilege in this task among others, plagiarism could easily be detected due to the similarity of synonym word vectors which make no or little changes on final sentence vector. Therefore, time consuming synonym word retrieval from lexicon has become inessential.

Table 1: Results of text alignment software submissions in PersianPlagDet-2016 (PAN16)

Rank	Team	Plagdet	Granularity	Precision	Recall
1	Fatemeh Mashhadi, Mehrnoush Shamsfard Shahid Beheshti University, NLP Research Lab	**0.922**	1.001	0.927	0.919
2	Hadi Veisi, Kayvan Bijari, Kiarash Zahirnia, Erfaneh Gharavi University of Tehran, Data & Signal processing Lab	**0.906**	1.000	0.959	0.858
3	Mozhgan Momtaz, Kayvan Bijari, Davood Heidarpour University of Tehran, COIN Lab	**0.871**	1.000	0.893	0.850
4	Mahdi Niknam, University of Qom	**0.830**	1.040	0.920	0.796
5	Faezeh Esteki, Faramarz Safi Esfahani Najafabad Branch, Islamic Azad University	**0.801**	1.000	0.933	0.701
6	Alireza Talebpour, Mohammad Shirzadi, Zahra Aminolroaya, Mohammad Adibi, Ahmad Mahmoudi-Aznaveh Shahid Beheshti University, Content lab /cyberspace research institute	**0.775**	1.228	0.964	0.836
7	Nava Ehsan University of Tehran	**0.727**	1.000	0.750	0.705
8	Lee Gillam, Anna Vartapetiance University of Surrey	**0.400**	1.528	0.755	0.414
9	Muharram Mansoorizadeh Bu-Ali Sina University	**0.390**	3.537	0.900	0.807

8. REFERENCES

[1] Asghari, H., Mohtaj, S., Fatemi, O., Faili, H., Rosso, P., and Potthast, M., 2016. Algorithms and Corpora for Persian Plagiarism Detection: Overview of PAN at FIRE 2016. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation,* Kolkata, India, December 7-10, 2016, CEUR Workshop Proceedings, CEUR-WS.org.

[2] Barrón-Cedeño, A., Vila, M., Martí, M.A., and Rosso, P., 2013. Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. Computational Linguistics 39, 4, 917-947.

[3] Bengio, Y., 2009. Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2(1), 1-127.

[4] Bengio, Y., Courville, A., and Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35, 8, 1798-1828.

[5] Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C., 2003. A neural probabilistic language model. Journal of machine learning research, 3(Feb), 1137-1155.

[6] Bengio, Y., Schwenk H., Senécal J.-S., Morin F., and Gauvain J.-L., 2006. Neural probabilistic language models, in Innovations in Machine Learning, pp. 137-186.

[7] Chen, C.-Y., Yeh, J.-Y., and Ke, H.-R., 2010. Plagiarism detection using ROUGE and WordNet. arXiv preprint arXiv:1003.4065.

[8] Chong, M.Y.M., 2013. A study on plagiarism detection and plagiarism direction identification using natural language processing techniques.

[9] Clough, P., 2003. Old and new challenges in automatic plagiarism detection. In National Plagiarism Advisory Service, 2003; http://ir. shef. ac. uk/cloughie/index. html.

[10] Collobert, R. and Weston, J., 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning ACM, 160-167.

[11] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P., 2011. Natural language

processing (almost) from scratch. Journal of machine learning research 12, Aug, 2493-2537.

[12] Dahl, G., Mohamed, A.-R., and Hinton, G.E., 2010. Phone recognition with the mean-covariance restricted Boltzmann machine. In Advances in neural information processing systems, 469-477.

[13] Elhadi, M. and Al-Tobi, A., 2008. Use of text syntactical structures in detection of document duplicates. In Digital Information Management, ICDIM 2008. Third International Conference on IEEE, 520-525.

[14] Elhadi, M. and Al-Tobi, A., 2009. Duplicate detection in documents and webpages using improved longest common subsequence and documents syntactical structures. In Computer Sciences and Convergence Information Technology, ICCIT'09. Fourth International Conference on IEEE, 679-684.

[15] Firth, J.R., 1957. A synopsis of linguistic theory, in Studies in Linguistic Analysis, Philological Society, Oxford.

[16] Franco-Salvador, M., Bensalem, I., Flores, E., Gupta, P., and Rosso, P., 2015. PAN 2015 Shared Task on Plagiarism Detection: Evaluation of Corpora for Text Alignment. In Volume 1391 of CEUR workshop proceedings CLEF and CEUR-WS. org.

[17] Glinos, D.S., 2014. A Hybrid Architecture for Plagiarism Detection. In CLEF (Working Notes), 958-965.

[18] Gollub, T., Stein, B., and Burrows, S., 2012. Ousting ivory tower research: towards a web framework for providing experiments as a service. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval ACM, 1125-1126.

[19] Hinton, G. E., Osindero, S., & Teh, Y. W., 2006. A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.

[20] Hinton, G.E., 1986. Learning distributed representations of concepts. In Proceedings of the eighth annual conference of the cognitive science society Amherst, MA, 12.

[21] Hoad, T.C. and Zobel, J., 2003. Methods for identifying versioned and plagiarized documents. Journal of the American society for information science and technology 54, 3, 203-215.

[22] Lathrop, A. and Foss, K., 2000. Student Cheating and Plagiarism in the Internet Era. A Wake-Up Call. ERIC.

[23] Le, Q.V. and Mikolov, T., 2014. Distributed Representations of Sentences and Documents. In ICML, 1188-1196.

[24] Mahdavi, P., Siadati, Z., and Yaghmaee, F., 2014. Automatic external Persian plagiarism detection using vector space model. In Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on IEEE, 697-702.

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

[26] Manning, C. D., & Schütze, H., 1999. Foundations of statistical natural language processing (Vol. 999). Cambridge: MIT press.

[27] Maurer, H. and Zaka, B., 2007. Plagiarism–a problem and how to fight it. Proceeding of Ed-Media 2007, 4451-4458.

[28] Mikolov, T., Chen, K., Corrado, G., and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[29] Mikolov, T., Yih, W.-T., and Zweig, G., 2013. Linguistic Regularities in Continuous Space Word Representations. In HLT-NAACL, 746-751.

[30] Mitchell, J., & Lapata, M., 2010. Composition in distributional models of semantics. Cognitive science, 34(8), 1388-1429.

[31] Nahnsen, T., Uzuner, O., and Katz, B., 2005. Lexical chains and sliding locality windows in content-based text similarity detection.

[32] Pennington, J., Socher, R., and Manning, C.D., 2014. Glove: Global Vectors for Word Representation. In EMNLP, 1532-1543.

[33] Potthast, M., Gollub, T., Rangel, F., Rosso, P., STAMATATOS, E., and STEIN, B., 2014. Improving the Reproducibility of PAN's Shared Tasks. In International Conference of the Cross-Language Evaluation Forum for European Languages Springer, 268-299.

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

[35] Sanchez-Perez, M.A., Gelbukh, A., and Sidorov, G. Dynamically Adjustable Approach through Obfuscation Type Recognition.

[36] Socher, R., 2014. Recursive Deep Learning for Natural Language Processing and Computer Vision PhD thesis, Stanford University.

[37] Socher, R., Huang, E.H., Pennin, J., Manning, C.D., and Ng, A.Y., 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In Advances in Neural Information Processing Systems, 801-809.

[38] Suchomel, S., Kasprzak, J., and Brandejs, M., 2012. Three Way Search Engine Queries with Multi-feature Document Comparison for Plagiarism Detection. In CLEF (Online Working Notes/Labs/Workshop) Citeseer, 1-8.

[39] Torres, S. and Gelbukh, A., 2009. Comparing similarity measures for original WSD lesk algorithm. Research in Computing Science 43, 155-166.

[40] Zini, M., Fabbri, M., Moneglia, M., and Panunzi, A., 2006. Plagiarism detection through multilevel text comparison. In 2006 Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS'06) IEEE, 181-185.