[1] Shahed Teko
[2] Khaled Omar

Arabic Plagiarism Detection using Text Summarization and Arabic word Embeddings

Journal of Electrical Systems

Abstract: - Plagiarism detection has become a latest research area in Natural Language Processing field. In today's with the huge available content of Arabic articles on the internet this make the text plagiarism is so easy and Spread widely in academic society, so many algorithm have been developed to decrease this Harmful habit, in this article we have developed an algorithm to detect plagiarism in Arabic text using Arabic text summarization and Arabic text embedding , that the proposed algorithm contains two main stages for plagiarism detection for the first stage the algorithm utilized T5 model for Arabic summarization , and for the second stage the algorithm used Arabic text embedding technology to convert Arabic text to numeric vectors (which taking into consideration keeping the syntax and the meaning of sentences(to calculate the similarity score between origin and suspected files, we have tested our proposed algorithm on Arabic dataset which contains origin and suspected files and the accuracy was about 90%.

Keywords: Arabic plagiarism detection, Arabic text summarization, Arabic word embedding.

I. INTRODUCTION

Plagiarism is to take someone work or research without any mention to the reference or without any citation to that work, this phenomenon has become widespread in the academic community and has a significant negative impact on the quality of education and scientific research.

Plagiarism detection is the process which done to detect plagiarism this detection can be done manually which need huge resources (human/time), or can by done automatically by using technology, that many systems have been developed to detect plagiarism in large number of languages including Arabic language.

The Arabic language is one of the Semitic languages and is characterized by its great influence on literature and culture. There are several attempts and systems that have been developed in the field of detecting plagiarism in the Arabic language, especially in scientific research, to maintain the quality of higher education and scientific research in the Arabic language.

The rest of this paper will be related work, proposed algorithm, evaluation, conclusion and future work.

II. RELATED WORK

In general Plagiarism detection systems have one of the two approaches:

1. External Plagiarism detection systems: in these systems suspicious files are compared with a collection of files to detect similarity and detect plagiarism if exist [1].

2. Intrinsic Plagiarism detection systems: in these systems the suspicious file is not compared with any collections of files, but the file is analyzed to get the writing stylometry, and these systems reports changes in writing stylometry as indicator of potential plagiarism [2].

Plagiarism detection algorithms could be classified according to the methodology to the following classes:

- Fingerprint plagiarism detection algorithms [3] [4] are the most popular applied algorithm in external plagiarism detection systems, that in these types of algorithms divides the file to its paragraphs and then every paragraph is divided to its sentences, and for every sentence a hash code is generated to represent this sentence(in many algorithms the code is generated on the word level), by the end of the scanning of the file , the output file is a sequence of hash codes, this code is called fingerprint code, that the fingerprint algorithm convert the suspected file and the origin file to hash codes, then by using the string matching algorithms , the fingerprint algorithm tries

[1] Syrian Virtaul University, Syria. Shahd.n.tako@gmail.com

[2] Damascus University, Syria. kh.om.mail@gmail.com

Copyright © JES 2024 on-line: journal.esrgroups.org

to find the similarity sections between the two files, the longest common subsequence (LCS) [5][6] is the most widely used algorithm to find the similarity between string series

- String matching plagiarism detection algorithms: in these types of algorithms the we use String similarity matching algorithms to detect plagiarism, there are many developed algorithm to detect similarity between string patterns, such as Jaro-Winkler Distance algorithm which do string comparison by put it in certain mathematical functions, Some algorithms which based on the string metric, include Needleman-Wunsch distance, Jaro-Winkler distance, Rabin-Karp algorithm. The advantage of the Rabin-Karp algorithm compared to other string-matching algorithm is the ability to search for multiple string patterns [7]
- Bag of words: in these types of algorithms the texts (suspected and origin files) are converted to new representation, that these representation the order of words is lost because it count the vocabulary existence in files or into file sentences, and then it measure the distance of these matrixes to detect plagiarism if exist between files[8].
- Citation based plagiarism detection algorithms also is used widely to find the plagiarism between files, the following figure shows the citation pattern comparison [9] [10]:

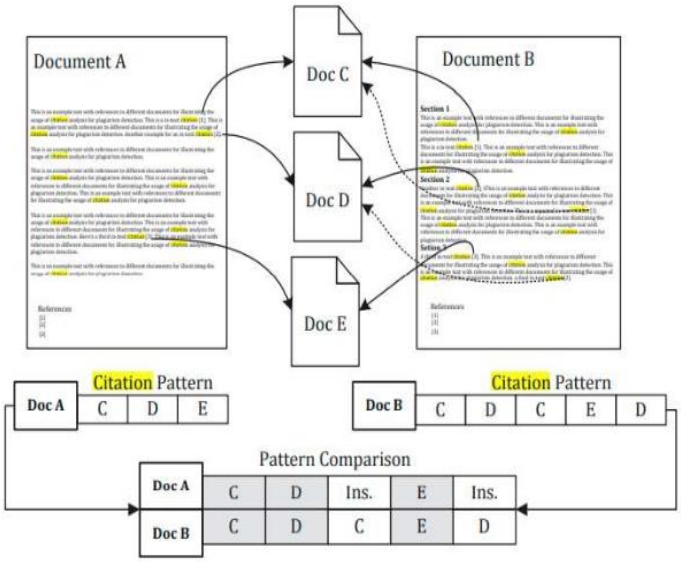

Figure 1. Depiction of Citation based plagiarism detection.

That these algorithms first scan the suspected file, and the origin file, then it generates a citation pattern for every file, then the algorithm compares between files citation patterns, by using one of the string-matching algorithms such as LCS [5][6].

That from the figure above shown that document A, document B include citation to references files C, D, E, this leads that the document A and document B are containing or talking about the same subject in semantic, and if the order of the citation of references in the two documents is identical this leads. Plagiarism detection using Rhetorical Structure Theory: in these types of plagiarism detection algorithms the detection is done based on the analyzing of text (this theory works in Arabic and English languages) to primary sentences and secondary ones, based on list of words called connectors, that these connectors generates many types of relations between sentences [11], the following table contains the types of relations between sentences (which are in English and Arabic Texts).

III. PROPOSED ALGORITHM

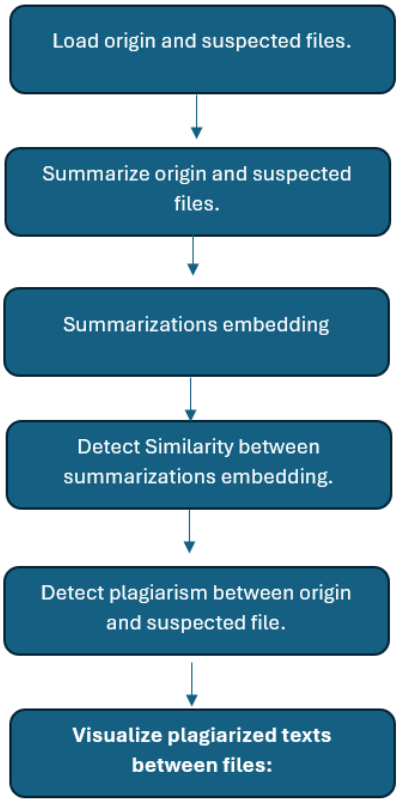

Figure 2. proposed algorithm workflow.

3.1 Load input files (origin and suspected file)

In this step the algorithm loads the input files, source file and suspected file, which are from txt format.

3.2 Summarize origin and suspected files:

In general text summarization can be classified in two main types, extractive text summarization and abstractive text summarizing based on the methodology which used in the summarization, that in extraction methods it involves concatenating sentences (part of sentences) taken from the input file to generate the final summary, whereas abstraction methods it involves generating novel sentences from information extracted from input file to generate the final summary.

The Unified Text-to-Text Transformer (T5) model was used as the main model for abstractive text summarization. The specific version of T5 which we used is called t5-arabic-base model which was trained on Arabic text. The dataset used in training is obtained from Aljazeera website and consists of news articles with their respective summaries. The obtained results were very encouraging [12].

By the end of this step the origin and suspected file are summarized using T5 model.

3.3 Summarizations embedding:

In this step the summarization of the origin and suspected files are converted to numeric representation using text transformation, the transformation is done by using a famous model which is Arabic Bert Embeddings AraBERT [13] is an Arabic pretrained language model based on Google's BERT architecture, There are two versions of the model, AraBERTv0.1 and AraBERTv1, with the difference being that AraBERTv1 uses pre-segmented text where prefixes and suffixes were splitted using the Farasa Segmenter.

AraBERT has been evaluated on many tasks such Sentiment Analysis on 6 different datasets (HARD, ASTD-Balanced, ArsenTD-Lev, LABR), Named Entity Recognition with the ANERcorp, and Arabic Question

Answering, AraBERT have proven to be very challenging to tackle. Recently, with the surge of transformers-based models [14].

By the end of this step the summarization of origin and suspected files are converted to numeric representation.

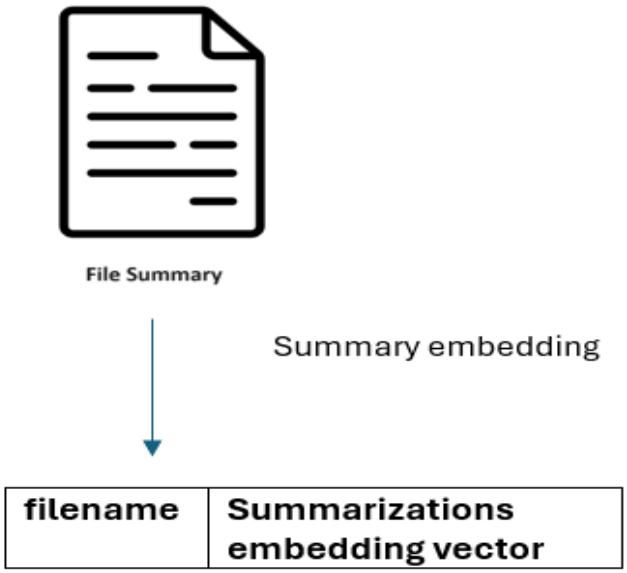

Figure 3. file summary embedding

3.4 Detect Similarity between summarizations embedding:

In this step the algorithm try to find the similarity between origin text embedding (numeric vector) and suspected file embedding (numeric vector) based on finding the cosine similarity score between these two vectors using the following formula:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Equation (1): cosine similarity equation [15]

Where:

A: is the origin text embedding vector

B: is the suspected text embedding vector

If the similarity score (which is the value of cosine similarity) is greater than 0.5 then the algorithm decides that there is suspected plagiarism between these two files, and based on that the algorithm tries to find the plagiarism in detail in the next step.

3.5 Detect plagiarism between origin and suspected file:

After detecting the initial plagiarism, the algorithm in this step tries to find the plagiarism in details, this accomplished by converting the original file and suspected file to numeric representations using text embedding techniques, as we mentioned in the previous steps we have used AraBERT model for text embedding, the embedding is done on the origin and suspected files sections, which means that the embedding is done on the abstract section for the two files , and for the introduction section , related work section , proposed method section, and conclusion section, and the references section the aim of this separation of embedding is to minimize the text

which will be as input to the model and to discover the plagiarism in all sections between the original and the suspected file.

The embedding is done on each section sentences level, this means that each sentence in one section is converted to numeric embedding and the algorithm stores the sentences and the embedding as the following:

Section1	S1	V1
Section1	S2	V2
.	.	.
Section n	Sn	V n

Figure 4. file storage embedding structure.

The origin file and suspected files are stored in the previous structure after finishing the full embedding step, then the algorithm tries to detect the plagiarism between the embedding vectors based on finding cosine similarity between these vectors, by applying equation (1) which calculates the cosine similarity score between vectors.

And if the similarity score between two vectors is more than similarity threshold (which chosen as 0.5) the algorithm flagged the two similar sentences based on the structure in figure (5).

3.6 Visualize plagiarized texts between files:

In this step the algorithm views the plagiarized sentences between origin and suspected files based on the flagged plagiarized sentences from the previous step.

IV. EVALUATION

4.1 Data set:

We have evaluated our developed system on dataset which contains abstracts of scientific papers belong to Damascus university published articles, [16] this data set contains about 1000 abstract and we have downloaded these data set and a plagiarism is done manually on this data, the plagiarism is done manually containing various types of plagiarism, full plagiarism, partially plagiarized.

4.2 Metrics:

We have used precision score for evaluation our developed algorithm, which is described in the following formula below:

Precision= $(Automatic\ plagiarism\ detection\ \cap\ manual\ plagiarism\ detection)/sentences\ count$

Where:

Automatic plagiarism detection: is the count of plagiarized sentences detected by the system.

Manual plagiarism detection: is the count of plagiarized sentences detected manually.

Sentences count: is the count of the sentences in the file.

V. Discussion

based on our evaluation and testing of our developed algorithm, we found that the precision reached to 90% , that almost all cases of plagiarism were detected by our algorithm, the remaining score of 10% is as a result of the summarization information.

VI. Conclusion And Future Work

In this research we have developed Arabic plagiarism detection system based on Arabic text summarization and Arabic word embedding techniques, first origin and suspected files are summarized, for Arabic text summarization we have used The Unified Text-to-Text Transformer (T5) model which was used as the main model for abstractive text summarization. The specific version of T5 which we used is called t5-arabic-base model which was trained on Arabic text, then generated summaries of origin and suspected files are converted to numeric representation using text transformation, the transformation is done by using a famous model which is Arabic Bert Embeddings AraBERT is an Arabic pretrained language model based on Google's BERT architecture, the tests on the developed system has Precision rate about 90%, and also is it very fast because the system will not analyzes the files which are not plagiarized in the summarization step. In the future we will add a new layer for this system which depends on machine learning algorithms to be able to detect plagiarism in all cases without lose any type or case of plagiarism.

References

[1] Kadir Yalcin, Ilyas Cicekli, Gonenc Ercan,An external plagiarism detection system based on part-of-speech (POS) tag n-grams and word embedding,Expert Systems with Applications,Volume 197,2022,116677,ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2022.116677.

[2] Eissen, S.M.z., Stein, B. (2006). Intrinsic Plagiarism Detection. In: Lalmas, M., MacFarlane, A., Rüger, S., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds) Advances in Information Retrieval. ECIR 2006. Lecture Notes in Computer Science, vol 3936. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11735106_66.

[3] Elkhidir, Mohamed & Ibrahim, Mohannad & Ibrahim, Shawgi & Awadalla, Mohamed & Khalid, Tarig. (2015). Plagiarism detection using free-text fingerprint analysis. 10.1109/WSCNIS.2015.7368306.

[4] M. Elkhidir, M. M. Ibrahim, T. A. Khalid, S. Ibrahim and M. Awadalla, "Plagiarism detection using free-text fingerprint analysis," 2015 World Symposium on Computer Networks and Information Security (WSCNIS), 2015, pp. 1-4, doi: 10.1109/WSCNIS.2015.7368306.

[5] R. A. C. Campos and F. J. Z. Martínez, "Batch source-code plagiarism detection using an algorithm for the bounded longest common subsequence problem," 2012 9th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), 2012, pp. 1-4, doi: 10.1109/ICEEE.2012.6421180.

[6] L. Bergroth, H. Hakonen and T. Raita, "A survey of longest common subsequence algorithms," Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000, 2000, pp. 39-48, doi: 10.1109/SPIRE.2000.878178.

[7] Yaqin, Ainul & Dahlan, Akhmad & Hermawan, Reno. (2019). Implementation of Algorithm Rabin-Karp for Thematic Determination of Thesis. 395-400. 10.1109/ICITISEE48480.2019.9003867.

[8] Qader, Wisam & M. Ameen, Musa & Ahmed, Bilal. (2019). An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges. 200-204. 10.1109/IEC47844.2019.8950616.

[9] Gipp, Bela & Beel, Joeran. (2010). Citation based Plagiarism detection: A new approach to identify plagiarized work language independently. HT'10 - Proceedings of the 21st ACM Conference on Hypertext and Hypermedia. 273-274. 10.1145/1810617.1810671.

[10] B. Gipp and J. Beel. Citation Proximity Analysis (CPA) - A new approach for identifying related work based on Co-Citation Analysis. Proceedings of the 12th International Conference on Scientometrics and Informetrics (ISSI'09), volume 2, pages 571--575, 2009.

[11] Omar, Khaled & Alkhatib, Bassel & Dashash, Mayssoon. (2013). The implementation of plagiarism detection system in health sciences publications in Arabic and english languages. International Review on Computers and Software. 8. 915-919.

[12] Wang, Mingye & Xie, Pan & Du, Yao & Hu, Xiaohui. (2023). T5-Based Model for Abstractive Summarization: A Semi-Supervised Learning Approach with Consistency Loss Functions. Applied Sciences. 13. 7111. 10.3390/app13127111.

[13] Fatima-zahra El-Alami, Said Ouatik El Alaoui, Noureddine En Nahnahi,Contextual semantic embeddings based on fine-tuned AraBERT model for Arabic text multi-class categorization,Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 10,Part A,2022,Pages 8422-8428,ISSN 1319-

1578,https://doi.org/10.1016/j.jksuci.
2021.02.005.

[14] Hugging Face – The AI community building the future. (n.d.). https://huggingface.co/

[15] Jiawei Han, Micheline Kamber, Jian Pei,13 - Data Mining Trends and Research Frontiers,Editor(s): Jiawei Han, Micheline Kamber, Jian Pei,In The Morgan Kaufmann Series in Data Management Systems,Data Mining (Third Edition),Morgan Kaufmann,2012,Pages 585-631,ISBN 9780123814791,(https://www.sciencedirect.com/science/article/pii/B9780123814791000137)

[16] Damascus University. (n.d.). Copyright (C) 2024 by Damascus University. https://www.damascusuniversity.edu.sy/index.
php?lang=2