Team AT at SemEval-2024 Task 8: Machine-Generated Text Detection with Semantic Embeddings

Yuchen Wei
Department of Computer Science
St. Francis Xavier University
x2020fct@stfx.ca

Abstract

This study investigates the detection of machine-generated text using several semantic embedding techniques, a critical issue in the era of advanced language models. Different methodologies were examined: GloVe embeddings, N-gram embedding models, Sentence BERT, and a concatenated embedding approach, against a fine-tuned RoBERTa baseline. The research was conducted within the framework of SemEval-2024 Task 8, encompassing tasks for binary and multi-class classification of machine-generated text.

1 Introduction

In the burgeoning field of Natural Language Processing (NLP), the distinction between human and machine-generated text is becoming an area of critical importance, particularly with the rise of advanced language models capable of producing text that closely mimics human writing. The advent of such technology poses a dual-faceted challenge: while it opens new frontiers for automation and assistance, it also necessitates robust detection mechanisms to prevent misuse and uphold information credibility. This research centers on the application of semantic embeddings to detect machine-generated text.

Semantic embeddings offer a nuanced approach to understanding and representing the meaning encapsulated within text, providing a fertile ground for discriminating between the subtleties of human and AI-authored content. This study contributes to this domain by evaluating the efficacy of various semantic embedding techniques in the context of SemEval-2024 Task 8's (Wang et al., 2024) challenges, which include the detection of machine-generated text across multiple generators and domains.

In this study, I concentrated on the application of semantic embeddings, examining and contrasting approaches such as GloVe and Sentence BERT. I developed classifiers for the task of classifying machine-generated text as part of SemEval-2024 Task 8. Specifically, my efforts were directed towards Subtask A (monolingual) and Subtask B, which involve the binary classification of machine-generated text and multi-class classification of machine-generated text, respectively.

2 Related Work

The identification and analysis of machine-generated text have become an increasingly pertinent field of study within the realm of Natural Language Processing (NLP). Previous research has primarily focused on detecting text authored by specific language models (Guo et al., 2023) or within narrow domains (Zellers et al., 2019). The latest iteration of this exploration is represented in the work by SemEval-2024 Task 8 (Wang et al., 2024), aiming at detecting text generated by a variety of models across multiple domains and languages, thus expanding the scope of investigation significantly beyond the existing literature.

Early approaches, such as those by Iyyer et al. 2014, utilized basic statistical features and machine learning models for text classification tasks, providing a foundation for subsequent research. Advancements were made by Pennington et al. 2014, who proposed a sophisticated embedding technique known as GloVe, which captures global word co-occurrence statistics (Bullinaria and Levy, 2007) to generate word representations. This technique has been widely adopted for its robustness in capturing semantic nuances.

The introduction of transformer-based (Vaswani et al., 2017) models, particularly BERT (Devlin et al., 2018) and its variants, has revolutionized the field, as demonstrated by Reimers and Gurevych 2019 with the adaptation of BERT for sentence-level embeddings (SBERT). These models have significantly outperformed traditional embeddings and

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 492–496
June 20-21, 2024 ©2024 Association for Computational Linguistics

N-gram models in various NLP tasks due to their deep contextual understanding and adaptability to different tasks and domains. Moreover, RoBERTa (Liu et al., 2019) (A Robustly Optimized BERT Pretraining Approach) refines the BERT model's training methodology to substantially improve performance across a spectrum of NLP benchmarks.

More recently, Large Language Models (LLMs) have revolutionized text generation, achieving human-like proficiency across diverse writing tasks. As LLMs like ChatGPT (Brown et al., 2020) and its successors become more adept at generating coherent and contextually relevant narratives, the importance of distinguishing between machine-generated and human-produced text grows, primarily to ensure transparency and mitigate the spread of misinformation. Consequently, developing robust detection methods for machine-generated text is crucial in maintaining the integrity of information and upholding trust in digital communications.

3 Methods

In this study, I explored four semantic embedding methods to evaluate against the fine-tuned RoBERTa baseline provided by the task coordinators (Wang et al., 2024). The methods employed encompass the GloVe (Pennington et al., 2014) embedding method, the training N-gram embedding method, Sentence BERT method, and the concatenated embedding method. In this section, I will present the methodologies applied to address Subtask A (monolingual) and Subtask B. Their primary distinction lies in the extraction of text features.

3.1 GloVe Embedding Method

Pre-trained GloVe embeddings are a set of vector representations for words that have been previously trained on large corpora, encapsulating rich semantic and syntactic relationships between words. In this approach, for each piece of text, GloVe embeddings were utilized to derive the text feature, calculated as the mean of the GloVe embeddings for each word within the text. Subsequently, a straightforward fully connected neural network, comprising several hidden layers, was constructed to perform classification.

I experimented with GloVe embeddings of varying dimensions (100d, 200d, 300d) and employed Smooth Inverse Frequency (SIF) weighted averaging as the method for averaging. This approach (Arora et al., 2017) has been demonstrated to enhance the performance of text embedding usage.

3.2 Training N-gram Embedding Method

In addition to GloVe embeddings, I explored the training of word embeddings through an N-gram neural network model. This model was designed to train a word embedding layer with the objective of predicting the subsequent word based on a given sequence of N words. Subsequently, the trained word embeddings were utilized to extract text embeddings, which then served as the basis for classification, similar to the methodology applied with GloVe embeddings.

3.3 Sentence BERT Method

Sentence BERT (Reimers and Gurevych, 2019) is a modification of the pre-trained BERT model that enhances its capabilities for generating sentence-level embeddings, facilitating more efficient and semantically meaningful comparisons between sentences. In this approach, similar to others, classification is conducted through a fully connected neural network; however, Sentence BERT is employed for the extraction of text features.

3.4 Concatenated Embedding Method

In this methodology, I concatenated word embeddings with Sentence BERT embeddings to serve as the text feature embeddings. The objective is to leverage the strengths of both approaches to enhance classification performance. The dimension of the concatenated embedding for each sample's text equals to the sum of the dimensions of the word embeddings and the SBERT embeddings. I experimented with combining GloVe and SBERT, as well as N-gram embeddings with SBERT. Ultimately, in a similar vein, a fully connected neural network was employed for inputting concatenated embeddings and performing the classification tasks.

4 Dataset and Experimental Setting

4.1 Dataset

The coordinators of SemEval-2024 Task 8 have introduced three subtasks focused on the detection of machine-generated text, encompassing multi-generator, multi-domain, and multi-lingual challenges. The first task (Subtask A) is framed as a binary classification challenge, with the goal being to differentiate between human-written and machine-generated text. Subtask A is divided into two segments: monolingual and multilingual. The monolingual segment contains 119,757 training samples, while the multilingual segment includes 172,417 training samples. In this research, my attention is solely directed towards the monolingual task, which exclusively involves texts in English. Its training set comprises 56,406 samples generated by machines and 63,351 samples authored by humans.

The second task (Subtask B) is structured as a multi-class classification challenge, wherein the labels for text samples encompass *human*, *ChatGPT*, *Cohere*, *Davinci*, *Bloomz*, and *Dolly*. This task requires classifiers to not merely determine whether a given text is machine-generated but also to identify the specific type of language model (e.g., ChatGPT (Brown et al., 2020), Cohere (Cohere Technologies, 2021)) responsible for its generation. This task encompasses a training set comprising 71,027 samples (11,997 samples for *human*, 11,995 samples for *ChatGPT*, 11,336 samples for *Cohere*, 11,999 samples for *Davinci*, 11,998 samples for *Bloomz*, 11,702 samples for *Dolly*).

The third task (Subtask C) focuses on locating the boundary within each mixed text sample. For this subtask, the provided samples are mixed texts, consisting of a human-written segment followed by a machine-generated segment. The primary objective is to identify the transition point between these two segments. This subtask includes 3,649 training samples. In my research, I did not engage with this particular subtask.

4.2 Experimental Setting

In this study, I applied my methodologies to Subtask A and Subtask B, assessing their effectiveness on the training sets using a K-fold cross-validation approach with K=5, as well as on the testing sets. Accuracy was selected as the evaluation metric for this analysis. I employed a fine-tuned RoBERTa model (Liu et al., 2019) as the baseline against which to compare my approaches. The released testing sets for Subtask A (monolingual) and Subtask B consist of 34,272 and 18,000 samples, respectively. Within the Subtask A testing set, there are 18,000 machine-generated samples and 16,272 human-written samples. For Subtask B's testing set, each label is represented by 3,000 samples.

5 Experimental Results

In this section, I will present and analyze the experimental outcomes derived from the implementation of my methodologies.

5.1 GloVe Embedding Method Results

The data in Table 1 elucidates the efficacy of the GloVe embedding methodology when applied to Subtask A (binary classification) and Subtask B (multi-class classification) of text classification. The results are segmented according to the dimensionalities of the GloVe embeddings—100, 200, and 300—and benchmarked against the performance of a fine-tuned RoBERTa model. A pattern of ascending accuracy aligns with the increase in GloVe dimensions for Subtask A, culminating in a maximum accuracy of 62.1% on the test set for the 300-dimensional GloVe model. Conversely, for Subtask B, the trend, though similar, is subdued, with the 300-dimensional model attaining an accuracy of 34.6% on the test set. The RoBERTa model, which serves as the baseline, outshines the GloVe models with a substantial margin, exhibiting peak accuracies of 73.6% on Subtask A and 48.6% on Subtask B during test evaluations.

It's clear that the dimensionality of GloVe embeddings has a direct correlation with the accuracy of the models; higher dimensions lead to more expressive embeddings and, consequently, better performance. However, despite the improvements seen with 300-dimensional embeddings, the GloVe models fall short when compared to the fine-tuned RoBERTa model.

Method Name	A K-fold	A Test	B K-fold	B Test
GloVe 100d	75.6%	59.6%	46.5%	31.3%
GloVe 200d	78.2%	61.4%	47.9%	32.9%
GloVe 300d	79.7%	62.1%	49.3%	34.6%
RoBERTa	93.8%	73.6%	63.1%	48.6%

Table 1: The experimental outcomes for the GloVe embedding method, spanning various dimensions.

5.2 Training N-gram Embedding Method Results

Table 2 presents the performance of the N-gram embedding method for both Subtask A and Subtask B, showing a progression in accuracy as the value of N increases, indicating that loner contexts inputed by N-grams contribute to more accurate models. Specifically, for Subtask A, the 2-gram model starts with a K-fold accuracy of 79.1% and a test accuracy of 60.3%, which gradually increases with the 5-gram model reaching a K-fold accuracy of 82.1% and a test accuracy of 61.4%. For Subtask B, the increase in N-gram size also correlates with a slight increase in accuracy, with the 5-gram model achieving a K-fold accuracy of 48.7% and a test accuracy of 33.9%.

When compared to the GloVe embedding method from the earlier table, the N-gram models demonstrate a competitive edge in K-fold accuracy for Subtask A, but this edge diminishes in the test results where GloVe 300d outperforms the N-gram methods. For Subtask B, the N-gram models show a similar pattern with slightly better performance compared to the GloVe 100d and 200d models but are still outperformed by the GloVe 300d and the fine-tuned RoBERTa model. RoBERTa continues to maintain a significant lead over both GloVe and N-gram methods, underscoring the effectiveness of contextualized embeddings over both static and N-gram embeddings for the tasks at hand.

Method Name	A K-fold	A Test	B K-fold	B Test
2-gram	79.1%	60.3%	47.9%	31.5%
3-gram	81.4%	61.4%	48.7%	33.3%
4-gram	80.5%	61.7%	49.3%	33.2%
5-gram	82.1%	61.4	48.7%	33.9%
RoBERTa	93.8%	73.6%	63.1%	48.6%

Table 2: The experimental results for the N-gram embedding method, across different values of N representing the number of words in the input context.

5.3 Sentence BERT Method Results

Table 3 illustrates the performance for the Sentence BERT (SBERT) method applied to Subtask A and Subtask B, with the variation in performance attributed to the different counts of hidden layers, denoted as H. The results reveal that for Subtask A, the model with one hidden layer (SBERT H=1) achieved a K-fold accuracy of 84.1% and a test accuracy of 66.3%. As the number of hidden layers increased, there was a marginal improvement in K-fold accuracy, peaking at 83.6% for four hidden layers (SBERT H=4), while the test accuracy remained relatively stable, peaking at 66.3% for one hidden layer. In Subtask B, the trend is less clear, with SBERT H=1 achieving the highest test accuracy at 38.1%, despite having a lower K-fold accuracy compared to models with more hidden layers. When compared to the GloVe method and the N-gram embedding method from previous tables, SBERT tends to offer improved test accuracy in both Subtask A and Subtask B.

Method Name	A K-fold	A Test	B K-fold	B Test
SBERT H=1	84.1%	66.3%	52.6%	38.1%
SBERT H=2	82.1%	65.6%	52.7%	37.5%
SBERT H=3	82.7%	65.8%	51.5%	36.7%
SBERT H=4	83.6%	65.8%	50.7%	37.1%
RoBERTa	93.8%	73.6%	63.1%	48.6%

Table 3: The experimental results for the Sentence BERT (SBERT) method, varying across different counts of H, which denotes the number of hidden layers.

5.4 Concatenated Embedding Method Results

Table 4 displays the experimental results for the concatenated embedding method, combining Sentence BERT (SBERT) with GloVe embeddings and a 5-gram model. The SBERT+GloVe model exhibits a K-fold accuracy of 85.4% and a test accuracy of 68.1% for Subtask A, while for Subtask B, it shows a K-fold accuracy of 53.2% and a test accuracy of 38.9%. The SBERT+5-gram model slightly outperforms the SBERT+GloVe in Subtask A with a K-fold accuracy of 86.7% and a test accuracy of 67.3%, and a K-fold accuracy of 54.1% for Subtask B, though the test accuracy is slightly lower at 38.3%. These results indicate that combining SBERT with 5-gram embeddings or GloVe embeddings could provide a marginal improvement over methods that apply each exclusively.

Method Name	A K-fold	A Test	B K-fold	B Test
SBERT+GloVe	85.4%	68.1%	53.2%	38.9%
SBERT+5-gram	86.7%	67.3%	54.1%	38.3%
RoBERTa	93.8%	73.6%	63.1%	48.6%

Table 4: The experimental results for the concatenated embedding method.

5.5 Competition Submission

Since my methods did not perform as well as the fine-tuned RoBERTa, I ultimately submitted the predictions of my fine-tuned RoBERTa model on the test sets for Subtask A and Subtask B. In the end, my predictions ranked 79th for Subtask A and 63rd for Subtask B.

6 Conclusion

In conclusion, this research has contributed to the field of detecting machine-generated text by exploring the efficacy of various semantic embedding methodologies. The results present the performance of several pre-trained semantic embeddings, like GloVe and SBERT, on the tasks of machine-generated text detection in SemEval-2024 Task 8.

Due to the superior performance of the fine-tuned RoBERTa model over all my methods I implemented, I ultimately chose to submit the prediction results obtained from RoBERTa for the SemEval competition.

7 Limitation

The limitations of this work mainly exist in two aspects. First, the methods used are too traditional and outdated. Secondly, its performance is not as good as the baseline.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39:510–526.

Cohere Technologies. 2021. Cohere natural language processing api. https://cohere.ai. Accessed: 2024-02-17.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 633–644.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.