Detect human or machine text

Tudor Chițu, Alexandru-Cristian Ingeaua and Florin Brad*
University of Bucharest, Romania
*Bitdefender, Romania

tudorchitu11@gmail.com, ingeaua.alexandru@gmail.com, fbrad@bitdefender.com

1. Introduction

Finetuning a BERT model to distinguish between **machine-generated** and **human-written texts**, using the **M4 dataset**. The goal is to train a binary classifier capable of identifying subtle patterns that differentiate AI-generated content from authentic human language.
Some of the **key challenges** include:

- Ensuring the model doesn't **overfit to specific generation styles** or models present in the dataset.
- Evaluating generalization to **unseen models or prompts**.
- Performing **feature analysis** to understand domain-specific cues and address them

2. Dataset and Preprocessing

Dataset Description:
We used the **M4 dataset**, a large-scale benchmark for detecting machine-generated text across multiple **domains** (news, reviews, stories, Wikipedia), **generators** (including ChatGPT, Cohere, Dolly-v2, and GPT3.5 (*text-davinci-003*)).. The dataset includes a wide variety of human- and machine-written texts, designed to reflect real-world detection challenges and cover a broad spectrum of generation styles.

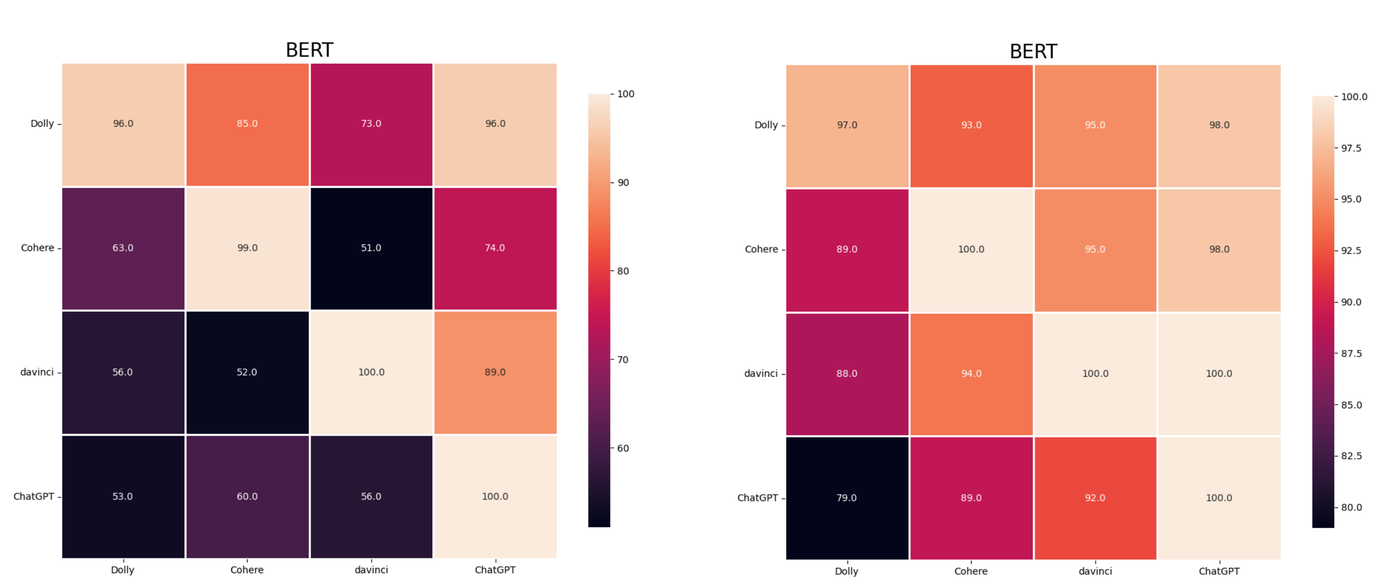
Data Preprocessing:
We applied the following preprocessing steps using the `bert-base-uncased` tokenizer from HuggingFace:

- Each input text was tokenized using: `tokenizer(batch["text"], padding="max_length", truncation=True, max_length=512)`.
- Texts were lowercased and tokenized into WordPiece tokens.
- Sequences were **truncated** to a maximum length of 512 tokens and **padded** to fixed length to ensure uniform input size.
- Special tokens such as `[CLS]` and `[SEP]` were automatically added by the tokenizer.

3. Approach and results

We **fine-tuned a pre-trained BERT model** on a binary classification task. Specifically, we utilized the *bert-base-uncased model* from the Hugging Face Transformers library (*BertForSequenceClassification*).

Firstly, we conducted experiments on two distinct domains: **arXiv** and **Wikipedia**, using a **cross-generator evaluation** strategy: training on one generator and testing across five.



Left: arXiv domain. Right: Wikipedia domain. Cross-generator classification accuracy heatmaps.

Key Observations:

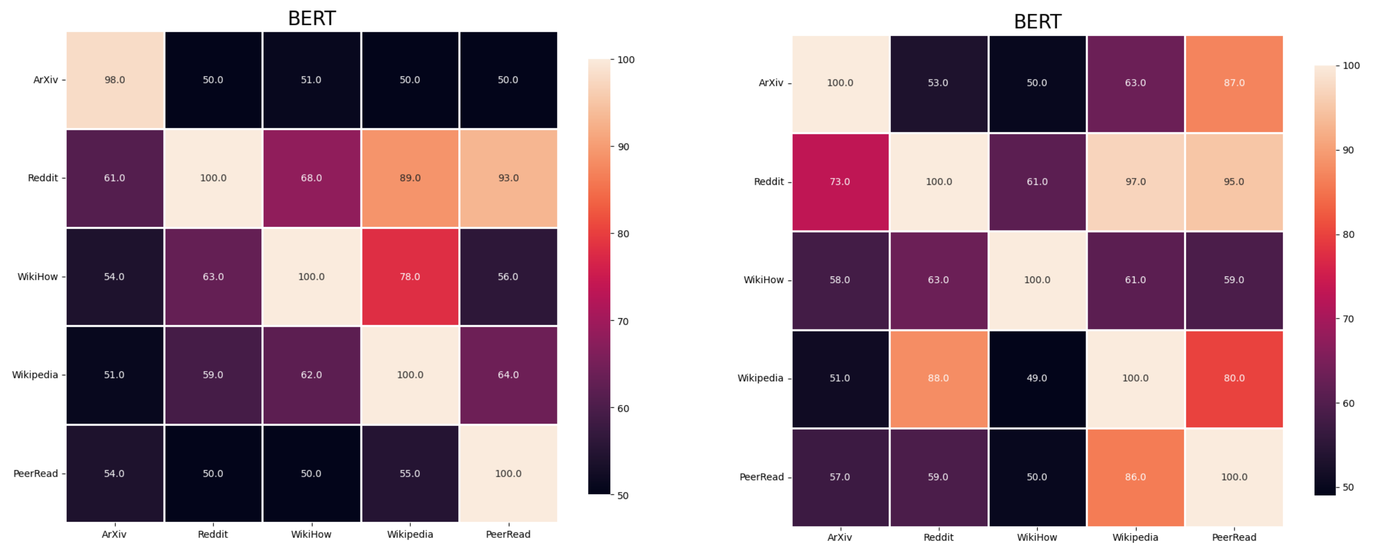
arXiv Domain

- High within-generator accuracy.
- Dolly-trained models generalized well (e.g., to Cohere: 85%, ChatGPT: 96%).
- Davinci- and ChatGPT-trained models had weak transfer performance.

Wikipedia Domain

- Very high accuracy across the board, both within and cross-generator.
- All models had near-perfect diagonal values (97%).
- Strong generalization from all models.
- Indicates BERT detects generator patterns better in informal domains.

Secondly, we conducted experiments on two distinct generators: **davinci** and **ChatGPT**, using a **cross-domain evaluation** strategy: training on one domain and testing across five.



Left: davinci generator. Right: ChatGPT generator. Cross-domain classification accuracy heatmaps.

Key Observations:

davinci generator

- Strong within-domain detection across all sources.
- Reddit-trained models generalize relatively well to other domains.
- ArXiv-trained models show poor cross-domain transfer.
- WikiHow→Wikipedia shows moderate transfer ability.
- PeerRead demonstrates limited generalization to other domains.

chatGPT generator

- Maintained strong within-domain detection.
- Reddit-trained models show superior cross-domain transfer.
- ArXiv→PeerRead transfer significantly improved compared to davinci.
- Wikipedia→PeerRead shows strong improvement compared to davinci.
- Overall cross-domain detection improved across most domain pairs.