Using Machine Learning to Detect Plagiarism in Written Works

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Wonyoung Choi

Spring, 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Rosanne Vrugtman PhD, Department of Computer Science

Daniel Graham PhD, Department of Computer Science

Using Machine Learning to Detect Plagiarism in Written Works

CS 4991 Capstone Report, 2022

Wonyoung Choi
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
wc7qpe@virginia.edu

Abstract

Ever since the inception of the world wide web, making written works more readily accessible, the potency of plagiarism has only increased. The result is difficulty determining originality in written documents. In order to combat this growing issue, I developed a machine learning solution to detect instances of plagiarism in written works. Initially, I acquired sample data composed of plagiarized works including the works that they were plagiarized on. I then used a natural language processing tool called Doc2Vec to train the machine learning model. This resulted in the creation of a machine learning model that could detect similarities between different written works. With this tool, users have the ability to determine which documents were potentially plagiarized to identify suspicious writing. This project could be improved upon by using a larger data set to detect plagiarism in a larger range of written works.

1. Introduction

According to studies, 50% of students in the United States have admitted to some form of cheating with plagiarism accounting for a large portion of that percentage [4]. With the ever-growing repository of accessible written works found online, it is easy to see that plagiarism is becoming an increasingly troubling issue. Many universities compare plagiarism to theft or misappropriation of property and have strong punishments for students caught in this act [7]. Plagiarism is particularly hard to catch today due to the large number of resources students can utilize. Developing a machine learning model to effectively identify plagiarized works is a potential solution that can help discourage acts of plagiarism.

Machine learning is a computational process that evaluates large quantities of data and creates a program that attempts to solve a particular problem [1]. The program that is created is known as a machine learning model and when trained with data, it can be used to categorize incoming data based on certain parameters. Machine learning is especially prevalent today as there is a large amount of data available that can be used to solve complex problems.

2. Related Works

In 1997 a similar application called the Turnitin was created. Turnitin is a website used to detect plagiarism in written works. The application works by comparing the newly submitted written work to materials in its database and finding similarities [5]. The tool was unable to detect plagiarism in works that plagiarized ~~off~~ materials not found in Turnitin's database. There were also cases in which Turnitin would claim a student's work was plagiarized when in reality the student used similar words but did not deliberately plagiarize.

Another popular plagiarism checker is an online tool called Unicheck, which detects evidence of plagiarism by searching through an extensive archive of 91 billion web pages [1]. The resource is trusted by 1100 academic institutions and is

seeing continued research and development to improve their detection mechanism. One way they are doing this is by implementing machine learning in order to find better ways to help students deliver original content It appears that machine learning is still an emerging idea that companies such as Unicheck are in the process of using to improve their plagiarism checkers.

In recent years, plagiarism has become a large topic especially at academic institutions. At the University of Virginia, the university invokes a single sanction expulsion to any student that plagiarizes their assignments. In 2022, this policy is being challenged to be amended with a 2-semester suspension rather than a full expulsion [4]. This event has become a considerable talking point among the university's student body. I believe that students and faculty care a lot about plagiarism and creating original content, which is why I am interested in this project.

3. Project Design

In order to train an effective machine learning model, I needed to acquire a large dataset of written papers. To accomplish this, I used an open access repository called ArXiV which was provided by an open-source machine learning platform called TensorFlow [2]. The repository contains over 200,000 scientific research papers which is more than enough to create an ample, working machine learning model.

I split the data into two different groups: the training group and the testing group. The training group would contain the bulk of the data, which would be used to train the machine learning model. The testing group would be used to validate the accuracy of the model after it had been trained.

Before the data could be used to train the model, I had to preprocess the data into a form that the training model could use. In particular, the words in the documents needed to be tokenized into a string where all the characters are lowercase and the punctuations removed. This was simple to do as all it required was to loop through the words in the document and make the necessary changes while appending them to an ongoing string. All the programming for this project was done in Python in a Jupyter notebook which is commonly used for machine learning projects.

The model was trained using a natural language processing tool called Doc2vec. Doc2vec takes each document that is passed into it and turns it into a quantifiable vector that can be compared to other vectors using math [6]. I chose to use this instead of Word2vec because I will be working primarily with whole documents rather than individual words or sentences, which Word2vec specializes in.

This was very useful in training my machine learning model as it allowed the model to compare different documents and see how they related to each other which includes noting any similarities. A similarity score is then created based on how similar the documents are which can be used to compare and group documents based on similarity. Figure 1 shows a portion of the similarity matrix returned from the model. The documents are numbered in increasing numerical order and a maximum similarity score of 1 represents an exact match.

	0	1	2	3	4	5
0	1.000000	0.089689	0.024245	0.010024	0.082042	0.151635
1	0.089689	1.000000	0.319430	0.071837	0.125479	-0.039478
2	0.024245	0.319430	1.000000	0.093307	0.172560	0.189093
3	0.010024	0.071837	0.093307	1.000000	0.152225	-0.110606
4	0.082042	0.125479	0.172560	0.152225	1.000000	0.037089
...	...	...	...	...	...	...
6435	0.113874	0.061146	0.315446	-0.101839	0.081607	0.615811
6436	0.093637	0.466034	0.299438	0.229767	0.081421	-0.031204
6437	0.194561	0.059442	-0.011432	-0.078604	-0.173878	0.146393
6438	0.107782	0.410800	0.424093	0.057956	-0.141426	0.049077
6439	0.184354	0.050756	0.036434	0.118318	-0.085642	0.212914

6440 rows × 6440 columns

Figure 1. Similarity Matrix

After training the model, I created a network graph shown in Figure 2 to more easily display the results. Each node represents a document and each connection between the nodes indicates that there is a high enough similarity score between those two documents. The high similarity criteria I decided on was a score of 0.9 or higher, which yielded the results shown in Figure 2 below:

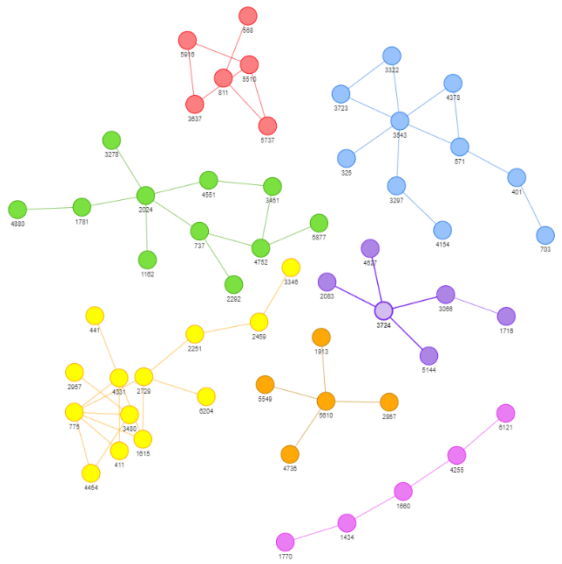

Figure 2. Similarity Network Graph

4. Results

The results of this project were promising. I was able to create a machine learning model that can detect similarities between written works. The program is able to process large amounts of written works at once and can confidently inform the user of suspicious documents. So far, the project has not seen any real-world use, as I mainly worked on this project as a learning experience. I do believe that this project can serve as an important study for future works.

5. Conclusion

Plagiarism is a known issue in academic institutions and this project demonstrates a potential solution in the form of a plagiarism detection tool that utilizes machine learning. The tool processes large amounts of data and categorizes different school reports based on similarity using Doc2Vec. This allows the user to detect two papers with high levels of similarity which can then be used as a basis for conducting further inspection to determine the legitimacy of the written work. This project has the potential to help academic institutions efficiently determine which papers are plagiarized.

6. Future Work

Currently, the biggest limitation of the tool is the amount of data it can process. The tool can currently process 6400 data points in a reasonable time, while higher amounts of data can take hours. In order to address this issue, a stronger computer must be used or the algorithm must be improved to produce a faster run time.

Another way to expand on the project is to locate a more diverse dataset of school assignments. At the moment, the project's dataset is composed of written works and does not include math assignments or programming assignments. This is due to a limitation with Doc2Vec and is worth noting for future work that may be conducted using this tool.

7. Acknowledgments

I would like to acknowledge the help of my teammates, Alex Kim and Joon Kim during the development of this project. I would also like to acknowledge my machine learning professor, Rich Nguyen, who provided the necessary knowledge required for developing this project.

References

[1] Anon. Beta test guide – unicheck help center. Retrieved March 14, 2022 from https://support.unicheck.com/hc/en-us/articles/360011066480-Beta-Test-Guide

[2] Anon. Scientific_papers : tensorflow datasets. Retrieved March 14, 2022 from https://www.tensorflow.org/datasets/catalog/scientific_papers

[3] Issam El Naqa and Martin J. Murphy. 2015. What is machine learning? - springerlink. (2015). Retrieved February 22, 2022 from https://link.springer.com/chapter/10.1007/978-3-319-18305-3_1

[4] Caroline Newman. 2022. Honor referendum: Students to vote on the future of single-sanction expulsion penalty. (February 2022). Retrieved March 14, 2022 from https://news.virginia.edu/content/honor-referendum-students-vote-future-single-sanction-expulsion-penalty?utm_source=DailyReport&utm_medium=email&utm_campaign=news

[5] S.K. Camara, Susanna Eng-Ziskin, Laura Wimberley, Katherine S. Dabbour, and Carmen M. Lee. 2016. Predicting students' intention to ... - link.springer.com. (November 2016). Retrieved February 21, 2022 from https://link.springer.com/content/pdf/10.1007/s10805-016-9269-3.pdf

[6] Susan Li. 2018. Multi-class text classification with Doc2vec & Logistic Regression. (December 2018). Retrieved March 14, 2022 from https://towardsdatascience.com/multi-class-text-classification-with-doc2vec-logistic-regression-9da9947b43f4

[7] Tshepo Batane. 2010. Turning to turnitin to fight plagiarism among ... - jstor.org. (April 2010). Retrieved February 21, 2022 from https://www.jstor.org/stable/pdfplus/jeductechsoci.13.2.1.pdf

[8] Wendy Sutherland-Smith. 2011. Crime and punishment: An analysis of university plagiarism policies. (October 2011). Retrieved February 21, 2022 from https://www.degruyter.com/document/doi/10.1515/semi.2011.067/html