Detecting machine-obfuscated content to bypass plagiarism detectors using Sentence BERT-based classifiers

Nihaal Subhash **Joshua Levitas** **Hanhee Yang**

Northwestern University

{nihaalsubash2022, jnl, hanyang2023} @u.northhwestern.edu

Abstract

Paraphrasing tools such as the SpinBot API pose a serious threat to academic integrity. It is extremely easy to copy an original paragraph and run it through a paraphrasing API to circumvent conventional plagiarism checkers. Methods have been explored to automatically classify machine-paraphrased content from human-generated content. These methods usually focus on using the entire paragraph to classify its originality. In this paper, we demonstrate it is possible to classify the content using just the first few sentences of the paragraph using BERT-based sentence embeddings. Additionally, we also explore the performance of MPNET sentence embeddings on this classification task.

1 Introduction

Unsurprisingly, plagiarism is a serious offense in the world of academia. It allows unfair and unauthorized use of someone else's work. Allowing students to plagiarize content gives them unfair advantages, diminishes their learning and development, and erodes academic integrity. Most institutions employ plagiarism detection services like Turnitin to combat this. Whilst these services are effective at identifying chunks of text copied as-is from original content, they have their limitations. A reliable method to "outsmart" these services is by simply paraphrasing the original content. It is especially easy to do this by using automated paraphrasing tools like the SpinBot API [1]. These tools make it extremely quick and easy to plagiarize content. They automatically rewrite the content by using synonyms, changing active to passive voice, etc. while retaining the original meaning of the content. In this paper, we explore methods to identify this paraphrased content.

2 Related Work

The original dataset used in this paper was developed by Foltýnek, Tomáš, et al. [2] The authors have classified both paragraphs and documents into original and obfuscated. We shall be focusing on only paragraphs in this paper. They have represented the paragraphs as vectors using embedding models. These include word embedding models like GloVe [3] and word2vec [4], sentence embeddings like USE [5], and document embeddings like PV-DBOW [6], fastText-rw [7] and fastText-sw [7]. They have experimented with three different classifiers - logistic regression, support vector machines, and naive bayes models. The peak accuracy achieved by Foltýnek, Tomáš, et al. [2] was 83.36% with the fastText-rw embedding model and an SVM classifier. These models process the entire paragraph to make a classification. With our approach, we show that one can achieve reasonable accuracy by processing just the first few sentences of the paragraphs using sentence embedding models.

3 Methodology

The easiest approach to classify the paragraphs would be to feed them into a BERT [8] model and train it for this specific downstream task. However, depending on the length of the input, this may prove to be a computationally heavy task. We argue that it is much more efficient to use the sentence embeddings of the first few sentences to achieve a reasonably high accuracy score. Paraphrasing models usually work on each sentence individually, therefore cross-attention across the words of a single sentence is sufficient, and calculating cross-attention across each word of the whole paragraph is unnecessary. We feed a varying number of sentences to the classifier models and evaluate how many sentences they need to achieve reasonable accuracy.

3.1 Dataset

We used the dataset generated by Foltýnek, Tomáš, et al. [2]. The training set consists of 4012 featured articles from English Wikipedia. Each was machine paraphrased through the SpinBot API [1] to obtain 4012 paraphrased articles. For the test set, 1990 articles are selected (and machine paraphrased as well). From these articles, paragraphs with fewer than 3 sentences were discarded. The final training set consists of 200,767 paragraphs for training - 98,282 originals and 102,485 paraphrased, and the test set, 79,970 paragraphs - 39,241 original and 40,729 paraphrased.

3.2 Sentence Embedding Model

We tested the SentenceBert [9] model for generating sentence embeddings. The sentence embeddings of the first 1 to 10 sentences of each paragraph are generated using SentenceBert, concatenated into a single vector, and then fed into the classifier model. SentenceBert generates a 384 dimensional vector for each sentence. Depending on the number of sentences chosen, the input to the classifier models can be a vector sized anywhere between 384 (1 sentence) to 3840 (10 sentences).

3.3 Classifier Model

We feed the concatenated sentence embeddings into classifier models. These are generally simple neural networks 5-6 layers deep. We have tweaked the architecture of the networks and the hyperparameters to achieve the best accuracies we could. The Sigmoid activation function is used for the final classification node. The ReLU activation function is used for every other node. The models are trained over 10 epochs. We used a batch size of 10 and a learning rate of 0.0001. Binary cross entropy is used as a loss function.

3.4 The MPNet Model

We also tested sentence embeddings based on the MPNet model [10]. The MPNet model improves upon the SOTA results of the BERT model, so we expect better results from the same. As of now, we have only tested the performance when feeding the sentence embeddings of the first three sentences. The output of the MPNet sentence embedding model is a 768 dimensional vector for each sentence. Thus, a 2304 dimensional vector is fed to the classifier model.

4 Evaluation

4.1 Human Baseline

As demonstrated by Foltýnek, Tomáš, et al.[2], human accuracy scores for this test lie anywhere between 40 to 100% for the general population. However, for experienced academics, the accuracy can reach 90% to 100%. The average accuracy for experienced academics, however, is 80%. We will treat this as the threshold for ”reasonable” or the “expert human average” in this paper.

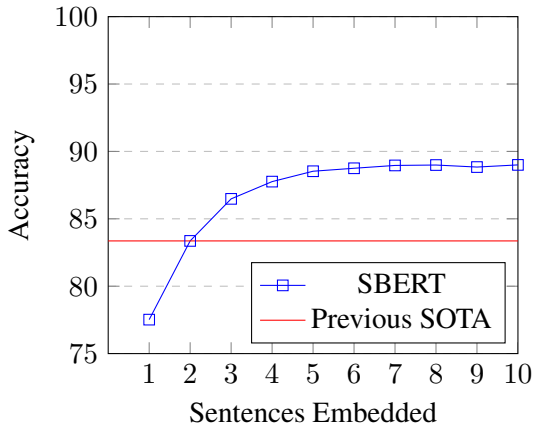
4.2 Sentence BERT Results

The results of using the SentenceBert model can be seen in Table 1 and Graph 1. We can observe that the performance scales somewhat linearly as more sentences of the paragraph are input to the classifier model. However, the accuracies are pretty close to each other. This is particularly true as 5 or more sentences are input to the model as the performance is within 1% of each other. The classifier model is able to achieve impressive results with just 3 sentences input to the model, achieving an accuracy of 86.48%. With 10 sentences input to the model, we are able to achieve a peak accuracy of 88.98%. The previous SOTA results are matched by inputting only 2 sentences into the model.

Embedding Model	**Sentences Encoded**	**Classifier Model**	**Accuracy**
SBERT	1	NN	77.52%
SBERT	2	NN	83.36%
SBERT	3	NN	86.48%
SBERT	4	NN	87.76%
SBERT	5	NN	88.53%
SBERT	6	NN	88.75%
SBERT	7	NN	88.96%
SBERT	8	NN	88.99%
SBERT	9	NN	88.84%
SBERT	10	NN	**89.00%**
MPNet	3	NN	87.64%
fastText-rw	Full ¶	SVM	83.36%*
Human Avg			80.00%

Table 1: Results

* previous SOTA results



Graph 1: SBERT Results

4.3 MPNet

The MPNet model gives impressive results with just 3 sentences input to the classifier model. The accuracy lies at 87.63% with just 3 sentences input to the model. As expected, this more complex model outperforms the equivalent Sentence BERT-based model.

5 Analysis & Conclusion

Our results are consistent with our hypothesis that it is possible to flag a paragraph as machine-obfuscated using only its first few sentences. We are able to achieve accuracies as high as 87.63% just from the first 3 sentences. Using the first 10 sentences gives us a peak accuracy of 89.00%. We are able to consistently beat the previous SOTA results by Foltýnek, Tomáš, et al. [2] while using a more efficient approach. Our model, which takes in as input only the first 2 sentences, is able to match the previous SOTA model, which processes the entire paragraph as its input. Our approach is much more efficient than feeding the entire paragraph into a classifier such as BERT [8] as cross-attention is calculated across each sentence individually. We are able to reliably surpass the average accuracy of expert humans as well. We believe that techniques like the ones proposed in our paper can be embedded as part of plagiarism detection software like Turnitin to further augment the detection rates.

6 Future Work

We would like to further explore the MPNet model for embedding sentences. We want to experiment with more state-of-the-art sentence embedding models like SimCSE [11]. We would also like to explore using other classifier models like convolutional networks to boost efficiency and/or performance. We would like to explore classifying entire documents and not just paragraphs using our approach. Our code has been made publicly available [12] to enable anyone to contribute easily.

Acknowledgements

We would like to thank David Demter for introducing us to SentenceBert and sentence embedding models during his classes at Northwestern which inspired us to explore said approaches in the paper. We are also grateful for his guidance during this project. We would like to thank Foltýnek, Tomáš, et al. [2] for making their work publicly available which allowed us to easily obtain the dataset for our work. We are grateful to the developers at Hugging Face for publicly releasing the SentenceBert [13] and MPNet-based [14] sentence embedding models through the sentence-transformers library which made our work significantly easier.

References

1. https://spinbot.com/API
2. Foltýnek, Tomáš, et al. "Detecting machine-obfuscated plagiarism." Sustainable Digital Communities: 15th International Conference, iConference 2020, Boras, Sweden, March 23–26, 2020, Proceedings. Cham: Springer International Publishing, 2020.
3. Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
4. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
5. Cer, Daniel, et al. "Universal sentence encoder." arXiv preprint arXiv:1803.11175 (2018).
6. Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." International conference on machine learning. PMLR, 2014.
7. Bojanowski, Piotr, et al. "Enriching word vectors with subword information." Transactions of the association for computational linguistics 5 (2017): 135-146.
8. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
9. Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).

10. Song, Kaitao, et al. ”Mpnet: Masked and permuted pre-training for language understanding.” Advances in Neural Information Processing Systems 33 (2020): 16857-16867.

11. Gao, Tianyu, Xingcheng Yao, and Danqi Chen. ”Simcse: Simple contrastive learning of sentence embeddings.” arXiv preprint arXiv:2104.08821 (2021).

12. https://github.com/nihaal7/SBERT_Plag_Detection

13. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

14. https://huggingface.co/sentence-transformers/all-mpnet-base-v2