Engineering Universe for scientific Research and management
ISSN (Online): 2319-3069
Vol. XIII Issue VIII
August 2021

A REVIEW ON MACHINE LEARNING TECHNIQUES TO DETECT PLAGIARISM

Vivek Kumar[1], Chinmay Bhatt[2], Bharti Chourasia[3]
CSE, SRK University, Bhopal, India[1,2,3]
kvivek371@gmail.com[1], chinmay20june@gmail.com[2], varsha_namdeo@yahoo.com[3]

ABSTRACT: The ease of access to the various resources on the web enabled the democratization of access to information but at the same time allowed the appearance of enormous plagiarism problems. Many techniques of plagiarism were identified in the literature, but the plagiarism of idea steels the foremost troublesome to detect, because it uses different text manipulation at the same time. Indeed, a few strategies have been proposed to perform the semantic plagiarism detection, but they are still numerous challenges to overcome. Unlike the existing states of the art, the purpose of this study is to give an overview of different proposition for plagiarism detection based on the Machine learning algorithms. The main goal of these approaches is to provide high quality of worlds or sentences vector representation. In this paper, we propose a comparative study based on a set of criterions like: Vector representation method, Level Treatment, Similarity Method and Dataset. One result of this study is that most of researches are based on world granularity and use the word2vec method for word vector representation, which sometimes is not suitable to keep the meaning of the whole sentences. Each technique has strengths and weaknesses; however, none is quite mature for semantic plagiarism detection.

KEYWORDS: Plagiarism, Extrinsic plagiarism detection, intrinsic plagiarism detection, Machine Learning, Neural network

1. INTRODUCTION

World Wide Web provides access to data present in the documents, databases, and other sources of information using internet service. The availability of knowledge and information in the digital form leads to "Plagiarism" by "Plagiarist". Plagiarism [1] is defined as the act of stealing and copying the intellectual work, ideas, results, or language of another person without giving credit to the original author and presents it as one's own original work. Plagiarism is not only a major concern in the field of academics but other domains as well such as politics, journalism, music industry, art, medical and scientific research are few to mention here. For this reason, in the academic and research world, various institutions like Elsevier, Springer, and many incorporate anti-plagiarism tools. The function of the plagiarism detection system is to capture the plagiarized content.

Plagiarism detection can be applied to two types of documents, namely Natural Language and Programming Language. Plagiarism detection for Natural language known as Text plagiarism detection and for Programming Language is known as Software or source code plagiarism detection [2].

Plagiarism detection on the basis of usage of original resources and reference collection can be further categorized as Extrinsic and Intrinsic Plagiarism detection. Extrinsic plagiarism detection technique uses reference collection for pair-wise comparison between suspicious document and source document based on features such as semantic features, syntactic features, and so on [3]. Intrinsic plagiarism detection systems do not take into account the reference of sources for plagiarism detection. Intrinsic plagiarism detection [4] techniques catch plagiarism cases when no reference collection is available for comparison

between suspicious documents and source documents. It uses features such as writing style of the author, vocabulary richness including other stylometric features such as deviation in writing style [5], most common words [6] used by the author, and their frequency and vocabulary richness [7].

On the basis of similarity of a text document in respect of language, plagiarism detection can be either monolingual or multilingual. Monolingual plagiarism detection deals with the detection of plagiarism cases where the source and suspicious documents are in the same language such as English–English, whereas Multilingual or cross-language plagiarism detection is used when the original and corresponding plagiarized document uses different languages such as English-Chinese.

2. RELATED WORK

Our study focuses on the detection of semantic plagiarism more precisely the identification of the plagiarism of ideas between two given texts, as illustrated below we dug on methods that detect this type of plagiarism:

In [8] proposed a plagiarism detection system, which rely on use sentences comparison in two phases. They first extract word vectors by word2vec algorithm, and then remove Persian stop words while text pre-processing. After that, for each sentence an average of all word vectors is calculated. 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. After this step which helps to find the nearest sentences in real time, in phase 2, lexical similarity of two sentences is evaluated by the Jaccard similarity measure. Two sentences which pass Jaccard similarity threshold considered as plagiarism at final step. In [9] proposed the use word2vec model in order to compute vector of features for every word. They choose documents from the corpus itself, however the documents used for testing was processed and the pre-processing that was made is stop words removal. The similarity between vectors was computed by using cosine similarity. [10] The aim of this approach is evaluating the validity of using the distributed representation to define the word similarity. They introduce three methods based on the following three document similarities: for two documents: The length of the longest common subsequence (LCS) divided by the length of the shorter document, the local maximal value of the length of LCS, and the local maximal value of the weighted length of LCS. The distributed representation was obtained from no particular data by word2vec.

Another approach uses the principle of Deep Structured Semantic Model (DSSM) proposed by [11]. DSSM is a Machine learning-based technique that is proposed for semantic understanding of textual data. It maps short textual strings, such as sentences, to feature vectors in a low-dimensional semantic space. Then the vector representations are utilized for document retrieval by comparing the similarity between documents and queries. After obtaining the semantic feature vectors for each paired snippets of text, cosine similarity is utilized to measure the semantic similarity between the pair. Similarly, with the previous methods, in [12] Machine learning documents or texts can be represented as vectors by the using document to vector technique (doc2vec). And the detection of plagiarism will be done by a simple comparison between all sentences of each two documents analyzed.

The approach proposed in [13] is based on converting a paragraph to vectors and it's inspired by the methods for learning the word vectors. The inspiration is that the word vectors are asked to contribute to a prediction task about the next word in the sentence. So, despite the fact that the word vectors are initialized randomly, they can eventually capture semantics as an indirect result of the prediction task. It will use this idea in their paragraph vectors in a similar manner. The paragraph vectors are also asked to contribute to the prediction task of the next word given many contexts sampled from the paragraph.

These approaches [12-13] are used to perform similarity detection between the document vectors but also use the cosine to compare the vectors. In paper [14] they represent each word w by a vector. It constructs these word vectors using GloVe. This approach uses the recursive neural networks

algorithm to have a vector representation of a sentence and use the cosine for calculate the similarity. In [15] two input sentences are processed in parallel by identical neural networks, outputting sentence representations. The sentence representations are compared by the structured similarity measurement layer. The similarity features are then passed to a fully-connected layer for computing the similarity score. Cosine distance measures the distance of two vectors according to the angle between them. The use of cosine to detect similarity between sentences remains a solution that carries many risks. InferSent [16] is an NLP technique for universal sentence representation developed by Facebook that uses supervised training to produce high transferable representations. They used a Bi-directional LSTM with attention that consistently surpassed many unsupervised training methods such as the Skip Thought vectors. They also provide a Pytorch implementation that they used to generate sentence embedding. So, this approach needs to define a similarity measure to compare two vectors, and for that goal, it'll be the cosine similarity.

The authors in [17] used word embedding, vector representations of terms, computed from unlabelled data, that represent terms in a semantic space in which proximity of vectors can be interpreted as semantic similarity. They propose to go from word-level to text-level semantics by combining insights from methods based on external sources of semantic knowledge with word embedding. They derive multiple types of meta-features from the comparison of the word vectors for short text pairs, and from the vector means of their respective word embedding. The features representing labelled short text pairs are used to train a supervised learning algorithm. In [18] present the Word Mover's Distance (WMD), a novel distance function between text documents. This work is based on recent results in word embedding that learn semantically meaningful representations for words from local co-occurrences in sentences. The WMD distance measures the dissimilarity between two text documents as the minimum amount of distance that the embedded words of one document need to "travel" to reach the embedded words of another document. This article [19] proposed an innovative word embedding-based system devoted to calculating the semantic similarity in Arabic sentences. The main idea is to exploit vectors as word representations in a multidimensional space in order to capture the semantic and syntactic properties of words. IDF weighting and Part-of-Speech tagging are applied on the examined sentences to support the identification of words that are highly descriptive in each sentence.

In paper [20] they address the issue of finding an effective vector representation for a very short text fragment. By effective they mean that the representation should grasp most of the semantic information in that fragment. For this, they use semantic word embedding to represent individual words, and we learn how to weigh every word in the text through the use of tf-idf (term frequency-inverse document frequency) information to arrive at an overall representation of the fragment comparing two tf-idf vectors is done through a standard cosine similarity. [21] This paper investigates the effectiveness of several such naive techniques, as well as traditional tf-idf similarity, for fragments of different lengths. This main contribution is a first step towards a hybrid method that combines the strength of dense distributed representations-as opposed to sparse term matching-with the strength of tf-idf based methods to automatically reduce the impact of less informative terms. This approach outperforms the existing techniques in a toy experimental set-up, leading to the conclusion that the combination of word embedding and tf-idf information might lead to a better model for semantic content within very short text fragments. Between two such representations they then calculate the cosine similarity.

In the architecture proposed in [22], word embedding is first trained on API documents, tutorials, and reference documents, and then aggregated in order to estimate semantic similarities between documents where the similarity between vectors is usually defined as cosine similarity. In paper [23], they propose to combine explicit semantic analysis (ESA) representations and word2vec representations as a way to generate denser representations and, consequently, a better similarity measure between short texts. In [24] they proposed a semantic similarity approach for paraphrase identification in Arabic texts by combining different techniques of

Natural Language Processing NLP such as: Term Frequency Inverse Document Frequency TF-IDF technique. The goal is to represent a word vector using word2vec. And also, to generate a sentence vector representation and after applying a similarity measurement operation based on different metrics of comparison, such as: Cosine Similarity and Euclidean Distance. This approach was evaluated on the Open Source Arabic Corpus OSAC and obtained a promising rate.

[25] This paper proposes a novel deep neural network-based approach that relies on coarse-grained sentence modeling using a convolutional neural network and a long short-term memory model, combined with a specific fine-grained word-level similarity matching model. In this component, they represent every sentence using their joint CNN and LSTM architecture. The CNN is able to learn the local features from words to phrases from the text, while the LSTM learns the long-term dependencies of the text. More specifically, they firstly take the word embedding as input to their CNN model, in which various types of convolutions and pooling techniques are applied to capture the maximum information from the text. Next, the encoded features are used as input to the LSTM network. Finally, the long-term dependencies learned by the LSTM become the semantic sentence representation.

[26] This approach proposes to explicitly model pairwise word interactions and present a novel similarity focus mechanism to identify important correspondences for better similarity measurement. They used GloVe word embeddings for vector representation of word and their model contains four major components: 1. Bidirectional Long Short-Term Memory Net-works (Bi-LSTMs) are used for context modeling of input sentences. 2. A novel pair wise word interaction modeling technique encourages direct comparisons between word contexts across sentences. Cosine distance (cos) measures the distance of two vectors by the angle between them, while L2Euclidean distance (L2Euclid) and dotproduct distance (DotProduct) measure magnitude differences. We use three similarity functions for richermeasurement. 3. A novel similarity focus layer helps the model identify important pair wise word interactions across sentences.4. A layer deep convolutional neural network (ConvNet) converts the similarity measurement problem into a pattern recognition problem for final classification.

The model of [27] is applied to assess semantic similarity between sentences. For these applications, they provide word-embedding vectors word2vec to the LSTMs, which use a fixed size vector to encode the underlying meaning expressed in a sentence (irrespective of the particular wording/syntax). By restricting subsequent operations to rely on a simple Manhattan metric, they compel the sentence representations learned by their model to form a highly structured space whose geometry reflects complex semantic relationships. [28] This paper proposes a model for com-paring sentences that uses a multiplicity of perspectives. We first model each sentence using a convolutional neural network that extracts features at multiple levels of granularity and uses multiple types of pooling. We then compare our sentence representations at several granularities using multiple similarity metrics (cos, LEuclid). We apply our model to three tasks, including the Microsoft Research paraphrase identification task and two SemEval semantic textual similarity tasks.

In this paper [29], they present convolutional neural network architecture for re-ranking pairs of short texts, where they learn the optimal representation of text pairs and a similarity function to relate them in a supervised way from the available training data. Their network takes only words in the input, thus requiring minimal preprocessing. In particular, they consider the task of re-ranking short text pairs where elements of the pair are sentences. They test our Machine learning system on two popular retrieval tasks from TREC: Question Answering and Microblog Retrieval. [30] This system combines convolution and recurrent neural networks to measure the semantic similarity of sentences. It uses a convolution network to take account of the local context of words and an LSTM to consider the global context of sentences. This combination of networks helps to preserve the relevant information of sentences and improves the calculation of the similarity between sentences. According to this state of the art we have been able to detect the strengths and weaknesses of each approach that helped us to build our approach.

3. TYPES OF PLAGIARISM

Various researchers define and categorize the plagiarism types according to their studies and analysis. Plagiarism types can be Copy and paste, Disguised, Shake and paste, Structural, Plagiarism by translation, Metaphor, Patchwork paraphrasing, and Idea plagiarism [8]. Besides these, Plagiarism types are given below:

3.1 Intentional Plagiarism

Intentional or deliberate plagiarism takes place when plagiarists copy the content, steal the idea or work done by others deliberately, and present it as their own ideas. The reason for practicing this plagiarism could be laziness among plagiarists, lack of confidence, stress, or anxiety due to competition and a lack of knowledge about the subject.

3.2 Unintentional Plagiarism

Unintentional or accidental plagiarism takes place when proper citations and references are not given. The reason for practicing this kind of plagiarism could be a lack of knowledge for citing the original sources or unintentionally represent the idea with similar words.

3.3 Self Plagiarism

Self-plagiarism [9] is an act of reusing the own previously published material without giving citations that it has used earlier and presented it as new. The reason behind practicing self-plagiarism could be to save time and efforts for publishing more work.

3.4 Mosaic Plagiarism

Mosaic plagiarism [10] is an act of making use of phrases and use of synonyms in place of original words from source but the idea remains the same without giving credit to the original author.

4. CONCLUSION

In this paper, we have mentioned many different methods used in detection of plagiarism of ideas that stand for the principal of Machine Learning, and by this brilliant study we could construct our critical base of the previous weaknesses which we have seen during our study. This helped us to get a general idea about the different methods of Machine learning used for plagiarism detection or especially semantic plagiarism detection. In addition to this, this study has given us the paths to follow for the construction of our approach by benefiting from the strengths of each method and bypassing the weak points of each method. Concerning the future work consists of construct and putting into practice our approach and comparing it with the other methods used at the level of the phase related work.

REFERENCES

[1.] Halak B, El-Hajjar M (2016) Plagiarism detection and prevention techniques in engineering education. In: 2016 11th European workshop on microelectronics education (EWME). IEEE, pp 1–3

[2.] Alzahrani SM, Salim N, Abraham A (2011) Understanding plagiarism linguistic patterns,textual features, and detection methods. IEEE Trans Syst Man Cybern Part C (Appl Rev) 42(2):133–149

[3.] Gupta D (2016) Study on extrinsic text plagiarism detection techniques and tools. J Eng Sci Technol Rev 9(5):8–22

[4.] Zu Eissen SM, Stein B (2006) Intrinsic plagiarism detection. In: European conference on information retrieval. Springer, Berlin, pp 565–569

[5.] Oberreuter G, VeláSquez JD (2013) Text mining applied to plagiarism detection: the use of words for detecting deviations in the writing style. Expert Syst Appl 40(9):3756–3763

[6.] AlSallal M, Iqbal R, Palade V, Amin S, Chang V (2017) An integrated approach for intrinsic plagiarism detection. Future Gen Comput Syst 700–712

[7.] Zu Eissen SM, Stein B, KuligM(2007) Plagiarism detection without reference collections. In: Advances in data analysis. Springer, Berlin, pp 359–366

[8] Erfaneh Gharavi, Kayvan Bijari and Kiarash Zahirnia. A Deep Learning Approach to Persian Plagiarism Detection. DOI: 10.1109/ICTCS.2017.42 Conference: Conference: The International Conference on new Trends in Computing Sciences

(ICTCS2017). University of Tehran Faculty of new Science and Technology Data & Signal processing Lab 2017.

[9] Dima Suleiman, Arafat Awajan and Arafat Awajan. Deep Learning Based Technique for Plagiarism Detection in Arabic Texts. 2017 International Conference on New Trends in Computing Sciences. Computer Science Department Princess Sumaya University for Technology 2017.

[10] Kensuke Baba, Tetsuya Nakatoh and Toshiro Minami. Plagiarism detection using document similarity based on distributed representation. 8th International Conference on Advances in Information Technology, IAIT2016, 19-22 December 2016, Macau, China. Fujitsu Laboratories, Kawasaki, Japan Kyushu University, Fukuoka, Japan.

[11] Naveed Afzal, Yanshan Wang and Hongfang Liu. MayoNLP at SemEval-2016 Task 1: Semantic Textual Similarity based on Lexical Semantic Net and Deep Learning Semantic Model. Proceedings of SemEval-2016, pages 674–679, San Diego, California, June 16-17, 2016. 2016 Association for Computational Linguistics. Department of Health Sciences Research Mayo Clinic, Rochester, MN.

[12] Tedo Vrbanec and Ana Mestrovic. The Struggle with Academic Plagiarism: Approaches based on Semantic Similarity. MIPRO 2017, May 22- 26, 2017, Opatija, Croatia. Faculty of Teacher Education, University of Zagreb, Croatia Department of [nformatics, University of Rijeka, Croatia.

[13] Quoc Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043.

[14] Adrian Sanborn and Jacek Skryzalin. Deep Learning for Semantic Similarity. MIPRO 2017, May 22- 26, 2017, Opatija, Croatia. Department of Computer Science Stanford University.

[15] Hua He, Kevin Gimpel,and Jimmy Lin. Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks. DOI: 10.18653/v1/D15-1181 Conference: Conference: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Department of Computer Science, University of Maryland, College Park, Toyota Technological Institute at Chicago and David R. Cheriton School of Computer Science, University of Waterloo.

[16] Christian S. Perone. Privacy-preserving sentence semantic similarity using InferSent embeddings and secure two-party computation.

[17] Tom Kenter and Maarten de Rijke. Short Text Similarity with Word Embeddings. CIKM '15 Proceedings of the 24th ACM International on Conference on Information and Knowledge Management Pages 1411-1420. University of Amsterdam, Amsterdam, The Netherlands.

[18] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin and Kilian Q. Weinberger. From Word Embeddings To Document Distances. Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Washington University in St. Louis, 1 Brookings Dr., St. Louis, MO 63130.

[19] El Moatez Billah Nagoudi and Didier Schwab. Semantic Similarity of Arabic Sentences with Word Embeddings. Proceedings of The Third Arabic Natural Language Processing Workshop (WANLP), pages 18–24, Valencia, Spain, April 3, 2017.©, 2017 Association for Computational Linguistic. LIM-Laboratoire d'Informatique et de Mathématiques, Université Amar Telidji de Laghouat, Algérie. LIG-GETALP Univ. Grenoble Alpes France.

[20] Cedric De Boom, Steven Van Canneyt, Thomas Demeester and Bart Dhoedt. Representation learning for very short texts using weighted word embedding aggregation. Journal Pattern Recognition Letters archive Volume 80 Issue C, September 2016 Pages 150-156. Department of Information Technology, Technologiepark 15, 9052 Zwijnaarde, Belgium.

[21] Cedric De Boom, Steven Van Canneyt, Steven Bohez, Thomas Demeester and Bart Dhoedt. Learning Semantic Similarity for Very Short Texts. 2015 IEEE International Conference on Data Mining Workshop (ICDMW). Ghent University – iMinds Gaston Crommenlaan 8-201, 9050 Ghent, Belgium.

[22] Xin Ye, Hui Shen, Xiao Ma, Razvan Bunescu, and Chang Liu. From Word Embeddings To Document Similarities for Improved Information Retrieval in Software Engineering. CSE ’16, May 14-22, 2016, Austin, TX, USA. School of Electrical Engineering and Computer Science, Ohio University Athens, Ohio 45701, USA.

[23] Yangqiu Song and Dan Roth. Unsupervised Sparse Vector Densification for Short Text Similarity. DOI: 10.3115/v1/N15-1138 Conference: Conference: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Department of Computer Science University of Illinois at Urbana-Champaign Urbana, IL 61801, USA.

[24] Adnen Mahmoud and Mounir Zrigui. Semantic Similarity Analysis for Paraphrase Identification in Arabic Texts. Conference: Conference: The 31st Pacific Asia Conference on Language, Information and Computation PACLIC 31 (2017), At University of the Philippines Cebu, Cebu, Philippines. LATICE Laboratory Research Department of Computer Science University of Monastir, Tunisia.

[25 Basant Agarwala, Heri Ramampiaroa, Helge Langsetha, Massimiliano Ruocco. A Deep Network Model for Paraphrase Detection in Short Text Messages. arXiv:1712.02820v1 [cs.IR] 7 Dec 2017. Dept. of Computer Science, Norwegian University of Science and Technology, Norway Swami Keshvanand Institute of Technology, India Telenor Research, Trondheim, Norway.

[26] Hua He and Jimmy Lin. Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement. Proceedings of NAACL-HLT 2016, pages 937–948, San Diego, California, June 12-17, 2016.c©2016 Association for Computational Linguistics. Department of Computer Science, University of Maryland, College Park David R. Cheriton School of Computer Science, University of Waterloo.

[27] Jonas Mueller and Aditya Thyagarajan. Siamese Recurrent Architectures for Learning Sentence Similarity. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16). Computer Science& Artificial Intelligence Laboratory Massachusetts Institute of Technology. Department of Computer Scienceand EngineeringM. S. Ramaiah Institute of Technology.

[28] Hua He, Kevin Gimpel, and Jimmy Lin. Multi-Perspective Sentence Similarity Modelingwith Convolutional Neural Networks. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1576–1586, Lisbon, Portugal, 17-21 September 2015.c©2015 Association for Computational Linguistics. Department of Computer Science, University of Maryland, College Park2Toyota Technological Institute at Chicago. David R. Cheriton School of Computer Science, University of Waterloo.

[29] Aliaksei Severyn, Alessandro Moschitti. Learning to Rank Short Text Pairs with Convolutional DeepNeural Networks. Google Inc. Qatar Computing Research Institute.

[30] Elvys Linhares Pontes, Stéphane Huet, Andréa Carneiro Linhares, Juan-Manuel Torres-Moreno. Predicting the Semantic Textual Similarity with Siamese CNN and LSTM. LIA, Université d’Avignon et des Pays de Vaucluse, Avignon, 84000 France Universidade Federal do Ceará, Sobral, Ceará Brazil École Polytechnique de Montréal, Montréal, Canada.