Task-8A Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text DetectionTexts

Manojdeep Dakavaram - mada1380
Sri Venkatesha Mani Narayanan – srna7983
Sharath Soundarrajan Vanisri – shso8405

Abstract

The proliferation of large language models (LLMs) has led to a surge in machine-generated content across diverse platforms, raising concerns about potential misuse and the need for effective mitigation strategies. In this paper, we tackle the important problem of automatically identifying text generated by machines and written by humans. We implement a binary classification framework consisting of monolingual, which concentrates on English sources. Our method builds a robust text and category dataset by utilizing an enhanced M4 dataset and integrating variables like text content, model attribution, and source origin. To accurately determine the text's origin, we use supervised learning-trained GPT and BERT-based classifiers. This contribution is a useful instrument for reducing difficulties associated with the general use of LLMs.

1 Introduction

The work at hand is identifying machine-generated text automatically to address growing concerns over possible abuse of large language models (LLMs). Distinguishing between human-written and machine-generated information in monolingual (English) is the aim of this binary classification system. The task overview document highlights the necessity of automated systems to reduce potential dangers and offers a thorough grasp of the difficulties presented by the growth of machine-generated content.

Using a text and category dataset, our system uses a strong approach that includes important variables including text content, model attribution, and source provenance. We employ supervised learning-trained GPT and BERT-based classifiers to efficiently capture the

subtleties and patterns that distinguish between text created by machines and human writers. To improve its capacity for generalization, the dataset is trained on a variety of models using data from different sources.

GitHub Repo URL:
https://github.com/Manojdeep-Dakavaram/NLP_Shared_task

2 Related Work

Several recent studies have explored the challenge of distinguishing between human-generated and machine-generated text especially when it comes to large language models (LLMs) like ChatGPT. Islam et al. (2023) from the University of Asia Pacific and United International University's Department of CSE address the issues with ChatGPT's capacity to generate text that nearly resembles handwritten language. Using both supervised and reinforcement learning to fine-tune the model, their machine learning-based method achieved a 77% accuracy on a Kaggle dataset with 10,000 texts. In a comparative study titled "The Imitation Game: Detecting Human and AI-Generated Texts in the Era of ChatGPT and BARD," Hayawi et al. (year) explore the wider implications of big language models generated by artificial intelligence. In their work, a novel dataset containing texts in multiple genres—including essays, stories, poetry, and Python code—that have been authored by humans and by LLM is introduced. The authors use many machine learning models to show how well these models can distinguish between language that is written by AI and text that is created by humans. Interestingly, they draw attention to the difficulties in categorizing material produced by GPT, especially when it comes to narrative writing.

Mitrovic´ et al. (year) add to the conversation about the detection of ChatGPT-generated text by emphasizing brief online reviews. Their research examines whether it is possible to train a machine learning model to differentiate between text generated by ChatGPT that appears to be human and text that was created by an original human. They are affiliated with the Dalle Molle Institute for Artificial Intelligence and the University of Applied Sciences and Arts of Southern Switzerland. The authors examine the model's decision-making process using an explainable artificial intelligence framework, illuminating particular patterns and traits. With an accuracy of 79%, their trials, which involve customized questions and reworded evaluations produced by humans, highlight the difficulties in telling human language from that created by ChatGPT.

3 Task and Background

The primary objective of the challenge is to classify text data into two groups: machine-generated (label 0) and human-generated (label 1). Using monolingual datasets (`subtaskA_train_monolingual.jsonl} for training and `subtaskA_dev_monolingual.jsonl} for development. The Pandas library is used to manipulate the data, and the datasets are organized in JSONL format. Text samples tagged with the appropriate sources and categories make up the training dataset. For example, texts are drawn from a variety of sources, including social media and news. The algorithm divides the data into subsets according to sources and labels in order to preprocess the data. Additionally, 20,000 records are chosen at random for each label category to establish a balanced training dataset, which is then divided into training and validation sets. The resulting datasets are used for training a machine learning model to predict whether a given text is human-generated or machine-generated.

Datasets Used:

Training Dataset:
subtaskA_train_monolingual.jsonl

- Language: Monolingual
- Genre: Mixed genres (news, social media, etc.)

Dev Dataset: subtaskA_dev_monolingual.jsonl

- Language: Monolingual
- Genre: Similar to the training dataset

The resulting training dataset consists of two subsets, each focusing on one of the binary labels (0 or 1), with associated text samples and sources. The balanced training dataset ensures an equal representation of human-generated and machine-generated samples for effective model training. The code employs a random sampling mechanism and Pandas functionality for data manipulation, creating a structured dataset for training and evaluation in the subsequent machine learning tasks.

4 System Overview

4.1 Key Algorithms and Modeling Decisions:

The system leverages a combination of key algorithms and modeling decisions to effectively address the task of distinguishing between human-generated and machine-generated text. The primary modeling decisions include the utilization of a binary classification approach, where the model is trained to predict whether a given text is human-generated (label 0) or machine-generated (label 1). The system employs 3 models, the BERT-based classifier architecture, DistilBERT, and GPT-2 (Generative Pre-trained Transformer 2).

The training process involves tokenizing the input text and source information using the chosen tokenizer. The combined text and source data are then fed into the models for representation learning. A dropout layer is strategically incorporated to prevent overfitting, followed by a fully connected layer (`Linear`) to produce the final output logits. The choice of the Adamax optimizer for parameter updates ensures efficient convergence during training.

The system incorporates a supervised learning approach for model training, using the cross-entropy loss function to optimize the model's ability to correctly classify human-generated and machine-generated text. The training is executed over multiple epochs, and the model's performance is monitored on a validation set to avoid overfitting.

4.2 Resources Used Beyond Provided Training Data:

While the primary focus is on the task-specific training data (`subtaskA_train_monolingual.jsonl`), the system benefits from pre-trained language model embeddings, specifically BERT. The BERT model, pretrained on extensive textual corpora, captures rich contextual information, enhancing the model's understanding of nuanced patterns in the given text.

Additionally, the system exploits external libraries such as Hugging Face Transformers for streamlined implementation of BERT-based models and PyTorch for efficient deep learning computations. These resources contribute to the robustness and efficiency of the system by leveraging state-of-the-art pre-trained language representations and well-established deep learning frameworks.

4.3 Challenging Aspects of the Task and System Approach:

The task of binary classification, distinguishing between human-written and machine-generated text, presents several challenges exacerbated by the rapid proliferation of large language models (LLMs). The system addresses these challenges through a thoughtful design and comprehensive approach:

1. Inherent Language Mimicry:

- Challenge: LLMs like BERT can closely mimic human language patterns, making it difficult to differentiate between machine and human-generated content.

- System Approach: To tackle this challenge, the system leverages BERT's contextual embeddings, capturing intricate language nuances during training. The inclusion of source information in the combined text enhances the model's ability to discern subtle distinctions.

2. Diverse Source Types:

- Challenge: The text sources encompass diverse domains such as news, social media, and academic contexts, introducing variability in writing styles and content types.

- System Approach: The system accounts for this variability by incorporating source information during tokenization. The inclusion of source details enriches the contextual understanding, aiding in the identification of source-specific language patterns.

3. Balancing Imbalanced Datasets:

- Challenge: The task involves binary classification with imbalanced datasets, potentially leading to biased model training.

- System Approach: The system addresses this by creating a balanced training dataset, ensuring an equal representation of human-generated and machine-generated samples. This balanced approach prevents the model from favoring the majority class during training.

The system takes a multifaceted approach to address the intricate challenges posed by the task of distinguishing between human and machine-generated text. Through the integration of contextual embeddings, source information, and careful dataset balancing, the system aims to contribute to the responsible deployment of LLMs in diverse linguistic contexts.

4.4 Stages of Algorithm:

Our algorithm for binary text classification, aimed at distinguishing between human-written and machine-generated text, progresses through well-defined stages, exemplifying its robustness in diverse contexts. In the initial phase, a varied dataset encompassing both human and machine-generated samples forms the basis for model training. The preprocessing stage employs a chosen tokenizer to tokenize and encode text, enhancing it with linguistic features like part-of-speech tags and named entities. For instance, the input "The scientific paper on climate change from the journal 'Nature'" undergoes tokenization, encoding, and augmentation, resulting in a structured input format.

Transitioning to the training stage, our algorithm utilizes a custom dataset class, TextAndCategoricalDataset, organizing and processing the prepared data. The key component, BERTClassifier, DistilBERTClassifier, GPT Classifier, employs a pre-trained model to extract contextual embeddings respectively. During training epochs, the model refines its parameters using the AdamW, Adamax optimizer, minimizing

cross-entropy loss. The algorithm learns to distinguish nuances between human-written and machine-generated text.

The prediction stage tests the algorithm's generalization ability. Given an unseen text, the model processes and evaluates it, offering a classification decision—whether the text is likely human-written or machine-generated. This mirrors encountering a new scientific article on climate change. The algorithm's decision, grounded in diverse training examples, underscores its prowess in generalization and real-world classification.

Our algorithm navigates distinct stages, from preprocessing and training to prediction, showcasing adaptability and utility in discerning between human and machine-generated text across diverse contexts.

4.5 Configurations

Our investigation delves into the efficacy of diverse systems and configurations for binary text classification, specifically in the nuanced task of distinguishing between human-written and machine-generated text. Three prominent language models—BERT (Bidirectional Encoder Representations from Transformers), DistilBERT, and GPT-2 (Generative Pre-trained Transformer 2)—form the cornerstone of our exploration, each representing a unique approach to natural language processing.

BERT, renowned for its bidirectional attention mechanism and contextual embeddings, plays a foundational role in our experimentation. Its proficiency in capturing intricate linguistic relationships makes it an ideal choice for discerning nuanced differences between human and machine-generated text. Leveraging the pre-trained BERT model, we fine-tune it on our task-specific dataset, tailoring parameters to optimize performance in the binary classification task.

In tandem with BERT, we implement DistilBERT, a resource-efficient version designed for computational efficiency without sacrificing performance. By comparing DistilBERT's performance with BERT's, we aim to evaluate the trade-off between computational efficiency and classification accuracy in binary text classification.

Our exploration extends to GPT-2, a potent generative language model excelling in autoregressive language generation. Tailoring GPT-2 for our classification task, we harness its contextual understanding capabilities to discern patterns and features distinguishing between human and machine-generated text.

5 Experimental Setup

5.1 Data Split

The data split strategy plays a crucial role in the robust evaluation and generalization of machine learning models. In our experimental design, we adhere to a conventional approach, dividing the dataset into three subsets: training, development (dev), and test. The training set constitutes the majority of the data, with 40,000 records, partitioned into training and testing subsets at an 80:20 ratio, respectively. This division ensures that the model is exposed to a sufficiently large and diverse training set while retaining a separate portion for assessing its performance on unseen instances during training.

The development set, encompassing 5,000 records, serves as an intermediate evaluation stage. This subset is instrumental in fine-tuning model hyperparameters and configurations without compromising the integrity of the final test evaluation. By introducing a dedicated dev set, we aim to mitigate overfitting and tailor the model for optimal performance on the specific task at hand.

Furthermore, to maintain a fair and unbiased assessment, the test set, which is distinct from the training and dev sets, remains untouched during model development and hyperparameter tuning. This set typically comprises 20% of the original dataset and serves as the ultimate benchmark for evaluating the model's performance on completely unseen data. This separation ensures the model's ability to generalize beyond the training and development datasets, providing a reliable indication of its efficacy in real-world scenarios.

In summary, our data split methodology follows a principled approach, balancing the need for an extensive training set, an intermediary dev set for fine-tuning, and a separate test set for rigorous evaluation. This systematic division aligns with best practices in machine learning

experimentation, contributing to the reliability and reproducibility of our results.

5.2 Preprocessing and Hyperparameter Tuning

In the course of our research, we implemented a meticulous preprocessing strategy and engaged in comprehensive hyperparameter tuning, both of which played pivotal roles in shaping the performance of our system. The preprocessing phase was tailored to the specific requirements of our task, focusing on the characteristics of the input data. The dataset was organized into separate collections, distinguishing between instances labeled as 1 and 0. Each subset, denoting machine-generated and human-written text, was further stratified based on the source of the content. Similar preprocessing steps were applied to the development set, leading to distinct collections for both labels, enriched with their respective sources.

The training dataset, comprising text and source information, was subsequently formatted into two distinct DataFrames – one for each label. Moreover, the dataset was shuffled to enhance the diversity of the training process. The resulting training DataFrame consisted of text, source, and label information, with the label representing the binary classification task of human-written versus machine-generated text.

In the pursuit of optimizing the performance of our system, hyperparameter tuning emerged as a critical facet of our research methodology. The hyperparameters employed in our experiments were meticulously selected to strike a delicate balance between model complexity and generalization capability.

The fundamental hyperparameters that played a pivotal role in shaping the characteristics of our models included the number of classes (num_classes), maximum sequence length (max_length), batch size (batch_size), number of epochs (num_epochs), and learning rate (learning_rate). Each of these hyperparameters was carefully chosen to cater to the specific requirements of our binary classification task, which aimed to discern between human-written and machine-generated text.

The hyperparameters used in our experiments were not arbitrarily chosen but were carefully considered and tuned to optimize the performance of our models in the specific context of identifying human-written versus machine-generated text. The impact and significance of each hyperparameter were evaluated to ensure that our models achieved a robust and reliable classification outcome.

5.3 Libraries/Tools Used

BERT, GPT, DistilBERT from Transformers from Hugging Face, Torch, Pandas, Sklearn, OS

5.4 Summarize the evaluation measures used in the task

The evaluation process for the task involves assessing the model's performance on a development dataset (dev_dataset) using the designed evaluation measures. In the provided code, the evaluation is conducted after the model has been trained and is in the evaluation mode. The dev_dataset is prepared using the TextAndCategoricalDataset class, incorporating text, source, and label information from the development set. This dataset is then loaded into a DataLoader with a specified batch size for efficient evaluation.

The evaluation function, named evaluate, iterates through the dev_dataset in batches. For each batch, the model generates predictions based on the input texts. These predictions are compared against the actual labels, and relevant metrics such as accuracy and a classification report are computed. The classification report provides a comprehensive summary of precision, recall, and F1-score for each class in addition to the overall performance metrics.

In the subsequent code, the accuracy and classification report are printed to assess the model's performance on the development dataset. This evaluation process allows for a detailed understanding of the model's ability to distinguish between human-written and machine-generated text, providing insights into the precision and recall of the classification task. These metrics serve as valuable indicators for the model's effectiveness in addressing the specific challenges posed by the binary classification of text sources into human-written or machine-generated categories.

5.5 Results:

BERT Results:

SNO	Data Processed	Epoch	Batch Size	Learning rate	Optimizer	DEV Accuracy	F1 Score
1	20000	5	16	0.00001	AdamW	0.6178	0.42
2	20000	10	32	0.00001	AdamW	0.5966	0.34
3	40000	10	16	0.000009	AdamW	0.7708	0.80
4	40000	15	8	0.00001	AdamW	0.5541	0.28
5	40000	10	15	0.000005	Adamax	0.8268	0.83

DistilBERT Results:

SNO	Data Processed	Epoch	Batch Size	Learning rate	Optimizer	DEV Accuracy	F1 Score
1	20000	5	16	0.00001	AdamW	0.5148	0.32
2	20000	10	32	0.00001	AdamW	0.4695	0.29
3	40000	15	16	0.000009	AdamW	0.6145	0.41
4	40000	15	8	0.00001	AdamW	0.55	0.39
5	40000	15	15	0.000009	AdamW	0.7210	0.78

GPT Results:

SNO	Data Processed	Epoch	Batch Size	Learning rate	Optimizer	DEV Accuracy	F1 Score
1	20000	5	16	0.00001	AdamW	0.3745	0.22
2	20000	10	16	0.000009	AdamW	0.5461	0.29
3	40000	15	16	0.000009	AdamW	0.7120	0.75

6 Conclusion

In conclusion, the presented system leverages a aspects of BERT, DistilBERT, and GPT-2 models for the classification task of distinguishing between human-written and machine-generated text. The ensemble of these models demonstrates promising results with an accuracy of 82% (BERT), 72% (DistilBERT), 71% (GPT) showcasing the effectiveness of utilizing diverse language models. The rigorous evaluation, as indicated by precision, recall, and F1-score metrics, attests to the robustness of the proposed approach.

The achieved accuracy underscores the system's capability to discern between human and machine-generated text across various genres. However, there are potential avenues for future work. Fine-tuning hyperparameters and exploring additional pre-processing techniques may further enhance model performance.

Future iterations could explore hyperparameter tuning to enhance model robustness. Additionally, the incorporation of alternative pre-trained language models and the inclusion of additional features in the input data could further refine the system's classification capabilities. Expanding the dataset and addressing potential biases can contribute to improved generalization to diverse scenarios.

References

1. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

   URL: https://arxiv.org/abs/1810.04805

2. Reference: Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

   URL: https://arxiv.org/abs/1910.01108

3. Reference: Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

   URL: https://arxiv.org/abs/2005.14165

4. Reference: Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

   URL: https://arxiv.org/abs/1810.04805

5. Reference: Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

   URL: https://arxiv.org/abs/1910.01108

6. Reference: Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

   URL: https://arxiv.org/abs/2005.14165

7. Reference: Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized BERT approach. arXiv preprint arXiv:1907.11692.

   URL: https://arxiv.org/abs/1907.11692

8. Reference: SqueezeBERT: What can computer vision teach language understanding? arXiv preprint arXiv:2006.11316.

   URL: https://arxiv.org/abs/2006.11316