Document Plagiarism Checker with Bert and Cosine-Similarity

Nilesh Tale

Prasad Shinde

Shivam Singh

Rohan Bodare
Department of Computer Engineering, Indira College of Engineering and Management,
Pune, Maharashtra, India

Prof. Shwetkranti Taware
Department of Computer Engineering, Indira College of Engineering and Management,
Pune, Maharashtra, India
May 15, 2022

Abstract Plagiarism is the practice of taking someone else's work or ideas and just copy pasting them with some slight changes or modification. Plagiarism is present in educational assignments in research work and in many more various fields. It is very difficult to check whether two documents are plagiarized or not. It becomes very difficult to check the plagiarism of documents due to limited sources of checking and verification of plagiarized document. We are unable to find whether two documents are plagiarized or not if the documents had a large volume of text or content. To overcome this problem, we propose BERT with Cosine Similarity to detect plagiarism with better accuracy than all other algorithms until now. This project focuses on computerized plagiarism detection that will provide fast and accurate plagiarism detection for the documents that contain medium to large contents or text with multilanguage support.

1 Introduction

Nowadays plagiarism becomes very serious problem in various work environments. Plagiarism not only done by students while writing assignments or journals but also by some working professionals also use the content from various sources without permission. In research work researchers uses the concept or ideas of previous research work and modifies some text by altering some words but this will still count as plagiarized work.

Internet is open-source resource from which one can easily get required content or text related to their work just on one click. They just copy paste that information and pretend like that was their work with some modifications in that information.

Plagiarism checking in documents might be easy for documents that contain 10 to 15 lines but as number of content and passages increases it becomes very difficult to check for plagiarism, hence for that purpose we require complex and detailed plagiarism checking from which we can easily analyze how much percent particular document is plagiarized with respect to other document.

It is very difficult to find plagiarized parts of text in documents especially when that document contains dense volume of content. For example, if particular teacher had to check whether submissions made by students are plagiarized or not, then teacher had to identify plagiarized parts in documents and need to remember previous parts in documents while checking next submission. This labor of checking plagiarism manually is very slow and had some gaps while checking plagiarism manually. Manual plagiarism checking is very slow for small as well as large documents that

contain more than 100 lines, therefore as number of paragraphs in documents increases the accuracy and speed of manual plagiarism checking ultimately decreases.

We need digital plagiarism checker that will do analysis and checking digitally and provide plagiarism report from which one can analyze the plagiarism between two documents. Plagiarism not only copy paste the others work but also limits the creativity of students and working professionals. Copy pasting of each other's work or ideas ultimately decrease the level of education and research work and limits the creativity.

2 Literature Survey

S. Pandey explains that plagiarism detection in large documents in manual manner is very difficult and time consuming and the plagiarism in various fields like research and in source code of any project leads to decrease quality of education and limits the creativity [1]. Digital Plagiarism detection is very useful for fast and accurate plagiarism detection. The authors propose a creative Natural Language Processing approach for intelligent plagiarism detection, but this plagiarism detection is unable to work on tables, charts, figures.

E. Pajic states that an emphasis of plagiarism detection can be performed by using feature extraction process and syntax, semantics of extracted features. In this process first of all similar features are extracted then semantics extraction is performed [2]. Tools overall have a very good classification fidelity. But execution time increases exponentially as numbers of processed documents increases.

S. Razzaq elaborates new and modern plagiarism detection tasks especially, text-based plagiarism detection and detection including monological plagiarism detection [3]. Algorithms used to detect plagiarism are based on the style of writing, but this method has disadvantages in determining plagiarism in short sentences.

H. Cheers discusses various approaches to calculate similarity between two documents and analyzed various techniques of plagiarism hiding like modification of contents and text that are not caught by these intelligent plagiarism systems [4]. But while calculating the similarity between java programming assignments the similarity score decreases ultimately. The author states that similarity count for small documents had more accuracy. But that accuracy decreases gradually when documents had large contents.

A. Rawal performed plagiarism detection on various datasets of short length sentences and analyzed the results. The author states that accuracy while comparing short documents is very high compared to dense documents [5]. The remarkable issue in the field of plagiarism checking is to check the semantics similarity between sentences because there are many general, we use to frame the sentences and this issue will raise at very high scale when number of sentences for comparison increases.

E. Gharavi expresses that there has been significant increase in the use of internet as use internet increases the plagiarism rate also increases because whole world is shifting towards digital era [6]. The author proposed an embedding based representation to detect plagiarism in documents using two level decision making. But while calculating the similarity between two documents the synonym words that are used are not taken under consideration.

H. Veisi discusses the comparison and representations of source and suspicious sentences, pair with highest similarity are considered as the candidates for plagiarism. The author performed comparison between source text and suspicious text and the sentence with highest similarity is used as scale [7]. For automated document plagiarism detection general availability of the internet and easy access to textual information enhances the detection.

In 8th paper, new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications [8].

In 9th paper, Aspect-based sentiment analysis (ABSA), which aims to identify fine-grained opinion polarity towards a specific aspect, is a challenging subtask of sentiment analysis (SA). In this paper, we construct an auxiliary sentence from the aspect and convert ABSA to a sentence-pair classification task, such as question answering (QA) and natural language inference (NLI). We fine-tune the pre-trained model from BERT and achieve new state-of-the-art results on SentiHood and SemEval-2014 Task 4 datasets.

3 Proposed Methodology

In proposed system we are going to develop a system that will help user to identify whether two documents are plagiarized or not and at last it will display the percentage of plagiarism with another document. This system will help to detect the plagiarism in various fields in educational assignments and in research field and ultimately increase the level of research and enhance the work of students. This system has multiple language support means you can check plagiarism in any language other than English. For plagiarism detection and displaying result in the form of pie chart we are using cosine similarity and Bidirectional Encoder Representations from Transformers (BERT) algorithm.

The proposed methodology for the purpose of detection of plagiarism and display the result in the form of pie chart has been depicted in the system overview given in the figure-1 above. The detailed steps utilized for this purpose

3.1 Plagiarism detection

In this step whole paragraph is divided into sentences then text classifications is carried out on the divided sentences that will extract and separate the parts of sentences into alternative words. Further, the text analysis is done using various analyzing techniques can be used according to the structure of sentences. Then after tokenization of sentences the similarity score is calculates using cosine similarity and Bidirectional Encoder Representations from Transformers (BERT) algorithm.

3.2 Cosine similarity algorithm

After tokenization whole sentence is divided into words these are treated as vector in cosine similarity. Cosine similarity is a metric, helpful in determining, how similar the data objects are irrespective of their size. We can measure the similarity between sentences or documents using Cosine Similarity. Example: -the concept of cosine similarity will be clearer by analyzing this example,

d1: the best Indian restaurant serves the best Indian
food.

d2: American restaurant serves the best American
and Indian food.

d3: Korean restaurant serves the best American,
Indian food.

	Indian	restaurant	serves	the	best	food	American	provide	Korean
d1	2	1	1	2	2	1	0	0	0
d2	1	1	1	1	0	1	2	0	0
d3	1	1	1	1	1	1	1	0	1
d4	0	1	0	2	2	1	1	1	0

Figure-2: repetition count table

The repetition count of words in four sentences as per stated in example is enlisted in above table. As we can see that word ‘Italian’ is appeared only one time in first sentence hence it is denoted by value 1. But word ‘restaurant’ appeared one time in all four sentences therefore we put value 1 under restaurant column in each sentence. Further word ‘best’ and ‘pasta’ repeated two times in first sentence so it takes value 2 by this way vectors are formed. These vectors are denoted as,

d1: [2,1,1,2,2,1,0,0,0]
d2: [1,1,1,1,0,1,2,0,0]
d3: [1,1,1,1,1,1,1,0,1]
d4: [0,1,0,2,2,1,1,1,0]

For calculating cosine similarity between first and fourth sentence the ratio of dot product of two vectors to product of magnitude of two vectors are taken. Mathematically it can be represented as,

Cosine similarity d1: d4

Cosine similarity= A. B / ||A||. ||B||

[2,1,1,2,2,1,0,0,0]. [0,1,0,2,2,1,1,1,0]

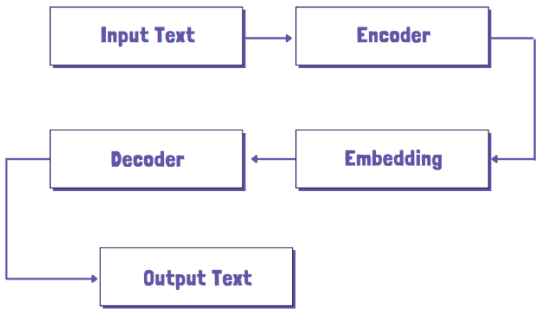
sqrt(4+1+1+4+4+1+0+0+0).sqrt(0+1+0+4+4+1+1+1+0)

Cosine similarity=10/13.4=0.74

The similarity score between first and fourth sentences is 0.74.

3.3 Bidirectional Encoder Representations from Transformers (BERT) algorithm

BERT (Bidirectional Encoder Representations from Transformers) is based on Transformers, a deep learning model in which each output element is connected to each input element, and the weightings between them are dynamically calculated based on their connection.



In BERT at first input text is encoded through Encoder that produces the input sequence for Embeddings. Then at the time of decoding, embeddings are then again decoded through Decoder and we receive the output text.

BERT is a language modeller who wears a mask. It creates embeddings for various activities using only the encoders.

The masked language model masks some tokens from the input at random, with the goal of predicting the masked word's original vocabulary id based solely on its context.

The goal of Masked Language Model (MLM) training is to hide a word in a sentence and then have the software guess which word was hidden (masked) based on the context of the hidden word. The goal of Next Sentence Prediction training is to have the software predict whether two provided sentences have a logical, sequential connection or are just random.

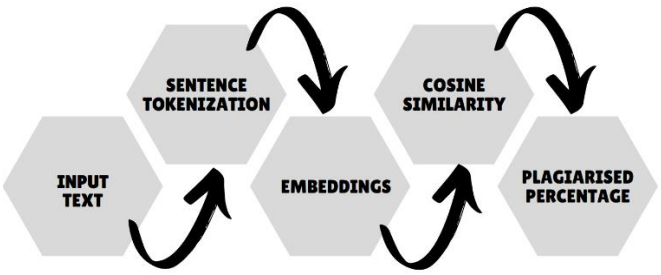
BERT is made feasible by Google's Transformers research. BERT's greater potential for recognising context and ambiguity in language is due to the transformer, which is a component of the model. Instead of processing each word individually, the transformer processes each word in connection to all other words in the phrase. The Transformer helps the BERT model to understand the whole context of a term by looking at all surrounding words, allowing it to better understand searcher intent.

BERT is first NLP technique that rely purely on the self-attention mechanism, which is enabled by the bidirectional Transformers at its core. This is essential because a word's meaning can frequently alter as a phrase progress. Each new word adds to the overall meaning of the word that the NLP algorithm is focusing on. The more words in a statement or phrase, the more unclear the emphasis word becomes. By reading bidirectionally, BERT adjusts for the augmented meaning by accounting for the effect of all other words in a phrase on the focus word and reducing the left-to-right momentum that biases words towards a given meaning as a sentence proceeds.

3.4 Applications of BERT

o Answering questions
o Summarization of the abstract
o Prediction of a sentence
o Generating conversational responses
o Determining the meaning of a word
o Inference from natural language
o Classification of emotions

3.5 BERT with Cosine-similarity:



For plagiarism checking we integrate both BERT and Cosine similarity to produce plagiarised percentage.

When we feed the text to BERT will first tokenize the sentences and create embeddings with it. Then these embeddings are given input to cosine similarity to generate the similarity score.

4 Results and Discussion

The proposed methodology for the purpose of detection of plagiarism between two documents with multilanguage support and its analysis has been presented in this research article. The technique has been obtained by using python programming language on visual studio code IDE. For development purposes, a machine with configuration consisting of an Intel Core i5 processor along with 600GB storage and 8GB RAM is used. The requirement of database is fulfilled by using MongoDB.

This technique of calculating plagiarism detection is evaluated for its performance with the help of some sample testing and their results after each sample test is enlisted in the section below.

The presented system has been efficiently tested for its accuracy using various cases. System is tested for three major cases with the accuracy score of the system means how correctly the system is able to perform the document plagiarism with respect to various input data.

Case1: Represents two documents in English language that contain almost same text.

Case 2: Represents two documents that contain text that have some differentiability.

Case 3: Represents two documents that contain text from other languages other than English, example, Hindi, Tamil, etc.

Various Cases	Accuracy
Case 1	99.9%
Case 2	87%
Case 3	82%

Figure-3: Accuracy table

The three major cases and their respective accuracy score are enlisted in table above. In first case we had taken two documents whose content is in English language almost same in that case accuracy is almost 99%. Further in case two we had tested the accuracy over two documents that contain some text with some differences and some modification its accuracy is nearly 87%. In third case we had considered two documents that contain text in other language and its accuracy 82%. After analyzing the accuracy table above for three cases and their accuracy score, we had plotted accuracy graph below representing performance of this technique in the form of accuracy.

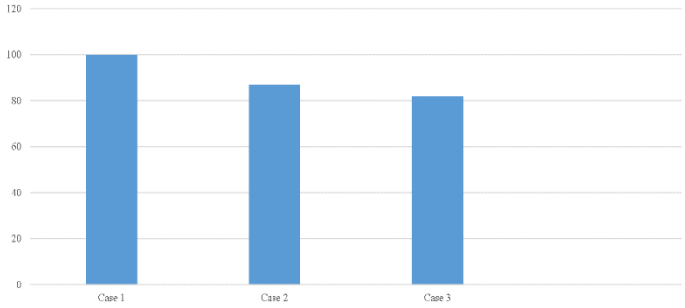


Figure-4: Accuracy graph

The values obtained as the resultant outcomes have been utilized to achieve an efficient graph in figure 4 above. The same three major cases are used to plot graph. In first case we had taken two documents in English language with almost same text whose accuracy falls under 95-100. For second case in which we had considered two documents having some modification in text yields accuracy in range 85-90. In last case we had tested accuracy for two documents whose text is in other language which gives accuracy between 80-85. The results indicate an effective and useful implementation of the approach which is highly promising outcome for the first-time implementation of such system.

Conclusion and Future work

The methodology for an effective approach for the purpose of document plagiarism detection is outlined in detail in this research article. In this digital era all documents are in digital form these documents are related to educational purpose or related research work. But these documents are presentation of your hard work and creativity it takes a lots of effort and time to study on that

topic and create a digital document that will help in further research or in future work related to that field. But now a days as internet is open source of information one can easily get others work on figure tips and they just copy paste others work which is very inappropriate practice because it limits the creativity in research work and degrades the quality of education. The manual testing of plagiarism is very hectic and almost impossible so we need digital machine for plagiarism detection. For this purpose, we designed computerized document plagiarism checker that will display plagiarism report in the form of pie chart with multilanguage support. Two copies of documents are taken in which one is source copy and other suspicious copy as input for this system. Then tokenization of paragraphs and sentences is done. After tokenization using BERT algorithm embedding is done that will create word vectors using cosine similarity the similarity score is calculated.

For the purpose of future research direction this approach can be used for checking plagiarism in various software firms, educational purpose and in research field for easier and more accurate document plagiarism detection with ease and swift.

References

[1] Pandey, Shikha, and Arpana Rawal. "An Improved NLP Approach for Detection of Plagiarism in Scientific Paper." International Journal of Applied Engineering Research" 13.9 (2018): 6733-6737.

[2] Ljubovic, Vedran, and Enil Pajic. "Plagiarism detection in computer programming using feature extraction from ultra-fine-grained repositories." IEEE Access 8 (2020): 96505-96514.

[3] Ilyas, Muhammad, et al. "Plagiarism Detection Using Natural Language Processing Techniques." Technical Journal 26.01 (2021): 90-101.

[4] Cheers, Hayden, Yuqing Lin, and Shamus P. Smith. "Academic source code plagiarism detection by measuring program behavioral similarity." IEEE Access 9 (2021): 50391-50412.

[5] El Mostafa, Hambi, and Faouzia Benabbou. "A deep learning-based technique for plagiarism detection: a comparative study." IAES International Journal of Artificial Intelligence 9.1 (2020): 81.

[6] Gharavi, E., Veisi, H., Bijari, K., & Zahirnia, K. (2016, December). A fast multi-level plagiarism detection method based on document embedding representation. In Forum for Information Retrieval Evaluation (pp. 94-108). Springer, Cham.

[7] Gharavi, Erfaneh, Kayvan Bijari, Kiarash Zahirnia, and Hadi Veisi. "A Deep Learning Approach to Persian Plagiarism Detection." FIRE (Working Notes) 34 (2016): 154-159.

[8] J Devlin, M W Chang, K Lee et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]", arXiv preprint arXiv:1810.04805, 2018.

[9] C. Sun, L. Huang and X. Qiu, "Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence", CoRR vol. abs/1903.09588, 2019.