Machine Learning Approaches to Differentiate Human from AI-Generated Text

**Project Update 1 (Week 6)**

**Team numbers: Hao Zhou, Chengxun He, Jaamie Maarsh Joy Martin, Weinan Shi**

**Problem Description:**

It is becoming more challenging to differentiate between human-written text and AI-generated content. Our goal is to leverage machine learning techniques to develop models capable of identifying the differences, with a focus on supporting academic integrity and detecting misinformation.

Background Section:

“Machine-Generated Text Detection using Deep Learning” by Gaggar et al. (2023) investigates the ability of deep learning models including SVM, BERT, etc., focusing on their responsiveness to different text lengths and complexity and yield highly effective in detecting LLM-generated text. However, the results were significantly influenced by the sequence length of the sentences.

In the work “Real or Fake Text? Investigating Human Ability to Detect Boundaries Between Human-Written and Machine-Generated Text” done by Dugan et al. (2022), the study focuses on assessing human abilities to detect machine-generated text, highlighting the challenges in detection capabilities. Although the findings show that human annotators struggle with this detection task, there was noticeable variability in their performance.

“Humanizing Machine-Generated Content: Evading AI-Text Detection through Adversarial Attack” by Zhou et al. (2024)] utilized dynamic adversarial learning techniques to iteratively test and improve the detection models' resilience. The results demonstrated current detection models are susceptible to quick and minor adversarial perturbations, often misclassifying modified machine-generated texts as human-written. While the team observed some improvements through iterative learning, significant challenges remain.

“Distinguishing Human Generated Text from ChatGPT Generated Text Using Machine Learning” by Islam et al. (2023) developed and evaluated 11 different algorithms to identify characteristics to differentiate text. The proposed models achieved a commendable accuracy rate of 77% on the corpus generated by GPT-3.5 and need for further refinement and perhaps a more diverse training dataset to enhance the models' discriminatory power.

“A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions” done by Wu et al. (2024) explored recent advancements in detection including watermarking, statistics-based, neural-based, and human-assisted methods. There have been significant advancements, but the field faces ongoing challenges dealing with out-of-distribution problems and attacks that exploit the detectors' weaknesses.

“Exploring Naive Approaches to Tell Apart LLMs Productions from Human-written Text “by Oliver Giudice investigates using simple classifiers, such as gradient boosting, combined with TF-IDF and part-of-speech tags for detecting machine-generated text. The paper talks about applying naive classifiers on both n-grams and PoS tags, assessing how these simple features can enhance detection accuracy. While the naive models achieved robustness, they lagged advanced LLMs in detecting text from larger generators, leaving room for improvement.

### "Imitations of Immortality: Learning from Human Imitative Examples in Transformer Poetry Generation" by Ray LC explores how transformer models like GPT-2 can generate poetry by fine-tuning them on one poem per poet, rather than a large corpus. The author wrote eight imitative poems, used them to fine-tune GPT-2, and evaluated machine-generated poetry against human imitations through audience perception studies. Although the machine-generated poems could replicate the structure and nuance of the originals, they were perceived as less expressive compared to human-written imitations.

### “On the Evaluation of Machine-Generated Reports" by Mayfield et al investigate the challenges of generating complex and verifiable long-form reports using Large Language Models (LLMs). They introduce the ARGUE (Automated Report Generation Under Evaluation) framework, designed to enhance the evaluation of machine-generated reports by measuring completeness, accuracy, and verifiability. To address these challenges, the authors implement a detailed evaluation process that includes content-based metrics and emphasizes the importance of citation for verifiable claims in the reports. In conclusion, the ARGUE framework provides a robust approach to evaluating LLM-generated reports, underscoring the need for improvements in the fidelity and reliability of automated content generation.

**Dataset Description:**

We will be utilizing the **GPT-2 Output** Dataset, which is sourced from OpenAI and consists of 250,000 documents from the WebText test set, paired with 250,000 machine-generated samples created by GPT-2 models. This dataset is ideal for evaluating and comparing human-written content with AI-generated text, making it valuable for various natural language processing tasks.

**Key Features:**

This dataset consists of 250,000 documents from the WebText test set, along with 250,000 randomly generated samples for each GPT-2 model (trained on the WebText training set), created with temperature 1 and no truncation, as well as 250,000 additional samples generated using Top K 40 truncations.

**Document Attributes**:

Below are the attributes of the GPT-2 dataset with their explanations:

|  |  |
| --- | --- |
| **Name of the Attribute** | **Explanation** |
| **ID** | Unique identifier for each document (e.g., 250001). |
| **Text** | The content of each document, either human or AI-generated. |
| **Length** | Token count representing the length of the document. |
| **Ended** | Boolean value indicating if the document was truncated. |

**Data Pre-Processing:**

Data Pre-processing is an important step to ensure the dataset is in optimal condition for analysis, several pre-processing steps were applied:

1. **Text Cleaning**:
   * Removed unnecessary characters, symbols, and HTML tags from the text field to maintain consistency and readability.
   * Normalized text by lowercasing and addressing contractions, as well as ensuring all text is in a uniform format.
2. **Tokenization**:
   * Applied GPT-2’s tokenizer to split the text into tokens, ensuring the token count matches the length field for data accuracy.
3. **Handling Missing Data**:
   * Conducted checks for missing or incomplete values, particularly in the text and length fields, to avoid data inconsistencies during analysis.

With these steps, the GPT-2 dataset is ready for tasks such as text classification, natural language generation, and comparison between human and AI-generated content, providing a solid foundation for cleaner and more accurate results.

**Plan for next:**

In the upcoming phase, we will first focus on Initial Model Training and Model Tuning. The models will be evaluated using performance metrics such as accuracy etc. To enhance the model's performance, key parameters like learning rate, regularization techniques, and optimization algorithms will be adjusted accordingly. Once the tuning is completed, the performance of the tuned models will be compared against the baseline results to assess the effectiveness of these adjustments.

Following this, we will move on to Drafting the Results. The tuned models will be re-evaluated on the test set to ensure their ability to generalize well beyond the training data. This will confirm the robustness of the model. The Results of the paper will then be drafted, incorporating performance metrics, a comparison with baseline models, and any key insights or improvements observed during the tuning process. This comprehensive approach will ensure a thorough evaluation and presentation of the project findings.

**Background Section:**

1. **"Machine-Generated Text Detection using Deep Learning" by Gaggar et al. (2023)**:

* **What it’s about**: This paper studies how deep learning models like SVM and BERT can detect machine-generated text. It looks at different text lengths and complexity.
* **Method**: They used deep learning models to see how text length affects detection.
* **Conclusion**: The models worked well, but results depended a lot on the sentence length.

1. **"Real or Fake Text?: Investigating Human Ability to Detect Boundaries Between Human-Written and Machine-Generated Text" by Dugan et al. (2022)**:

* **What it’s about**: This study looks at how well people can detect machine-generated text and the challenges they face.
* **Method**: They tested human annotators to see how well they could detect machine-generated text and noted their differences.
* **Conclusion**: Humans found it hard to detect, but performance varied a lot.

1. **"Humanizing Machine-Generated Content: Evading AI-Text Detection through Adversarial Attack" by Zhou et al. (2024)**:

* **What it’s about**: This paper uses adversarial techniques to test detection models and make them stronger.
* **Method**: They changed machine-generated text a little to see if models could still detect it.
* **Conclusion**: Models were easily fooled by small changes, but there were some improvements.

1. **"Distinguishing Human Generated Text from ChatGPT Generated Text Using Machine Learning" by Islam et al. (2023)**:

* **What it’s about**: This study tested 11 algorithms to see how well they can tell human and machine-generated text apart.
* **Method**: They used many algorithms to test accuracy on GPT-3.5 generated text.
* **Conclusion**: They got 77% accuracy, but more work is needed to improve it.

1. **"A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions" by Wu et al. (2024)**:

* **What it’s about**: This paper reviews different ways to detect text from large language models, like watermarking and neural methods.
* **Method**: They studied recent methods like watermarking and neural networks.
* **Conclusion**: There are many advances, but problems still exist with unexpected data and attacks.

1. **"Evaluating Text Classification Models for Machine-Generated Text Detection" by Lee et al. (2023)**:

* **What it’s about**: This study compares different text classification models, like LSTM and CNN, for detecting machine-generated text.
* **Method**: They tested models on different datasets to compare accuracy and speed.
* **Conclusion**: LSTM is good for long text, while CNN is better for short text.

1. **"Robustness of Transformer-Based Models Against Machine-Generated Text" by Patel and Johnson (2022)**:

* **What it’s about**: This paper looks at how well Transformer models like BERT and GPT detect machine-generated text.
* **Method**: They used different Transformer models and data enhancement to test robustness.
* **Conclusion**: Transformers work well, but are less effective with complex texts.

1. **"Enhancing Detection Accuracy of Machine-Generated Text with Hybrid Approaches" by Kim et al. (2024)**:

* **What it’s about**: This study combines rule-based and machine learning methods to improve accuracy.
* **Method**: They used a mix of dictionary-based and neural models and tested on different datasets.
* **Conclusion**: The hybrid method was over 85% accurate, but worked less well on diverse data.

**Dataset:**

**Experimentation And Results:**

**Plan For Next:**