A Multi-Level Plagiarism Detection System Based on Deep Learning Algorithms

El Mostafa HAMBI and Faouzia Benabbou

University of Hassan 2, Faculty of Sciences Ben M'sik, Casablanca, Morocco

Summary
With the high rate of online scientific publications and the accessibility to retrieval of information, has proved enormous problem of plagiarism. The techniques for detecting plagiarism are becoming increasingly advanced, whereas plagiarism of ideas still one of the greatest challenges. In this regard, some methods have been proposed in order to minimize the act of plagiarism of idea. In this paper, we propose a system of plagiarism detection of ideas based on Deep Learning Algorithms. The proposed approach deals with some problems encountered in detecting the plagiarism of ideas such as: loss of meaning or the difficulty of detection of semantic similarity between documents. Thus, our system consists of using in a first place doc2vec to have a vector representation of each sentence of a document and then we use the siamese LSTM to make learn our system that pair of documents is similar and finally we use the CNN algorithm to classify the different types of plagiarism.

Key words:
Plagiarism; Deep Learning; Preprocessing; Doc2vec; neural network; Long short-term memory (LSTM); Convolutional neural network (Cnn); Siamese neural network.

1. Introduction

The development of information technology (IT) and especially the Internet has considerably increased the availability of information and leads consequently to the rising of plagiarism especially in scientific research domain. Acquiring without right, ideas of original works is considered a case of plagiarism and it is widespread problem in academic institutions. The Plagiarism is considered both illegal and immoral. Plagiarism detection is one of the research topics in Natural Language Processing (NLP). It aims to detect the reusing, reproducing and changing the text from one form to another one. The fraudsters perform different types of plagiarism as described below [1]:

- Copy-paste, textually (word by word) in which the content of the text is copied from one or more sources. Copied content could be slightly modified.
- Paraphrasing, to change the grammar, to use the synonyms of words, to reorganize the sentences of the original work and finally to delete some parts of the text.
- The use of false references, the addition of references which are false or that do not even exist.
- Plagiarism with translation, the contents are translated and used without reference to the original work.
- Plagiarism of ideas is the most difficult plagiarism to detect because it is more evolved than the previous types; also it is not simple manipulations made on the text, but a more advanced form. This type of plagiarism consists in using the concepts and ideas of others with a reformulation of sentences

As part of NLP research topic, the plagiarism detection methods are based on NLP techniques to process and analyze the structure of documents. Many solutions have been proposed for plagiarism detection, and most are based on the principle of concept extraction using a corpus like WordNet to have a semantic representation. However, the use of this approach poses the problem of the appropriate concept choice which semantically represents a word. Also, and especially in such methods, the similarity between the synonym words is not taken into account, the problem of ambiguity may arise, and the meaning of the processed sentences may be lost. Here some techniques of plagiarism detection [2]:

- Lexical methods: These methods consider text as a sequence of characters or terms. The documents which have the common terms are similar [3].
- Syntactical methods: Some methods use text's syntactical units for comparing the similarity between documents and this is a realization of the intuition, that similar documents would have similar syntactical structure [4].
- Semantic methods: These methods use a semantic similarity for comparing documents, methods that use synonyms, antonyms, hypernyms, and hyponyms are placed in this category [5], [2].

In this paper we concentrate our attention on the semantic plagiarism which is the most difficult to detect since it usually includes many techniques for the recovery of the text in another form. That is why automatic plagiarism detection methods have been developed to serve as a countermeasure against fraudsters.

Manuscript received October 5, 2019
Manuscript revised October 20, 2019

This is why we are interested in approaches that allow the semantic analysis of documents as the representation of a text in a vector space or what we say context representation. This method is very important for many natural language processing (NLP) applications such as text classification, information retrieval, automatic summarization, and plagiarism. As example of such method, the word2vec proposed by Mikolov and al. [6] is very popular and has attracted great attention over the last two years. It has been shown that vector representations of words learned by word2vec models have a semantic meaning and are useful in various tasks of NLP. The doc2vec technique is inspired by word2vec to have a vector representation of a sentence. Its importance lies on the fact that it makes easier to compare words and sentences, to find relations between two texts as well as minimizing the need of lexicons use [7].

Deep learning algorithms are an important component of computational intelligence which has the core domain machine learning research in it. In-depth learning provides models composed of multiple layers of processing capable of learning representations of data with multiple levels of abstraction. These methods yielded very encouraging results in speech recognition, image recognition, object detection, and many other areas such as NLP [8] [9]. Many deep learning architectures were used for the NLP like a simple Neuron Networks applied to have for example the word embeddings, or more complex algorithms like the Siamese LSTM which is used to detect the similarity of objects, Recurrent Neural Network (RNN), and the Convolutional Neural Network (CNN) which gage best score in classification.

In this paper we propose a Multi-Level Plagiarism Detection System based on the LSTM and CNN algorithms. Our tests on PAN Dataset show that our system is able to detect different types of plagiarism and specially the semantic one.

The remainder of this paper is organized as following: The second section is about defining related work. Additionally, the third section is devoted to illustrate deep learning algorithms used in our study. Concerning the fourth Sections, we define our approach and an overall illustration of this approach; and the last section contains a validation of our approach.

2. Litterature Review

Our work focuses on what we call a semantic detection part or more precisely detection of plagiarism of ideas. The semantic approach is aimed to attain the highly performance in terms of detection and should address the issues of polysemy and synonymy that are not handled by the lexical (straight forward term matching) approach. Below we will review some approaches that are based on deep learning techniques to detect semantic plagiarism. A comparative study was made in [29] based on criteria like: Level treatment (word/sentence), similarity method and performance provided.

Some plagiarism detection systems use a sentence-by-sentence comparison [8] [10]. This method has several problems, for example: at the level of the vector representation of a sentence, it calculates the average of the vectors of the sentence, and this can cause a problem of the loss of the meaning of the sentence.

The approach [9], proposed the use of word2vec model in order to compute vector of features for every word. The detection of plagiarism is done via the comparison between the vectors that represent the words. However, the problem with this method is that two documents that share the same vectors could be non-plagiarized.

Again, the approach in [11] aims to evaluate the validity of using the distributed representation to define the word similarity and the Longest Common Subsequence problem seeks a longest subsequence of every member of a given set of vectors. Also, the detection of the similarity always poses problems because; the comparison is carried out between words and thus the semantic aspect is neglected.

In the approach [12], they use the principle of Deep Structured Semantic Model (DSSM) proposed by Huang et al., (2013) [13]. It maps short textual strings, such as sentences that can increase the possibility of losing the meaning of the analyzed document.

In the two references [14] and [15], the authors are inspired by the wor2vec to construct a vector representation of a paragraph. For these two approaches, comparing two documents by these words can produce several problems since semantically speaking two documents that share the same words may be not similar.

The approach [16] uses the recursive neural networks algorithm to have a vector representation of a sentence but according to the study that is done by [14], [15] we found that the use of doc2vec gives us trampling results.

Infer Sent [17] is an NLP technique for universal sentence representation developed by Facebook that uses supervised training to produce highly transferable representations. So, this approach used the cosine similarity to compare two vectors. The use of cosine to detect similarity between sentences remains a solution that carries many risks, according to the study, which is done by [14], [15]. They found that the use of doc2vec gives trampling results.

The approach [18] uses the labeled feature representation for short text pairs, but when we use long text, the system will be very slow. The approach [19] presents 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 is in need to "travel" to reach the embedded words of another document. The approach [20] proposes an innovative word embedding-based system devoted to calculate the semantic similarity in Arabic sentences; the cosine similarity is applied to compute a similarity score between sentences may create numerous problems.

In the paper [21] the authors addressed the issue of finding an effective vector representation for a very short text fragment. By the word 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 they learn how to weigh every word in the text with 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.

The approach [22] investigates the effectiveness of several such naive techniques, as well as traditional tf-idf similarity, for fragments of different lengths. They calculate the cosine similarity for detecting plagiarism. In the proposed approach [23], the 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 the paper [24], they proposed the combination between the 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 this paper [25], they proposed a semantic similarity approach for the paraphrasing of the identification in Arabic texts by combining different techniques of Natural Language Processing NLP. They use the calculation of cosine for detecting similarity which can create many of problems.

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 proposition. For instance, we find the most of the mentioned approaches have utilized cosine measure to detect documents similarity, and this measure has some weakness. Also, these approaches are used to test the similarity between two sentences so the similarity between two documents has not been taken into account. In addition, there are methods that calculate the average of the vectors of words in order to have a vector representation of a sentence; this can cause the problem of the loss of meaning. The different publications shown that the use of the doc2vec principle remains the most relevant solution [14], [15]. That's why we took inspiration from it to build our detection plagiarism system.

3. Background Concepts

3.1 Deep learning for Plagiarism Detection

Over the past few years, Deep Learning (DL) architectures and algorithms have made impressive advances in different fields such as image recognition and speech processing. Their application to Natural Language Processing (NLP) was less impressive at first, but has now proven to make significant contributions, yielding state-of-the-art results for some common NLP tasks. Named entity recognition (NER), part of speech (POS) tagging or sentiment analyses are some of the problems where neural network models have outperformed traditional approaches. The progress in machine translation is perhaps the most remarkable among all. Below are the different uses of deep learning at NLP level [23].

Siamese LSTM for Learning documents Similarity: LSTM is a king of recurrent neural network and it is great when we have an entire sequence of words or sentences. This is because RNNs can model and remember the relationships between different words and sentences. Manhattan LSTM models have two networks LSTMleft and LSTMright which process one of the sentences in a given pair independently. Siamese LSTM, a version of Manhattan LSTM where both LSTMleft and LSTMright have same tied weights such that LSTMleft = LSTMright. Such a model is useful for tasks like duplicate query detection and query ranking. Here, duplicate detection task is performed to find if two documents are similar or not. Similar model can be trained for query ranking using hit data for a given query and its matching results as a proxy for similarity [26].

Convolutional neural network: CNN is a class of deep, feed-forward artificial neural networks (where connections between nodes do not form a cycle) that uses a variation of multilayer perceptions designed to require minimal preprocessing. These are inspired by animal visual cortex. CNNs are generally used in computer vision; however, they have recently been applied to various NLP tasks like a text classification [26].

3.2 Doc2vec

In this method, a text is considered as bag of words where there is no more order, and with each word we associate a weight which makes it possible to measure its importance in the text. A text is transformed into a vector in a large space where each coordinate corresponds to the degree of importance of a given word in the text. This new representation contains a major part of syntactic as well as semantic rules of the text data. Much larger units such as "phrases, sentences and documents" should be described as a vector. The paragraph vector learning approach is based on word vector learning methods. The inspiration is

that the vector words are asked to contribute to a prediction task regarding the next word in the sentence. So, despite the fact that the word vectors are randomly initialized, it can eventually capture the semantics as an indirect result of the prediction task. They use this idea in our paragraph vectors in a similar way. Paragraph vectors are also invited to contribute to the task of predicting the next word, in many contexts sampled from the paragraph [27].

4. Proposed Approach

This section contains the description of our approach to detect the plagiarism of ideas validated on PAN Dataset that contains the different types of plagiarism possible; in a first place, we will quote the steps of this approach as follows:

- Context representation of documents
- Deep Learning phase
- Detection of plagiarism

4.1 Architecture of our approach

The figure below represents a global vision of our approach in which we define the different steps that include the learning phase and the test phase; we will detail this architecture in the following paragraphs.

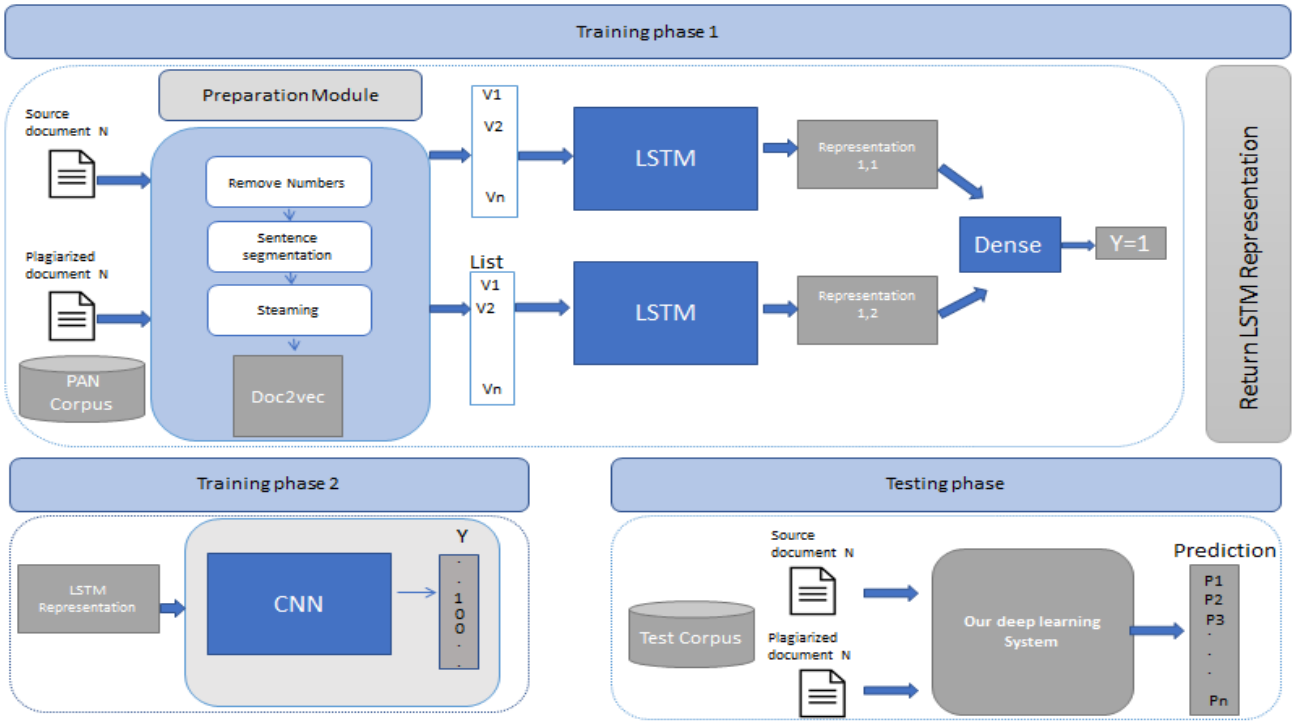

Fig. 1 Global architecture

A- Preparation of the learning base: The first part consists of preparing our corpus of documents which contains the source documents and the plagiarized documents corresponding to the source document. We have chosen “Learning Corpus” as a golden name, this corpus mentions the language used in our learning system, so we can prepare several corpuses with different languages to have a multilingual system.

The initial module is a preprocessing of the dataset composed of the source and suspicious documents. This includes paragraph and sentence segmentation, tokenization, lemmatization and vector construction as explained bellow.

- Sentence Segmentation and Tokenization: Each document is represented as a set of sentences. The tokenization of each source and suspicious sentence is made then.
- Steaming: convert words into their basic dictionary forms for easy comparisons.
- Construct sentence Vector using doc2vec: After the Word2Vec model has proved effective and useful [4,7], so we can easily group and find similar words in a huge corpus, people then began

to think further: is it possible to have a higher level of representation? Sentences, paragraphs or even documents. To do this, we chose to work with the DM (distributed memory) model. We treat the paragraph as an additional word. Then, it is concatenated / averaged with local context word vectors during the prediction. In other words, we treat each document as an additional word; Document ID / Paragraph ID are represented as a single vector; documents are also embedded in the continuous vector space.

B- Training phase: Through several studies, it has been proven that Deep Learning models can achieve an exceptional level of classification accuracy. Models are trained through a large set of labeled data and neural network architectures that contain many layers. Indeed, for the preparation stage, it will be used for the construction of our learning system. Each document will be transformed to a list of sentences which will be representing as list of vectors using the doc2vec principle. More precisely, we will make our system learn the different types of plagiarism existed in the corpus, that is to say, we will build a supervised neural network with an input that contains the source documents and plagiarized documents.
The supervised neural network used for our study is the Siamese LSTM which will take the vector representation of source and plagiarized documents to learn a type of plagiarism. And to consolidate our approach we added another phase which concerns a classification of the types of plagiarism learned in the first part, so we used the CNN to perform this treatment, this part will take the LSTM representation provided by the first phase learning as an input data so that it can launch the classification or accurately labeled this representation to a type of plagiarism. To detect whether a pair of the document is similar or not, it is sufficient to give this system two documents and then it will calculate the probability of plagiarism detected between these two documents, and then it will give us the probability of plagiarism for each type of plagiarism classifies in the second phase of our system, then this learning system is a Siamese LSTM which contains the input data corresponding to the vectors of the source documents and the vectors of the plagiarized documents of our Learning Corpus, each pairs of documents of PAN corpus will be labeled with 1 which means is a type of plagiarism, our system train the different types of plagiarism that exist in the PAN corpus used in our study, then after the execution of learning phase, we will have the features generated as a result which represent different kinds of plagiarism which existed in our Learning Corpus and finally we will classify each representation of a pair of documents provided by the Siamese LSTM by a type of plagiarism using the convolutional neural network. Each pair's LSTM representation will be labeled by a one hot vector that illustrates a type of plagiarism.

5. Test On Pan Dataset

5.1 PAN Corpus

The PAN contains documents in which artificial plagiarism has been inserted automatically. The plagiarism cases have been constructed using a so-called random plagiarist, a computer program which constructs plagiarism according to a number of random variables. The variables include the percentage of plagiarism in the whole corpus, the percentage of plagiarism per document, the length of a single plagiarized section, and the degree of obfuscation per plagiarized section. This corpus contains a number of pairs of documents (source and plagiarism) in our study we will use 75% of these couples for learning then 25% to perform our tests [28].

5.2 Testing Phase

So, we can give use the 25% that stays in PAN, the figure above gives an overall view of our approach. In fact, represents the pair of documents sources and plagiarized of 25% of PAN corpus these two documents will be prepared at the preprocessing phase. And later, these documents will be represented by a list of vectors that will subsequently be as a base of a deep learning system. The system will detect later if the documents inputs are similar or not and it will give us the probabilities of each kind of plagiarism trained in our system.
Finally, if the system detects some kind of plagiarism, it will add this plagiarized document to the corpus of plagiarized documents and the source document to the corpus of source documents to strengthen our system to detect this type of plagiarism later.

6. Validation and Result

In our validation we used python as a programming language, we used keras tensorflow to build our siamese LSTM and CNN and finally we worked with nltk gensim at the preprocessing phase. The model is ready for training; a training loop feeds the dataset examples into the model to help it make better predictions. Iterate each epoch. An epoch is one pass through the dataset. Within an epoch, iterate over each example in the training Dataset grabbing its features (x) and label (y). Using the example's features, make a prediction and compare it with the label. Measure the inaccuracy of the prediction and use that to calculate the model's loss and gradients. Use an optimizer to update the model's variables and repeat for each epoch.

Table 1: Trainig Result

	Loss	Accuracy
Epoch 000	1.077	35.000%
Epoch 050	0.555	70.000%
Epoch 100	0.354	94.167%
Epoch 150	0.229	98.363%
Epoch 200	0.163	**98.333%**

The following graphs represent the learning phase by displaying the growth of the accuracy value and the decrease the loss value which strengths our learning system.

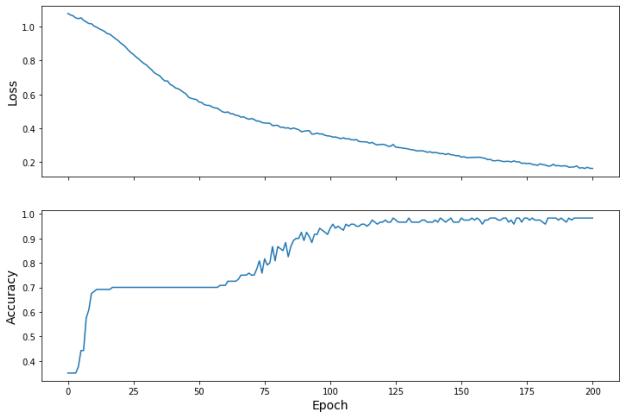

Fig. 2 Loss and accuracy graphs

Unlike the training stage, the model only evaluates a single epoch of the test data; we iterate over each example in the test set and compare the result against the actual label. This is used to measure the model's accuracy across the entire test set, we obtain as test set accuracy: 96.667% which is good, and this illustrates that the learning phase has gone well.

As we have already specified on this article, we used 25% of the PAN data set to perform our test, so we chose just 10 pair of documents for the displayed below in a table that illustrates the probabilities of types of plagiarism the length of this result is 150. The index of the largest probability corresponds to the index of the plagiarism kind.

Table 2: Result of our test

Couple of documents	**Type 1**	**Type 2**	**Type 3**	**Type 4**	**Type 5**	**Type 6**
Pair of documents 1	0.006	0.004	0.62	0.72	0.002	0.009
Pair of documents 2	0.0032	0.78	0.45	0.008	0.005	0.1
Pair of documents 3	0.0041	0.65	0.001	0.21	0.55	0.001
Pair of documents 4	0.0077	0.003	0.001	0.003	0.69	0.052
Pair of documents 5	0.0041	0.008	0.002	0.58	0.002	0.791
Pair of documents 6	0.0047	0.001	0.61	0.008	0.35	0.21
Pair of documents 7	0.78	0.006	0.003	0.034	0.71	0.007
Pair of documents 8	0.0088	0.001	0.001	0.66	0.1	0.56
Pair of documents 9	0.0011	0.001	0.25	0.001	0.20	0.10
Pair of documents 10	0.87	0.006	0.002	0.003	0.1	0.34

Each value represents the probability of plagiarism of the corresponding type, for example the value 0.006 represents the probability of type 1 of plagiarism already trained in our learning phase. This gives us the ability to detect several types of plagiarism at the same time which makes our system more relevant.

7. Conclusion

In this paper, we have mentioned many different methods used in detection of plagiarism of ideas that are based on the Deep Learning principal. This study showed us the interest of the use of deep learning in the detection of plagiarism. As a result, the extraction of characteristics without losing the sense of the document is one of their functions. Indeed, we are capable of detecting these measurements through the result of our neuron network which provide probabilities about each kind of plagiarism already trained in a learning phase. As regard the performance of this approach, compared to other works, this system is capable of detecting different types of plagiarism which is important in the detection of plagiarism of ideas.

Finally, for our future work, we will have consolidated our approach with other tests using other data sets, with different languages; this allows us to develop our approach to be more efficient.

References

[1] Tuomo Kakkonen, Maxim Mozgovoy. Hermetic and Web Plagiarism Detection Systems for Student Essaysan Evaluation Of The State-Of-The-Art. Journal of Educational Computing Research, v42 n2 p135-159 2010. University of Joensuu, Finland, University of Aizu, Japan [en ligne] 2010.

[2] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, Kilian Q. Weinberger. From Word Embeddings To Document Distances? Washington University in St. Louis, 1 Brookings Dr., St. Louis, MO 63130, 2016

[3] S. M. Alzahrani, N. Salim, and A. Abraham, “Understanding plagiarism linguistic patterns, textual features, and detection methods,” Trans. Sys.Man Cyber

Part C, vol. 42, no. 2, pp. 133–149, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1109/TSMCC.2011.2134847.

[4] Uzuner, O., and Katz, B., and Nahnsen, T.: Using Syntactic Information to Identify Plagiarism. In: 2nd Workshop on Building Educational Applications using NLP (2005).

[5] Uzuner, O., and Katz, B., and Nahnsen, T.: Using Syntactic Information to Identify Plagiarism. In: 2nd Workshop on Building Educational Applications using NLP (2005).

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

[7] Bela Gipp State-of-the-art in detecting academic plagiarism. International Journal for Educational Integrity. University of California, Berkeley and University of Magdeburg, Department of Computer Science.

[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] Erfaneh Gharavi, Hadi Veisi, Kayvan Bijari, and Kiarash Zahirnia. A Fast Multi-level Plagiarism Detection Method Based on Document Embedding Representation. Published 2016 in FIRE Workshop. Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran.

[11] 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.

[12] 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.

[13] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, Larry Heck. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. Published by ACM International Conference on Information and Knowledge Management (CIKM).

[14] 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.

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

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

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] Shaojie Bai, J. Zico Kolter, Vladlen Koltun. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv:1803.01271v2 [cs.LG] 19 Apr 2018.
[27] Quoc Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043..
[28] S. Argamon and P. Juola. Overview of the International Authorship Identification Competition at PAN 2011. In V. Petras, P. Forner, P.D. Clough (eds.) CLEF Notebook Papers/Labs/Workshop, 2011.
[29] HAMBI El Mostafa , Faouzia Benabbou. A System for Ideas Plagiarism Detection: State of art and proposed approach. MISC2018. Information Technology and Modeling LaboratoryScience Faculty Ben M'sik Casablanca, Morocco.

Faouzia Benabbou is a professor of Computer Science and member of Compute Science and Information Processing laboratory. She is Head of the team "Cloud Computing, Network and Systems Engineering (CCNSE)". She received his Ph.D. in Computer Science from the Faculty of Sciences, University Mohamed V, Morocco, 1997. His research areas include cloud Computing, data mining, machine learning, and Natural Language Processing. She has published several scientific articles and book chapters in these areas.

El Mostafa HAMBI is a Ph.D. student of Computer Science. His research areas include data mining, deep learning, and Natural Language Processing.