

A New Online Plagiarism Detection System based on Deep Learning

El Mostafa Hambi¹, Faouzia Benabbou²

Information Technology and Modeling Laboratory
Faculty of Sciences Ben M'sik, University Hassan II, Casablanca, Morocco

Abstract—The Plagiarism is an increasingly widespread and growing problem in the academic field. Several plagiarism techniques are used by fraudsters, ranging from a simple synonym replacement, sentence structure modification, to more complex method involving several types of transformation. Human based plagiarism detection is difficult, not accurate, and time-consuming process. In this paper we propose a plagiarism detection framework based on three deep learning models: Doc2vec, Siamese Long Short-term Memory (SLSTM) and Convolutional Neural Network (CNN). Our system uses three layers: Preprocessing Layer including word embedding, Learning Layers and Detection Layer. To evaluate our system, we carried out a study on plagiarism detection tools from the academic field and make a comparison based on a set of features. Compared to other works, our approach performs a good accuracy of 98.33 % and can detect different types of plagiarism, enables to specify another dataset and supports to compare the document from an internet search.

Keywords—Plagiarism detection; plagiarism detection tools; deep learning; Doc2vec; Stacked Long Short-Term Memory (SLSTM); Convolutional Neural Network (CNN); Siamese neural network

I. INTRODUCTION

According to Risquez et al. [1] “the Plagiarism is conceptualized as the theft of others’ words or ideas without citing the proper reference and thus without giving the accurate credit to the original author”. Depending of depth of transformation performed on the original text, the plagiarism can be classified into five categories [2]:

- Copy & paste plagiarism (word by word) [3]: it is the act of copying text and passing without reference of original authors.
- Paraphrasing [4]: the content is copied from different source without acknowledging authors.
- Use of false references [5]: There are certain cases where user quotes the original sources, but the information provided in the articles are not match with the source provided at the end of the article.
- Plagiarism with translation [6]: it is the act of translating text from language to another.
- Plagiarism of ideas [7]: it is the most difficult plagiarism to detect where fraudsters steal other authors' ideas and present them in a fully modified version of the original text and own the new version.

Plagiarism is applied in different areas such as literature, music, software, scientific articles, newspapers, advertisements, websites, etc. Despite the sanctions applied in cases of cheating and plagiarism in Bulgarian universities, more than 50% of teachers believe that these procedures are not efficient [8]. As the use of internet increases plagiarism becomes a big challenge in schools, universities to maintain the academic integrity. Thus, the use of efficient plagiarism detection tools has become very urgent in many higher education institutions. However, the effectiveness of these plagiarism detection systems depends on their ability to discover different fraudsters’ strategies to modify the text without changing its semantics [9].

As part of NLP research topic, the plagiarism detection methods are based on natural language techniques to process and analyze the structure of documents. Many solutions have been proposed for plagiarism detection, and most of them are based on concept extraction using corpus such as ontologies (e.g. WordNet) to perform a semantic representation of documents. However, these approaches depend on the quality of corpus and an appropriate annotation to choose the best concept that semantically represents a word. In addition, the problem of ambiguity may arise when choosing the concept that semantically represents the word, so the meaning of the processed sentences may be lost if we choose the wrong concept [10]. Some examples of this classical plagiarism detection methods are [11]: Fingerprinting, String matching, Jaccard similarity, Bag of word analyzing and Shingling.

With the emergence of artificial intelligence, many techniques have been proposed, ranging from supervised, unsupervised machine learning techniques to deep learning, and have been successfully applied in various fields. In-depth learning provides models with multiple processing layers capable of learning data representations with multiple levels of abstraction. Recently many applications of deep learning in NLP domains, has been proposed and their performance was very encouraging as Chatbots programming, sentiment analysis and Question and Answering. In this context, we propose an online plagiarism detection system based on Doc2vec technique for word embedding, and SLSTM and CNN deep learning algorithms. Our system can perform many tasks of plagiarism detection and the results found are very promising.

The rest of this paper is organized as follows. Section 2 presents a review of the most relevant plagiarism software. Section 3 illustrates our plagiarism detection system. In Section 4 we describe the components of our online system.

Section 5 draws some interpretations about the current state of existing discovery tools and compare them with our system. In the section 6, we finish by a global conclusion.

II. PLAGIARISM TOOLS REVIEW

A. Software Description

In the context of academic plagiarism, few tools are proposed, and this section is devoted to describing the most recognized in the scientific community and in different universities. In our latest state of the art we focused on the proposed systems for plagiarism detection based on deep learning, unfortunately we did not find any implementation of these systems. Asim M. El Tahir Ali et al. have proposed an interesting comparative study from five plagiarism detection tools [13]: PlagAware, The PlagScan, CheckForPlagiarism.net, iThenticate and PlagiarismDetection.org. Inspired by this research, we conducted an overview of the top plagiarism detection tools based on some important criteria that a good system would have. Firstly, we used the comparison parameters in [13] as:

- Add a new database is the ability to add a new database in comparison and plagiarism detection.
 - Add a new corpus is the ability to add a new corpus for learning to detect other types of plagiarism.
 - Internet Checking is the ability to use internet results in plagiarism detection.
 - Academical Checking is used to check the research publications and compare them to already published papers.
 - Multiple document comparison is the capacity of software to support multiple document comparison.
 - Multiple language support is the ability to support multiple language in document analysis.
 - Sentence Structure/synonymy show that software detection is capable to make sentence structure and synonymy analysis.
 - In our study, we include other parameters to evaluate the relevance of the plagiarism detection tools:
 - Types of plagiarism to detect is a feature which allows the selection of the type of plagiarism to be detected.
 - Machine learning means a machine learning model used in the approach.
 - Similarity based means if the software is based on matching techniques and similarity measurement.
 - Free license or not.
 - Size limitedness describes if the size of the document is limited (e.g. some tools limit the size document to 1000 words).
 - Document file is the file format to be analyzed (e.g. txt, pdf, docx, etc.).
- Classical methods use a corpus to extract the concepts, but recent researches rely on word embeddings techniques as Word2Vec, GloVe, BERT, to preserve the semantic and the syntactic context of the text.
 - Type of plagiarism detected presents whether if the software displays the types of plagiarism encountered or not and gives the rate each type checked.
 - Reports generation describes if the software exports the results as a report.
- 1) *The PlagAware tool*: It is an online tool [12] that uses a classic search engine to detect plagiarism and offers several reports helping the user to decide if the analyzed text is plagiarized or not. It is possible to add new database, to check documents from the internet results, and to compare multiple documents. Verifying sentence structure analysis and synonymy replacement is not supported. The languages supported are German, English and Japanese. It is used in universities to check the originality of the works to be published. PlagAware performs a complete scan of the document, and each sentence is analyzed to subsequently detect whether it contains plagiarism or not.
- 2) *The PlagScan tool*: It is an online tool used for academic plagiarism detection. This tool uses a local database that include millions of documents and includes the results of the internet search for making comparison. It supports adding a new database over the internet. It detects several types of plagiarism such as: copy and paste or words switching [14]. PlagScan supports the UTF-8 encoding languages and all Latin or Arabic languages. It is used in universities to check the originality of the works to be published. Sentence structure analysis and synonymy replacement are not supported. This tool uses a plagiarism detection algorithm that contains three consecutive word matches to subsequently detect plagiarism methods which use the replacement the words by their synonyms. In addition, they apply matching algorithms to detect documents similarity.
- 3) *The CheckForPlagiarism.net tool*: It is a tool for detecting academic plagiarism developed by a professional academic team. It can detect several types of plagiarism. It uses its own database that include millions of documents from several databases with different domains. It performs an internet Checking, Sentence structure analysis and synonymy replacement is detected, and it is possible to compare multiple documents. CheckForPlagiarism.net checks several types of documents, including, newspapers, PDFs, magazines, journals, books, articles etc. It supports several languages: Spanish languages, Portuguese, German, English, Korean, French, Italian, Arabic and Chinese languages. Each document is assigned by a fingerprint and used in document comparison [15].
- 4) *The iThenticate tool*: iThenticate is an online academic plagiarism detection tool for researchers, publishers and authors [16]. It possible in iThenticate to add a new database or use Internet for comparison and in addition it uses its own

database that contains several documents like books, newspapers and articles. Sentence structure analysis and synonymy replacement checking is not supported by iThenticate but it is possible to compare multiple documents. It supports more than 30 languages likes English, Russian, Arabic, etc. Many online scientific journals use it for submitted papers checking. iThenticate performs a matching advanced technics in similarity analysis highlight material within a manuscript that matches documents found in the iThenticate database algorithm to check the contents of a document against an extensive database of published scholarly writing [16].

5) *The PlagiarismDetection.org tool*: This is an online tool that is mostly used by teachers and students [17]. It used its own database that contains millions of documents, but it is possible to add a new database or use internet Checking. Sentence structure analysis and synonymy replacement checking is not supported neither multiple document comparison. It supports all languages using Latin characters. the technique is based on the n-gram method.

6) *The Urkund tool*: URKUND is a web plagiarism prevention system. Today, a vast majority of universities around the world use Urkund to effectively and detect

plagiarism. This system allows to compare the content of a document with several other resources from different sources (Internet, database, internal documents, etc.). Document formats accepted is doc, docx, pdf, etc. Urkund is multilingual, detects plagiarism by paraphrasing and replacements by synonyms, and returns the rate of similarity with the other documents [21].

7) *The Turnitin tool*: The Turnitin Plagiarism Detection System allows users to check their documents and compare them with web content and other documents that have already been downloaded by institutions as well as with certain journals [22]. For each submission, a report is produced identifying the sources of its similarities as well as the percentage of correspondence with the submitted document. Turnitin uses a matching algorithm to find strings of words within assignments that are identical to those within its repository.

B. Comparison and Analysis

In this section we propose a qualitative comparison of plagiarism software detection. We focus on the features and properties of the tools rather than their performance in the first instance. Based on the comparison parameters cited above the results are reported in Table I.

TABLE I. COMPARISON OF THE PLAGIARISM DETECTION TOOLS

Features	PlagAware	PlagScan	iThenticate	CheckForPlagiarism.net	Plagiarismdetecting.org	URKUND	Turnitin
Add new database	x	x	x	x	x	✓	✓
Add a corpus for new training	x	x	x	x	x	x	x
Internet Checking	✓	✓	✓	✓	✓	✓	✓
Academical Checking	✓	✓	✓	✓	✓	✓	✓
Multiple document comparison	✓	✓	✓	✓	✓	✓	✓
Multiple languages	✓	✓	✓	✓	✓	✓	✓
Sentence structure/ synonymy	x	x	✓	✓	x	✓	✓
Types of plagiarism	x	x	x	x	x	x	x
Machine Learning	x	✓	x	x		✓	x
Free License	x	✓	✓	✓	x	x	✓
Similarity based	✓	x	✓	x	x	x	✓
Size limitedness	✓	✓	✓	✓	✓	x	x
All type of files	✓	✓	✓	✓	✓	✓	
Classical method	✓		✓	✓	✓	x	✓
Type of plagiarism detected	x	x	x	x	x	x	x
Reports generation	✓	✓	✓	✓	✓	✓	✓

From Table I, we can see that all studied plagiarism detection tools can perform Internet Checking to verify if there is any similarity with any resources on internet. Also, the document analyzed can be written in Multilanguage. These systems are almost used in the Academical context to check student reports, thesis, or research papers. Multiple documents comparison is also provided by these tools. But as we see, most of them does not a have the feature of adding a new corpus. This new feature enables adding a corpus to be used as the basic dataset for the plagiarism detection step. It is an opportunity to use more corpus for improving the learning phase. The new corpus contains a source document, suspicious documents and the type of plagiarism. As we can see in Table I, none of the analyzed tools specify the type of plagiarism that has been detected from sources, nor give the user the possibility to specify the type of plagiarism he wants to be detected. Based on this comparison and to benefit from our previous work [18], we propose an implementation with new features to deal with a plagiarism in textual documents. In the next section, we describe the background of the approach and its components and the services that our framework can provide.

III. GENERAL FRAMEWORK OF OUR APPROACH

The proposed plagiarism detection tool is based on our previous research validated with PAN Dataset where data are labeled with the types of plagiarism [20]. Fig. 1 represents a global architecture of our framework which is based on two Deep learning architectures Siamese Long Short-Term Memory (SLSTM) and Convolutional Neural Networks (CNN).

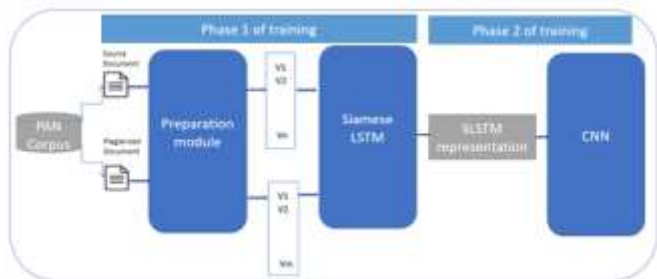


Fig. 1. Global Architecture.

The approach based mainly on three steps as described below:

- **Context representation of documents:** The corpus consists of a set of source documents and a set of suspicious documents plagiarized from each source using a specific kind of plagiarism. Both of sources and the plagiarized document are transformed with doc2vec a list of sentences vectors to be used as input to the SLSTM model.
- **SLSTM Learning phase 1:** The Siamese LSTM is used to learn the different kind of plagiarism in dataset.
- **CNN Learning phase 2:** The output of the first stage is a SLSTM representation of documents. To consolidate our approach, we used CNN deep learning model to detect the types of plagiarism learned in the first part of

the approach. The goal is the classification of the document as plagiarized or not with the type of plagiarism detected in it if yes.

IV. DEEP LEARNING PLAGIARISM DETECTION SYSTEM

In this section we will present the proposed plagiarism detection framework by illustrating the technical architecture and its different layers. Fig. 2 shows an overview of the proposed system. The system is composed by the following six layers: Front-end Layer, http layer, Controller Layer, preprocessing layer, Learning layer and Detection Layer. Here bellow, we present the description of each layer and its implementation.

A. Front End Layer/ Http Layer/ Controller Layer

The Front end is a platform for building mobile and desktop web applications that communicate with the http layer which offers web services to consume. The flask package provides some classes to build a Service layer and exposes an API that interacts with the model. The first idea is to remove all logic of the routes and model of the Flask application and put it in the service layer. The second goal is to provide a common API that can be used to manipulate a model regardless of its storage backend. The controller layer concerns a middleware between the flask layer and the other layers of our system.

B. Preprocessing Layer

At first, the corpus is preprocessed as shown in Fig. 3. For each document we realize the cleaning, segmentation and stemming phases [18]. Then the output is given as input to the doc2vec word embedding model layer.

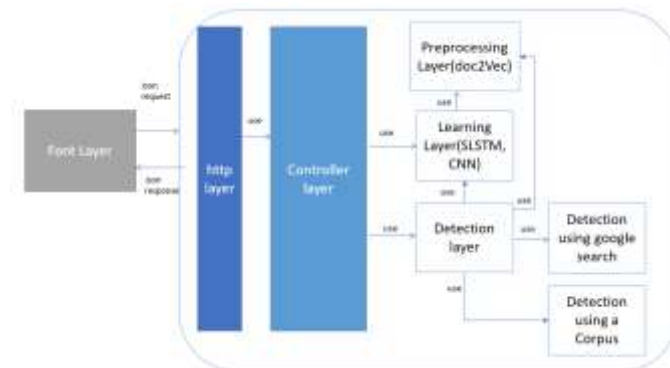


Fig. 2. The Distribution of Application Layers.

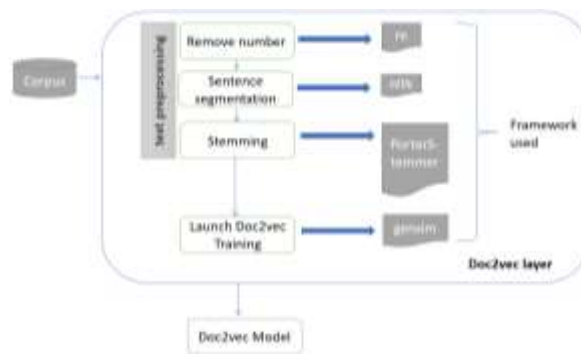


Fig. 3. Data Preprocessing and Vectorization.

Then we launch the training phase to generate a doc2vec model which we will use later to transform each sentence of a document to a vector. We worked with the re framework to build a regular expression that removes numbers, nltk to segment a document by sentence, PorterStemmer to apply the stemming principle that makes a word in the initial form and gensim to start training the doc2vec model.

C. Learning Layer

In this layer, we applied twice the learning process as shown in Fig. 4. In the first step, we used SLSTM algorithm for learning from the output of doc2vec and the output is given to CNN Model to learn again to build our efficient learning model. At the end of this phase we restore the SLSTM model which will be used to test whether a pair of documents are similar or not and we also get the SLSTM representation. In this step we used the keras tensorflow.

To carry out the classification of documents and add the types of plagiarism that have been detected, we used the keras tensorflow to build our CNN model. Hence, the outputs of the SLSTM model are used in the second learning phase which consists of classifying the types of plagiarism already learned in the first part.

D. Detection Layer

For document classification task (whether is plagiarized or not), the users can make choice to use a new corpus, internet search results or a new corpus for comparison. The corpus contains a list of sources documents that will be used in learning step or to search for similarity with the text to be verified. The second option uses python Google package to get the link of the first n search results and compare the text analyzed with the contents of these links. More details will be given in the next subsections.

1) Add a new corpus: To add a new corpus, we respect the process in Fig. 5. Firstly, the user adds the pairs file, which is a text file that contains several lines and each line represents a type of plagiarism. Secondly, the user uploads the corpus containing the source and plagiarized document mentioned in the pair document above.

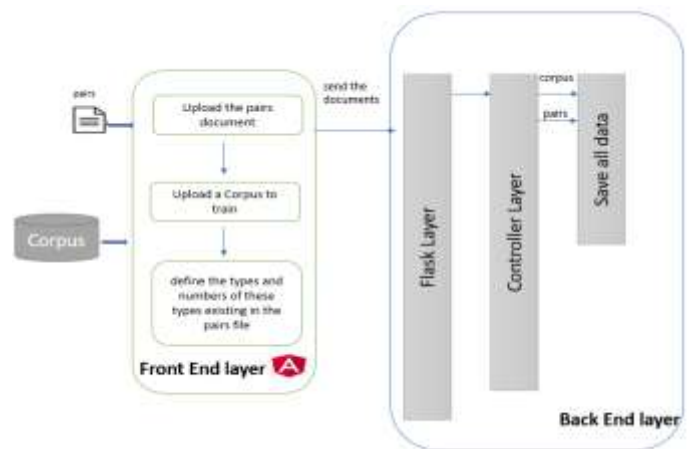


Fig. 5. Add New Corpus.

Finally, the user defines the types and numbers of plagiarism cases. But the number of plagiarism types entered must correspond to the number of lines existing in the pair file. After adding a new corpus, we can launch the training phase which follows the process in Fig. 6.

2) Add a new corpus for comparison: Our framework can also compare a document to a special corpus containing a set of desired source documents to compare with. We must first add corpus which will be the basis of comparison, and the system will compare the document to each document in corpus to detect a kind of plagiarism. Fig. 7 presents the process of this task. The comparison is carried out by using the following steps:

- Select corpus trained and corpus of comparison.
- Segment the analyzed document to a list of paragraphs.
- Retrieve a list of paragraphs for each document in corpus of comparison.
- Using our deep learning system, we compare each paragraph of the analyzed document with all the paragraphs returned via the corpus of document.

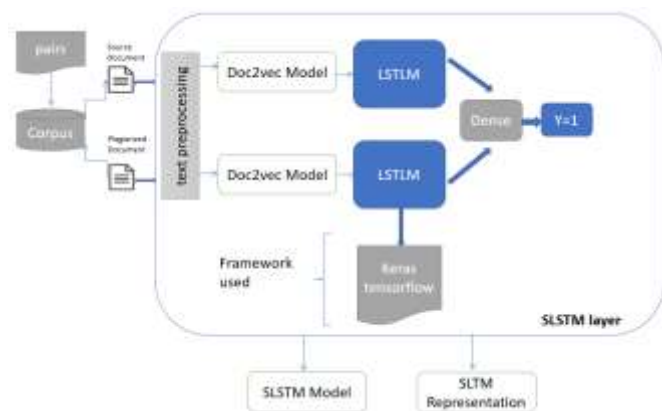


Fig. 4. Learning with SLSTM Model.

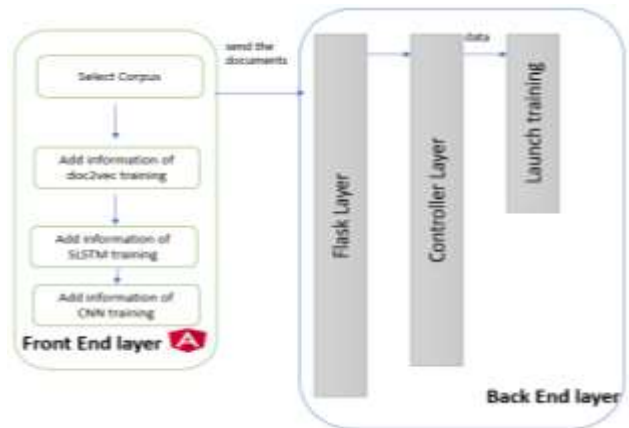


Fig. 6. Launch Training.

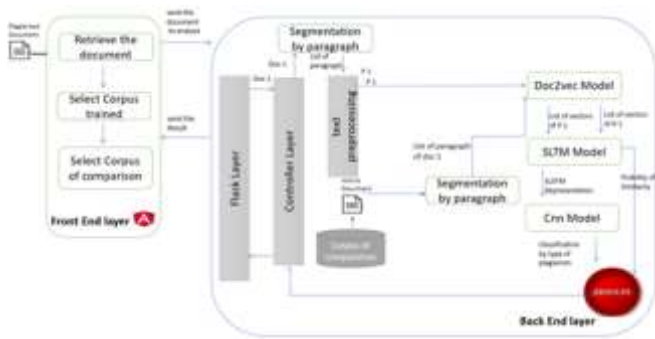


Fig. 7. Plagiarism Detection from Corpus Process.

3) *Using google research engine:* Our system can also detect the plagiarism in documents using google search result as illustrate in the following Fig. 8.

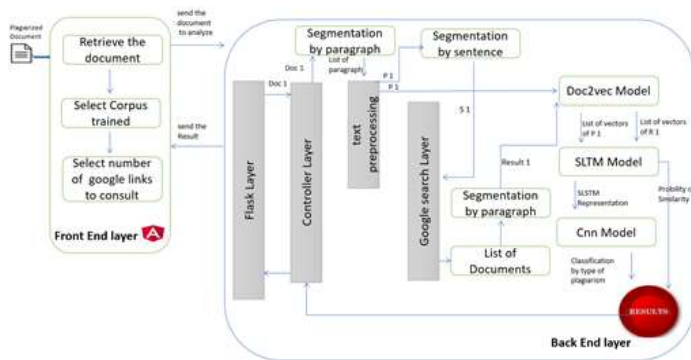


Fig. 8. Check Plagiarism from Internet.

The plagiarism detection consists of the following steps:

- Segment the analyzed document to a list of paragraphs and the list of sentences.
- Use the sentences in this paragraph to retrieve the various links which contain the suspected texts.
- Retrieve a list of paragraphs for each link found.
- Using our deep learning model to compare each paragraph of the analyzed document with all the paragraphs returned via the Google searches.

More precisely we assume that the document contains N paragraphs, if for example the first paragraph contains S sentences, so we launch S internet search to retrieve $S \times N$ result then we assume once again that each result will offer us P paragraphs which are considered as suspected initials. So, the first paragraph of the analysis document is compared with $N \times S \times P$ paragraph.

V. EXPERIMENTS

In this section, we present different possibilities that our system provides in terms of plagiarism detection. We can proceed three kind of comparison: Two text comparison, online comparison and using an intern corpus for a comparison.

A. Add New Corpus

For this task we proposed the following IHM in Fig. 9:

Fig. 9. Add New Corpus.

B. Training a New Corpus

To do that, we must fill some information about corpus, doc2vec training, SLSTM training and finally CNN training. The data requested are used to develop the accuracy rate of our training. For this phase we proposed the following IHM in the Fig. 10.

Fig. 10. IHM to Launch New a New Training.

This part contains hyperparameters used to adjust the three models in learning process, for more information see [19][14].

C. Comparison of Two Texts

Given two documents, we can make a comparison of two given documents by following the steps in Fig. 11 and Fig. 12. The two documents will be preprocessed and converted to a list of vectors with doc2vec model. The system will detect later if the input documents are similar or not using SLSTM Model and it will report the probabilities of each kind of plagiarism trained in our system when we use CNN Model. Fig. 13 provides an example of two documents comparison.



Fig. 11. Two Documents Comparison.

Result
Value of Similarity: 0.1246744

Type	Similarity
NUMBER_NO_PLAGIARISM	0.1137468
NUMBER_NO_OBFUSCATION	0.1116743
NUMBER_RANDOM_OBFUSCATION	0.1124852
NUMBER_TRANSLATION_OBFUSCATION	0.1117394
NUMBER_SUMMARY_OBFUSCATION	0.113384
NUMBER_NO_PLAGIARISM10	0.1133344
NUMBER_HUMAN_RETELLING	0.1118285
NUMBER_SYNONYM_REPLACEMENT	0.1136874
NUMBER_CHARACTER_SUBSTITUTION	0.1130215

Fig. 12. Results of Comparison of Two Documents.

And we get the result below which result contains the probability of similarity between these two texts, in fact, we also recover the probabilities of each type of plagiarism learned at the training phase.

D. Online Checking

For performing plagiarism detection from documents returned from Google research engine, we need to fix several parameters as the learned corpus, number of sites to consult and finally the text to analyze, as mentioned below it proposed an IHM in Fig. 13.

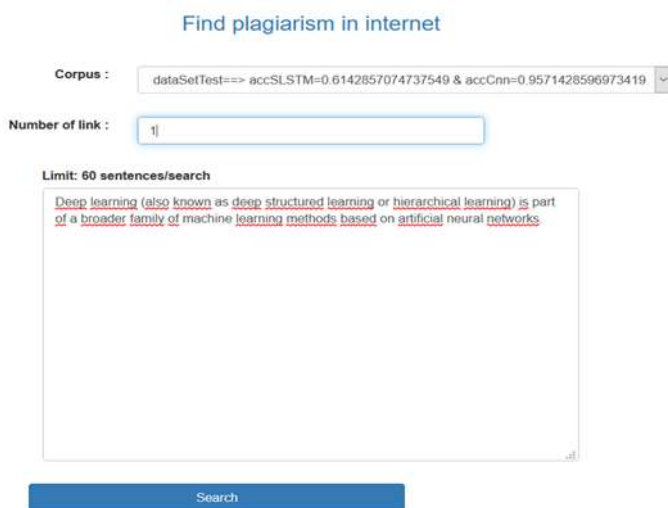


Fig. 13. Plagiarism Detection from Internet IHM.

The results in Fig. 14 shows the source text, link of the source text, the probability of similarity. In the right, the table presents the probabilities of each type of plagiarism learned in the training phase present in the document.

E. Using Corpus for a Comparison

Fig. 15 below represents the result of detection of plagiarism using a corpus of source documents instead of consulting the results of the internet. The results consist of a list of blocks containing the following information:

- the paragraphs analyzed.
- the name of the source document.
- probability of similarity.

In addition, we propose a table in Fig. 16 containing the probabilities of each type learned in the training phase.

Result

Type	Similarity
NUMBER_NO_PLAGIARISM	0.2223422
NUMBER_NO_OBFUSCATION	0.2145277
NUMBER_RANDOM_OBFUSCATION	0.2223422
NUMBER_TRANSLATION_OBFUSCATION	0.2223422
NUMBER_SUMMARY_OBFUSCATION	0.2223422
NUMBER_NO_PLAGIARISM10	0.2223422
NUMBER_HUMAN_RETELLING	0.2223422
NUMBER_SYNONYM_REPLACEMENT	0.2223422
NUMBER_CHARACTER_SUBSTITUTION	0.2223422

Fig. 14. Example of Plagiarism Detection from Internet.



Fig. 15. Plagiarism Detection from Corpus IHM.



Fig. 16. Example of Plagiarism Detection from Corpus.

F. Proposed System Features

In comparison with existing systems, our plagiarism detection system has all the properties used in the comparison above. We have added new features making it an able to make followed action:

- Upload and Add Any Dataset.
- Add New Corpus for Training Plagiarism.
- Internet Checking.
- Academical Checking: We can add the corpus of publication or get them through Google result.
- Two documents comparison but it could be extended to more than two.
- Multiple languages detection: We can use any language, but you must choose the corpus already trained by this language.
- Check all type of plagiarism.
- Personalize the types of plagiarism to detect: We can define several kinds of plagiarism in our training phase.
- Use the deep learning approaches: our approach uses deep learning algorithms.
- Document size is limited: not limited.
- Show the type of plagiarism detected: Yes.
- Reports generation: Yes.

VI. CONCLUSION

In this paper, we proposed a new system for the detection of plagiarism based on the deep learning methods. Its interest is the extraction of characteristics without losing the sense of the document by using doc2vec word embedding technique. The proposed system has the ability to detect not only that there is plagiarism but also the probabilities of the existence of each type of plagiarism. We presented the different functionalities offered by our system, either at the level of the personalized learning phase or the different ways of detecting

plagiarism offered. Compared to the other tools studied in this paper, our proposition offers more functionalities as adding and training new corpus or using a special corpus for comparison. As for our perspectives, we will improve the various interfaces of the application to make it more accessible to the general public and improve the response time due to the learning time. It would also be interesting to compare the performance of different approaches in a quantitative way.

REFERENCES

- [1] Risquez, A., Dwyer, M. O.; Ledwith, A. (2011). «Thou shalt not plagiarise»: from self-reported views to recognition and avoidance of plagiarism». *Assessment & Evaluation in Higher Education*, vol. 2, no. 1, p. 34-43. <http://doi.org/10.1080/02602938.2011.596926>. 3 Ruipérez, G.; García-Cabrero, J.C. (2016). «Plagiarism and Academic Integrity in Germany». *Comunicar*, vol. 24, no. 48, p. 9-17. <http://doi.org/10.3916/C48-2016-01>.
- [2] Liddell, J. (2003). *A Comprehensive Definition of Plagiarism*. *Community & Junior College Libraries*, 11(3), 43–52. doi:10.1300/j107v11n03_07.
- [3] Thomas Lancaster, Fintan Culwin. *A Visual Argument for Plagiarism Detection using Word Pairs*. School of Computing University of Central England Perry Barr Birmingham B42 2SU United Kingdom. Faculty of Business, Computing and Information Management London South Bank University Borough Road London SE1 0AA. *Plagiarism: Prevention, Practice and Policies 2004 Conference*.
- [4] Zubarev D.V. Sochenkov I.V. *Paraphrased plagiarism detection using sentence similarity*. Federal Research Center “Computer Science and Control” of Russian Academy of Sciences, Moscow, Russia.
- [5] Maxim Mozgovo, Tuomo Kakkonen, Georgina Cosma. *Automatic student plagiarism detection: future perspectives*. university of aizutsuruga, ikki-machi, aizu-wakamatsu, fukushima, 965-8580 japan. *articleinjournal of educational computing research* · january 2010.
- [6] Sousa-silva, r. -detecting translanguing plagiarism and the backlash against translation plagiaristslanguage and law / linguagem e direito, vol. 1(1), 2014, p. 70-94.
- [7] Eman s. Al-shamery, Hadeel qasem Ghani. *plagiarism detection using semantic analysis*. published 2016. *computer science indian journal of science and technology*. doi:10.17485/ijst/2016/v9i1/84235 corpus id: 55709933.
- [8] Roumiana Peytcheva-forsyth, Harvey Mellar, Lyubka Alekseiva. *using a student authentication and authorship checking system as a catalyst for developing an academic integrity culture: a bulgarian case study*. *journal of academic ethics* 17, 245-269(2019).
- [9] Amruta Patil, Nikhil Bomanwar. *survey on different plagiarism detection tools and software’s*. computer department, mumbai university. (ijcsit) *international journal of computer science and information technologies*, vol. 7 (5) , 2016,2191-219.
- [10] Hage, J. rademaker, P Vugt, n. *a comparison of plagiarism detection tools*, tech-nical report uu-cs-2010-015,june 2010, department of information and comput-ing sciences utrecht university, utrecht, the netherlands.
- [11] G. Bela, “state-of-the-art in detecting academic plagiarism. *international journal for educational integrity*,” vol. 9no.1 june, 2013 pp. 50-71 issn 1833-2595.
- [12] <https://www.plagaware.com/>(lastaccessfebruary 1, 2020).
- [13] Asim m. El tahir Ali, Hussam m. Dahwa Abdulla, and Vaclav Snasel. *overview and comparison of plagiarismdetection tools*. department of computer science, v̄sb-technical university of ostrava,17. listopadu 15, ostrava - poruba, czech. dateso 2011, pp. 161–172, isbn 978-80-248-2391.
- [14] [http://www.plagscan.com.](http://www.plagscan.com/) (last access february 1, 2020).
- [15] <http://www.checkforplagiarism.net> (last access february 1, 2020).
- [16] <http://www.ithenticate.com/index.html> (last access february 1, 2020).
- [17] <http://www.plagiarismdetection.org> (last access february 1, 2020).

- [18] El mostafa Hambi and Faouzia Benabbou. a multi-level plagiarism detection system based on deep learning algorithms. information technology and modeling laboratory faculty of sciences ben m'sik casablanca, morocco. international journal of computer science and network security. vol. 19 no. 10 pp. 110-117.
- [19] Hambi el Mostafa , Faouzia Benabbou. a system for ideas plagiarism detection: state of art and proposed approach. misc2018. information technology and modeling laboratory science faculty ben m'sik casablanca, morocco.
- [20] Martin Potthast, Matthias Hagen, Anna Beyer, Matthias Busse, Martin Tippmann, Paolo Rosso, and Benno Stein. overview of the 6th international competition on plagiarism detection. web technology & information systems, bauhaus-universität weimar, germany natural language engineering lab, universitat politècnica de valència, spain. at pan 2014.
- [21] <https://www.orkund.com/the-orkund-system/>.
- [22] Dr. Karl o. Jones. practical issues for academics using the turnitin plagiarism detection software. international conference on computer systems and technologies.