NLP Final Project

Chang Chen 12212739
Yuankun Tian 12213031
Shan Zhang 12213052

June 7, 2025

Abstract

We use given datasets to fine-tuned BERT model and train a supervised classifier, which classifies if a text is generated by human or machine. Then we used Fast-DetectGPT, as a representation of zero-shot method, to do the same classification mission and compared their performances.

1 Introduction

The LLM-fine-tuning-based supervised classifier is expected to perform well on the training dataset but has a lower robustness across different datasets. The reason is because the feature it extracts from the training dataset does not have to persist on the ones generated by other LLMs or in other categories. In contrast, some zero-shot methods should have better robustness across source models and categories, since their extracted features are concluded artificially and have been proved to be effective across datasets.

2 Review

2.1 LLM-fine-tuning-based Supervised Classifier

Bert-base-uncased is used as the base LLM in the fine-tuning task. The bert-base-uncased is a pretrained BERT (Bidirectional Encoder Representations from Transformers) model released by Google in 2018. As a base-sized variant, it contains 12 Transformer encoder layers with 768 hidden dimensions (110M parameters total). The "uncased" designation indicates its text preprocessing converts all input to lowercase and removes accent markers. Its key features include bidirectional context encoding through self-attention mechanisms, pretraining via masked language modeling (MLM) and next sentence prediction (NSP) tasks, and standard BERT input embeddings (token/segment/position). Typical applications span general NLP tasks like text classification, named entity recognition, and sentence similarity analysis. With a 512-token length limit, it offers better computational efficiency than larger variants (e.g., bert-large) for GPU-constrained environments. We load it via Hugging Face's Transformers library (BertTokenizer/BertModel.from_pretrained('bert-base-uncased')).

We implemented a BERT-based classifier to distinguish between human-written and LLM-generated Chinese text. The pipeline consisted of:

- Data Preparation: Collected Chinese text samples from two sources - human-written articles (from JSON files in ./zh_unicode) and machine-generated outputs (Qwen model outputs in ./zh_qwen2). Texts were labeled (0=human, 1=LLM) and randomly shuffled.
- Model Architecture: Initialized a bert-base-chinese model with sequence classification head (2 output classes) using Hugging Face Transformers. The model was configured with label mappings (id2label/label2id) for interpretability.
- Preprocessing: Tokenized texts using BERT's Chinese tokenizer with max_length=512, padding and truncation for uniform input size.
- Training: Split data 80/20 for train/validation. Fine-tuned for 3 epochs with: learning_rate=2e-5, batch_size=8, weight_decay=0.01. Used epoch-wise evaluation and saved best model.

- Saved final model.
- Evaluate the final model on the validation set and obtain its accuracy, precision, recall, F1 score, and ROC AUC.

2.2 Fast-DetectGPT

Fast-DetectGPT, introduced by Bao et al.[**?**], detects human and machine generated texts by measuring their conditional probability curvature. It is based on the assumption that human and machine tend to choose different tokens during text generation process. More specifically, machine should tend to choose the token with a higher conditional probability $p_\theta(\tilde{x}_i|x_{<i})$ at the i-th position, given the preceding texts $x_{<i}$. Given a passage x and a model p_θ, the conditional probability function is defined as

$$p_\theta(\tilde{x}|x) = \prod_j p_\theta(\tilde{x}_j|x_{<j}),$$

where the tokens $\tilde{x}_j$ are independently predicted given x. As a special case, $p_\theta(x|x)$ equals to $p_\theta(x)$. Given a passage x, a sampling model p_φ and a scoring model p_θ, the conditional probability curvature is defined as

$$d\left(x, p_\theta, q_\varphi\right) = \frac{\log p_\theta(x \mid x) - \tilde{\mu}}{\tilde{\sigma}}$$

where

$$\tilde{\mu} = E_{\tilde{x}\sim q_\varphi(\tilde{x}|x)}\left[\log p_\theta(\tilde{x} \mid x)\right] \quad \text{and} \quad \tilde{\sigma}^2 = E_{\tilde{x}\sim q_\varphi(\tilde{x}|x)}\left[\left(\log p_\theta(\tilde{x} \mid x) - \tilde{\mu}\right)^2\right].$$

Then Fast-DetectGPT compare the curvature with a threshold ϵ. If the curvature is larger than ϵ then it is classified as machine generated, vice versa.

Algorithm 1 Fast-DetectGPT Machine-Generated Text Detection Algorithm

Require: Text passage x, sampling model q_φ, scoring model p_θ, decision threshold ϵ
Ensure: `True` (machine-generated) or `False` (human-written)

function FastDetectGPT(x, q_φ, p_θ)
  $\tilde{x}_i \sim q_\varphi(\tilde{x} \mid x), \quad i \in [1..N]$ ▷ Conditional sampling
  $\tilde{\mu} \leftarrow \frac{1}{N}\sum_i \log p_\theta(\tilde{x}_i \mid x)$ ▷ Estimate mean
  $\tilde{\sigma}^2 \leftarrow \frac{1}{N-1}\sum_i \left(\log p_\theta(\tilde{x}_i \mid x) - \tilde{\mu}\right)^2$ ▷ Estimate variance
  $\hat{d}_x \leftarrow \left(\log p_\theta(x) - \tilde{\mu}\right)/\tilde{\sigma}$ ▷ Compute curvature
  return $\hat{d}_x > \epsilon$
end function

3 Experiment Setting

3.1 Dataset

We used Ghostbuster-data and FACE2 dataset as English and Chinese datasets.

Ghostbuster-data is an English dataset provided by an essay, which also provides another supervised method. The dataset includes essay, reuter, wp, perturb, perturb_old and other, totally 6 categories. In our experiment, the other category is ignored for convenience. The perturb and perturb_old include different methods to perturb the original text generated text, mocking the realistic case when artificially reparagraphing machine-generated text to escape AI detection. Those perturbation methods includes changing token capitalization, rewriting, words replacement, adding or removing individual letter and so on, each containing 200 samples.

FACE2 dataset is a Chinese dataset provided by another essay, which provides FACE as a zero-shot method. It contains news, webnovel, wiki, totally three categories. Its machine samples is generated by Qwen2.

3.2 Training Hyperparameters

- **Data Preparation**:
  - Dataset shuffled and randomly split with 8:2 ratio (Training : Validation)
- **Training Configuration**:
  - Batch size: 8
  - Training epochs: 3
  - Learning rate: 2×10^{-5}
  - Optimizer: AdamW
  - Loss function: Binary Cross Entropy
- **Model Architecture**:
  - Base model: `bert-base-chinese`
  - Framework: BERT-base-uncased architecture
  - Output: Binary classification (human vs LLM-generated)
- **Hardware**:
  - GPU: L40
- **Tokenizer**:
  - Truncation and padding enabled
  - Maximum sequence length: 512

4 Results

4.1 Across languages

The average performance of different methods on different language datasets is shown in the table1 below. Zero-shot method accuracy, precision, and recall are acquired by maximizing f1 score. Fast-DetectGPT uses Qwen2.5-0.5B as the reference model and scoring model. The results show that

Table 1: Performance Comparison of Text Detection Methods

Method	**English (gpt_essay)**	**Chinese (qwen2_news)**
Supervised (Bert-base)	Accuracy: 0.9950 Precision: 0.9900 Recall: 1.0000 F1 Score: 0.9950 AUROC: 0.9684	Accuracy: 0.7925 Precision: 0.7142 Recall: 0.9719 F1 Score: 0.8233 AUROC: 0.9273
Fast-DetectGPT	Accuracy: 0.9502 Precision: 0.9534 Recall: 0.9467 F1 Score: 0.9500 AUROC: 0.9893	Accuracy: 0.8805 Precision: 0.8840 Recall: 0.8760 F1 Score: 0.8800 AUROC: 0.9490

Zero-shot method, Fast-DetectGPT in this case, has a clearly much better robustness across different languages compared to the supervised classifier.

Table 2: Model ROC AUC Comparison (English)

	essay	reuter	wp	perturb	perturb_old
gpt	1	0.9864	0.9954	0.5614	0.595
claude	0.9984	0.9319	0.8984		

4.2 Supervised Classifier across Categories

For English ghostbuster dataset, we trained a model based on the GPT-generated essay dataset. The ROC AUC of the model on the training dataset and other datasets is shown in Table2.

It shows that the model performs quite poorly on the two perturbed datasets.

For Chinese FACE2 dataset, we trained two models. One is based on Qwen2-generated news texts, while the other is based on Qwen2-generated Wiki texts. ROC AUC of two models is presented in Table3 and Table4 respectively. The results shows that supervised classifier has a limited robustness across different categories.

Table 3: Classifier ROC AUC (news text based)

	news	webnovel	wiki
qwen	0.9386	0.9365	0.7760

Table 4: Classifier ROC AUC (Wiki text based)

	news	webnovel	wiki
qwen	0.6847	0.6959	0.873

If we train the model based on news text, then its performance on Wiki texts would be quite low. If we train it based on Wiki texts, then its performance on news and webnovel would be poor. This shows the limitation of this supervised method.

4.3 Fast-DetectGPT across Categories

Although on the original machine generated texts Fast-DetectGPT's ROC AUC are all quite high, it performs poor on perturbed texts, with a ROC AUC around 0.522. It seems perturbation on texts has an effect of escaping detection both with supervised or unsupervised methods in this experiment.

5 Conclusion

First, by comparison of the average performances, including accuracy, precision, recall, f1 and ROC AUC, of BERT-fine-tuning classifier and Fast-DetectGPT, we conclude that Fast-DetectGPT has a better robustness across English and Chinese than the former. Second, by observation during experiments, we find that supervised classifier performs much faster than Fast-DetectGPT. Third, perturbation on the text can easily make these two methods lose effect.