Plagiarism Detection in Students' Answers Using FP-Growth Algorithm

Sabina Nurlybayeva[1], Iskander Akhmetov[1,3(✉)], Alexander Gelbukh[2], and Rustam Mussabayev[3]

[1] Faculty of Information Technology, Kazakh-British Technical University, Almaty, Kazakhstan
s_nurlybaeva@kbtu.kz
[2] Instituto Politecnico Nacional, CIC, Mexico City, Mexico
gelbukh@gelbukh.com
[3] Institute of Information and Computational Technologies, Pushkin Street 125, Almaty, Kazakhstan
http://kbtu.edu.kz
http://iict.kz

Abstract. According to statistics, over the past year, the quality of education has fallen due to the pandemic, and the percentage of plagiarism in the work of students has increased. Modern plagiarism detection systems work well with external plagiarism, they allow to weed out works and answers that completely copy someone else's published ideas. Using natural language processing methods, the proposed algorithm allows not only detecting plagiarism, but also correctly classifies students' responses by the amount of plagiarism. This research paper implements a two-step plagiarism detection algorithm. In the experiment, the text was converted into a vector form by the GloVe method, and then segmented by K-means and the result was obtained by the FP-Growth unsupervised learning algorithm.

Keywords: Plagiarism detection · Natural language processing · Machine learning

1 Introduction

Plagiarism is the "wrongful appropriation" and "stealing and publication" of another author's "language, thoughts, ideas, or expressions" and the representation of them as one's own original work. Plagiarism is considered academic dishonesty and a breach of journalistic ethics. The problem of plagiarism is also encountered among the writing and journalistic community, when articles copy the content, and the works have the same plots. Plagiarism has become a problem not only for publicists, but also in educational institutions, this problem is becoming more and more serious. Based on the research of 6,096 undergraduate students at 31 universities, 67.4% was found committed in plagiarism. The

I. Batyrshin et al. (Eds.): MICAI 2021, LNAI 13068, pp. 153–162, 2021.
https://doi.org/10.1007/978-3-030-89820-5_12

results of similar study on several different campuses with more than 6,000 participants from the high school and undergraduate students, showed 76% were found committed in plagiarism [9]. Thus in order to protect academic integrity plagiarism detection has gained a lot of importance these days.

Plagiarism means taking the work or ideas of someone else and passing them off as your own. The most common and well known form being textual plagiarism. For the purpose of this thesis, all references to plagiarism will be to textual plagiarism; copying the text from a source text and presenting it as your own answer. Plagiarism comes in many forms. One can directly copy a text, but detecting pure verbatim plagiarism is a fairly easy task, and plagiarists are quickly caught doing this with current tools. In order to mask an act of plagiarism, the text is often rewritten, words in a sentence rearranged, replaced with synonyms, or the text may be summarized. This makes it harder for automated systems to detect the plagiarised text. Detecting semantic meaning in a text is especially challenging to do with a computer algorithm. They are however very adept at lexical analysis. Most plagiarism detection tools use the structural and lexical similarities of documents.

Machine learning allows users to find optimal settings automatically based on statistics from a data set made up of pre-classified plagiarism and non-plagiarism cases. By defining passages, documents or sentences in a data set as plagiarism, or even the kind of plagiarism methods used on each passage, a system could potentially be tailored to each institution, or even teacher preference.

This study will develop a method for detecting plagiarism and classifying it according to the level of uniqueness of the text using natural language processing techniques. This approach implies a two-stage plagiarism detection algorithm that trains the model without marking, segmentation and searching for frequent elements.

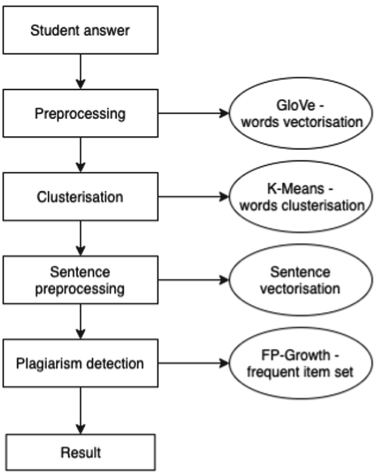


Fig. 1. Algorithm stages

The K-means algorithm is implemented for segmentation of words vectors into similar clusters, from which in the future, using the FP - Growth unsupervised algorithm, frequent sets of elements will be derived. In our context, we receive sentences that are plagiarized. Proposed plagiarism detection algorithm is described in the Fig. 1 above.

2 Related Works

With the development of the internet, people are actively looking for a solution to the plagiarism problem. The most popular tracking systems are Skyline Inc. software, Sherlock software and iThenticate [4].

Skyline. Inc. developed standalone plagiarism-detector anti plagiarism software, which detects plagiarized text. It is an autonomous Microsoft Windows-based computer desktop application made with Visual C # .Net. It follows the exact substrings detection method and it is used in the academic environment. At any rate, it has its shortcomings which is that it runs just on windows operating system, it raises a huge amount of false positives (it flags a sentence as plagiarized even though it is not). It also lacks plagiarism prevention mechanism because there is no module or subsystem in place to deter or discourage plagiarism [3].

Sherlock, is a program used to recognize copyright encroachment for essays, computer source codes files and other kinds of textual documents in digital form. Sherlock works by converting text it receives into digital signatures to measure the similarity between the documents. A digital signature is a number formed by changing several words (3 by default) in the input into a series of bits and joining those bits into a number. Sherlock is developed with C programming language and it requires compilation before being installed either on Unix/Linux or Windows. It doesn't have a GUI as it is a command-line program.

iThenticate compares a given document against the document sources available on the World Wide Web. It also compares the given document against proprietary databases of published works (including ABI/Inform, Periodical Abstracts, Business Dateline), as well as numerous electronic books and produces originality reports. The originality reports provide the amounts of materials copied (in percentages) to determine the extent of plagiarism, was developed using PHP and supported by an MSQL back end database [4].

Over the past decade, Machine Learning solutions have displaced legacy technologies. In the problem of recognizing plagiarism, three main types of solving the problem can be distinguished: distance-meausuring algorithm, character n-gram algorithm and clustering algorithm.

Previously, works such as Plagiarism Detection Using the Levenshtein Distance and Smith-Waterman Algorithm [11] where the Levenshtein distance between two strings is given by the minimum number of operations, and that needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character is seen. A commonly-used bottom-up dynamic programming algorithm for computing the Levenshtein distance involves the use of an $(n + 1) \times (m + 1)$ matrix, where n and m are

the lengths of the two strings. This algorithm is based on the Wagner-Fischer algorithm for edit a distance.

The Smith-Waterman algorithm is a classical method of comparing two strings with a view to identifying highly similar sections within them. It is widely-used in finding good near-matches, or so-called local alignments, within biological sequences.

Generally, when researchers compute the similarity of the texts, they first use the Levenshtein distance method to divide the table, after dividing the table and then making some of the portions don't calculate, they applied simplified Smith-Waterman algorithm to the rest of the table, because of the less nodes of the table will be compute than Levenshtein distance.

We also came across Intrinsic Plagiarism Detection Using Character n-gram [1]. This algorithm attempts to quantify the style variation within a document using character n-gram profiles and a style change function based on an appropriate dissimilarity measure originally proposed for author identification.

In this supervised method, the classification model is trained with a small number of features which are the proportions of the n-gram classes. In detail, method is composed of the following steps:

1. Segment each document d into fragments si by using the sliding window technique. Let S denotes the set of these fragments.
2. Build the n-gram class document model without considering numerals. Researchers choose to consider the frequency of a n-gram ngi as the number of its occurrence in d such that it is counted once per fragment. Therefore, the minimum value that could take a frequency is 1 if ngi appears only in one fragment, and its maximum value is —S— (the number of fragments in d) if ngi occurs in each fragment si S.
3. Represent each fragment si by a vector of m features fj, j 0,. . ., m–1. So that, each fj is the proportion of the n-grams that belong to the class labeled j to the total number of n-grams in si.
4. Combine into one dataset the fragment vectors obtained from all the training corpus documents. Then, label each vector with its authenticity state, i.e. plagiarized, if the fragment plagiarism percentage exceeds 50% and original otherwise.
5. Use the Naïve Bayes algorithm as classifier.

In this project, we used some basic idea from the above methods and implemented a plagiarism detection on a completely different concept. We used unsupervised machine learning algorithm called using Frequent Pattern (FP) Growth Algorithm.

3 Dataset and Features

In this study, the freely available Clough-Stevenson corpus [2] was applied. The corpus consists of answers to five short questions on a variety of topics in Computer Science field. The five short questions are:

- What is inheritance in object oriented programming?
- Explain the PageRank algorithm that is used by the Google search engine.
- Explain the Vector Space Model that is used for Information Retrieval.
- Explain Bayes Theorem from probability theory.
- What is dynamic programming?

Each question has 19 students answers and 1 Wikipedia answer. We are also given which answers are plagiarised.

Table 1. Dataset statistics

Number of questions	5
Number of answers per question	20
Number of Non or lightly plagiarised answers	57
Number of Heavily plagiarised answers	19
Mean number of words per answer	216

There are four different types of answers. They are cut, light, non-plagiarised and heavy. Cut refers to answers that are fully copy-pasted from Wikipedia answer. Heavy refers to answers that are heavily copied from Wikipedia answers, light refers to answers which are slightly copied from Wikipedia answers with extreme changes to the structure of answers. Non refers to non plagiarised work. Statistics about dataset is given in the Table 1 given above.

Thus the input for our system is various documents which contain student answers for various questions. The output is the list of students that have plagiarised work for each question.

4 Methodology

In our experiment we used K-mean algorithm and FP-Growth algorithm. The FP-Growth Algorithm, proposed by Han [10], is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree). FP-Growth algorithm is normally used to find frequent item-sets given a number of transactions. In natural language processing, we can model our text data into vectors using word vector vector representation techniques. Many word vector representation techniques have been developed in recent times. We will use GloVe [8] to represent our textual data in form of vectors. Thus we get a vector representing each unique word in our data.

For getting similar words in context and meaning, many methods have been used like Brown Clustering Algorithm [6] or Word2Vec [7]. However in our project, we use K-means [5] Clustering Algorithm because of its features like

completeness, exclusivity and fast execution. In our project, clusters of similar words are created. Then a vector for each sentence is built based on those clusters. If the two words belong to the same cluster then, the same cluster number is applied to both the words in the vector. After that a vector for each text document is created based on sentence vectors. If two sentence vectors are similar, then we assign a unique number to those sentences. Thus we get a vector for each document which we can use as item-sets in the FP-Growth Algorithm and the number of documents will be our transactions. Once we get frequent item sets we know which documents are plagiarised from each other.

5 Experiment

Preprocessing. For pre-processing we have to make sure that all the words are lower cased with all the punctuation removed. We use the NLTK sentence tokenizer to get list of sentences so that we know where does each sentence begin and end. There are various latin words in our answers as well so we use "latin1" encoding for reading the text.

Vectorisation. After preprocessing the text we needed a way to cluster similar words. Since, we were using K-means clustering which is a distance based clustering method, we needed each unique word in our corpus to have a numerical value. The choice was between several algorithms: bert glove word2 century. Since the initial task is to determine plagiarism, the word itself is important to us, but not its meaning. The BERT algorithm will generate several vectors for the words of homonyms, since it takes into account the position of the word within the sentence and considers the context of use.

The most famous word2vec word embedding model is the predictive model, that is, it trains itself trying to predict the target word in context (CBOW) or context words from the target (skip gram).

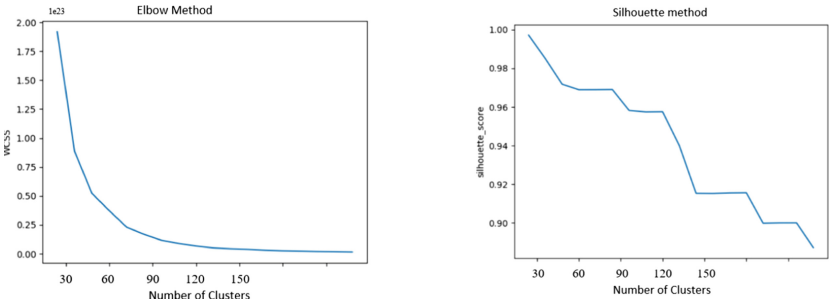
And the GloVe model uses a hit-count matrix to perform attachments that is more suited to this typical task. Each row of the matrix represents a word, while each column represents contexts in which words may appear. Matrix values represent the frequency with which a word occurs in a given context. Downsizing is then applied to this matrix to create the resulting embedding matrix (each row will be a word embedding vector). The main intuition underlying the model is the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning.

After performing GloVe on our preprocessed data we get vector of each unique word in our corpus. We can use this representation for clustering and further processing.

Implementation. K-means clustering algorithm is that it is not guaranteed to find the most optimal cluster arrangement, if you pick the wrong starting points.

One method for overcoming this is to run the algorithm a number of times with different randomly selected starting points, and then pick the solution that has the lowest total squared Euclidean distance. This approach is used in the scikit-learn package, defaulting to 10 separate repetitions. Since text-based data is usually high-dimensional and sparse, we first use the dimensionality reduction method to reduce the dimension of the high-dimensional text feature vectors. Then use the improved density peaks algorithm to determine the number of clusters and the initial clustering centers, after which the K-means algorithm is used for clustering [12].

In the experiment, two options for implementing the K-means algorithm were proposed: with the use of dimensionality reduction and without, i.e. the algorithm worked with 126 clusters, the number is determined by the number of words in the dataset—1.262 and the size reduction factor is 0.1. For an objective assessment of the method, an iterative selection of K was launched in the range from 10 to 150, and using the Silhouette and Elbow method metrics, it was revealed that the best indicator is at K = 126. Results of two metrics are shown in Fig. 2 below. Since less than 126 clusters according to the WCSS metric shows elements that are too far from the centroids, i.e., little similarity of elements within the cluster, and after 126 clusters, the resulting groups have high similarity and do not have a clear separation boundary by Silhouette metric.



(a) Elbow method (b) Silhouette method

Fig. 2. K-means algorithm metrics

Scikit-learn implementation of K-Means returns an object that indicates the cluster to which each input vector belongs. Thus after performing K-means clustering, we get similar word clusters. Now, we need to apply FP-growth Algorithm to find frequent item sets. But in order to do that, we first need to represent our data in the form of transactions of various item-sets. In order to do that we created our own vector representation explained below.

Vector Representation. From the obtained clusters and its respective labels from K-means clustering algorithm, we now use these clusters to convert our data in each documents to vectors. For this, we do the following.

- Determine the cluster label for each of the word in sentences and substitute the word with its respective cluster label.
- From the above step, we merge all the words and form sentence vectors.
- From these sentence vector, we form answer vector by assigning a unique number if two sentence vectors are different. If two sentence vectors are same we assign each of them the same number.

Thus we get a vector for each answer. This vector can be viewed as an item-set and each answer can be viewed as a transaction. Thus we get a list of transactions which contain item-sets. We can feed this into FP-Growth Algorithm to get frequent item sets.

FP-Growth Algorithm. The FP-growth algorithm is currently one of the fastest approaches to frequent item set mining. One of the currently fastest and most popular algorithms for frequent item set mining is the FP-growth algorithm. It is based on a prefix tree representation of the given database of transactions (called an FP-tree), which can save considerable amounts of memory for storing the transactions. The basic idea of the FP-growth algorithm can be described as a recursive elimination scheme: in a preprocessing step delete all items from the transactions that are not frequent individually, i.e., do not appear in a user-specified minimum number of transactions. Then select all transactions that contain the least frequent item (least frequent among those that are frequent) and delete this item from them. Recourse to process the obtained reduced (also known as projected) database, remembering that the item sets found in the recursion share the deleted item as a prefix. On return, remove the processed item also from the database of all transactions and start over, i.e., process the second frequent item etc. In these processing steps the prefix tree, which is enhanced by links between the branches, is exploited to quickly find the transactions containing a given item and also to remove this item from the transactions after it has been processed.

Thus after performing FP-Growth Algorithm we get frequent item sets. In our context we get frequent sentences which are plagiarised. We can find which sentences are plagiarised by just looking at the transactions which in our case are student answers. Now it is important to know that we need to consider only three or more frequent item-sets meaning only three or more plagiarised sentences. This is because most of the students can have one one or two similar sentences like "This is called Inheritance". We cannot penalise students for that.

6 Results

As, described in Sect. 2, Each question has 19 students answers and 1 wikipedia answer.

Having launched the proposed solution to the problem of plagiarism detection on the Corpus, we compared the metrics of the classifier with the metrics from the studied articles from Sect. 2.

Of the studies studied using machine learning, the most successful algorithm can be distinguished - it is N-gram classes algorithmBensalem2014, since the metrics of this algorithm are higher than those of Levenshtein Distance and Smith-Waterman Algorithm [11].

Table 2. Comparison of the proposed approach with the most popular algorithms

Model	Precision	Recall	F-score
K-means and FP-Growth algorithm (this work)	**90.4%**	**52.3%**	**66.3%**
Levenshtein Distance and Smith-Waterman Algorithm [11]	39.3%	31.7%	35.1%
N-gram classes [1]	31.3%	49.2%	38.3%

Thus, comparing the indicators of the algorithms from the Table 2, the solution proposed in this work has the best metrics. For example, comparing with the most successful N-gram classes algorithm, our approach has a small gap in the recall metric - only 3% but the precision metric is almost 60% higher.

7 Conclusion

Many methods have been proposed to detect and stop plagiarism. But, still there are many questions which are to be answered. Natural Language Processing has greater possibilities of providing a sound and concrete mechanism which is capable of detecting plagiarism in any document. In this paper we have tried to show finding plagiarism using FP-growth with its advantages and tried to implement clusterisation algorithm with dimension reduction. We have developed a system based on principles of vector representations.

8 Future Work

A s an extension to this work, we can include vector representations using other techniques such as word2vec. Another future work is to change the domain from documents to programming code plagiarism detection. In the task of detection plagiarism in source code, we would need stricter noise removal as well as the support count of FP-growth algorithm will increase.

Acknowledgment. This research is conducted within the framework of the grant num. AP09058174 "Development of language-independent unsupervised methods of semantic analysis of large amounts of text data".

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of the CONACYT, Mexico and grants 20211784, 20211884, and 20211178 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico.

References

1. Bensalem, I., Rosso, P., Chikhi, S.: Intrinsic plagiarism detection using N-gram classes. In: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pp. 1459–1464 (2014). https://doi.org/10.3115/v1/d14-1153
2. Clough, P., Stevenson, M.: Developing a corpus of plagiarised short answers. In: 31, pp. 527–540 (2005)
3. El Tahir Ali, A.M., Dahwa Abdulla, H.M., Snášel, V.: Overview and comparison of plagiarism detection tools. In: CEUR Workshop Proceedings, vol. 706, pp. 161–172 (2011). ISSN: 16130073
4. Foltýnek, T., et al.: Testing of support tools for plagiarism detection. Int. J. Educ. Technol. High. Educ. **17**(1), Article no. 46 (2020). https://doi.org/10.1186/s41239-020-00192-4. arXiv: 2002.04279. ISSN: 23659440
5. Li, Y., Wu, H.: A clustering method based on k-means algorithm. In: Phys. Procedia **25**, 1104–1109 (2012). https://doi.org/10.1016/j.phpro.2012.03.206. ISSN: 18753892
6. Liang, P.: Semi-supervised learning for natural language. In: Massachusetts Institute of Technology. Department of Electrical Engineering and Computer Science, p. 86 (2005). http://hdl.handle.net/1721.1/33296
7. Mikolov, T., et al.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, October 2013. arXiv: 1310.4546. ISSN: 10495258
8. Pennington, J., Richard, S., Manning, C.: GloVe: global vectors for word representation. Br. J. Neurosurg. **31**(6), 682–687 (2017). https://doi.org/10.1080/02688697.2017.1354122. ISSN: 1360046X
9. Scanlon, P.M., Neumann, D.R.: Internet plagiarism among college students. J. College Stud. Dev. **43**(3), 374–385 (2002). ISSN: 08975264
10. Shafiee, A., Karimi, M.: On the relationship between entropy and information. Phys. Essays **20**(3), 487–493 (2007). https://doi.org/10.4006/1.3153419. ISSN: 08361398
11. Su, Z., et al.: Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm. In: 3rd International Conference on Innovative Computing Information and Control, ICICIC 2008, pp. 1–3 (2008). https://doi.org/10.1109/ICICIC.2008.422
12. Sun, Y., Platoš, J.: High-dimensional text clustering by dimensionality reduction and improved density peak. In: Wireless Communications and Mobile Computing 2020 (2020). https://doi.org/10.1155/2020/8881112. ISSN: 15308677