Aman Behera
Chemical Engineering, Sophomore
Indian Institute of Technology (IIT), Roorkee

January 7, 2024

TOPIC- — Project Report

1. AI-Generated Text Detection using BERT

AI-Generated Text detection using BERT(Bi-directional Encoder Representation Transformer) is from the family of LLMs, which has been used for the classification of human-authored texts and AI-generated texts. It uses various feature selection techniques in the core for categorizing the texts generated by the two groups. It detects the two based on semantic differences, usage of vocabulary, statistical distributions, and sentiment analysis measures. This detection method comes in the family of Black-Box Detection Algorithms for AI-Text Detection.

1.1. Table of Contents

Following is the list of contents in this report of mine

2. Introduction

AI-generated content is becoming increasingly sophisticated, making it challenging to distinguish between genuine and computer-generated text. Fraud emails and Fake news are becoming every day's stories. My project aims to tackle this issue by leveraging the power of BERT (Bidirectional Encoder Representations from Transformers) to identify and flag AI-generated text segments. This would allow the AI to advance at a much faster pace but not at the expense of human security and fundamental integrity.

3. Work Flow

The solution follows a comprehensive approach to detect AI-generated texts

1. Data Preprocessing: We clean and preprocess the textual data, removing the noise, stop words, punctuations and non-alphabetical words. This was done using Bert-Preprocess.
2. Additional Datasets I collected various datasets that were hosted by various competitions like the one in analysis, concatenated them, and used the collective data to train my model. This increased my training instances from 1378 to around 50k. Allowing my model to train on a large set of varying data allowing effective feature identification and implementation.
3. Model Training: Using a BERT-based sequence classification model, we train the system to distinguish between genuine and AI-generated text with a high degree of accuracy. Here I used bert-base-uncase, one may also use bert-large-cased but that may significantly increase the training time as it has 24 encoder layers instead of 12 in the case of the base version, which would reduce the submission score
4. Predictions: Once trained, the model generates predictions for test data, highlighting potential AI-generated content segments. The predictions have been made on the test data.
5. Result Analysis: The results are then saved in a CSV file, which is then submitted to the contest.

4. Comprehensive explanation on how BERT detects AI-generated texts

An extensive analysis was done by me on how BERT detects AI-generated texts and the low-lying features Bert captures to segregate the texts. For this task, I read various research papers on blogs to understand various aspects in and out of the range of the problem statement, allowing me to get a better grip on the underlying mechanism responsible. I have put all the research papers and the blogs I used for this purpose in the folder.

4.1. The analysis of mine answers various questions like:

1. What is black-box and white-box detection? | And which one of the both is the problem statement?
2. What are watermark embeddings and how are they used by white-box detectors?
3. Why detecting generated texts is important?
4. What is LLM hallucinations ? | And how can it be prevented?
5. What feature does Bert use to identify b/w the generated texts and human-authored texts?
6. How we do structure the pipeline of solving this problem?
7. What are the possible edge cases and the explanations for them?

The analysis which can be accessed from here

5. Youtube Video Explanation

I did a complete video explaining the analysis I had done, as words speak more than text. It would give a better understanding of how I structured the pipeline, and what exactly was my thought process during action. The videos can be accessed, from my GitHub which hosts the links to the videos

6. Edge Cases

A few edge cases like the one above and possible explanations of them have been included in the Edge-Cases folder hosted in my GitHub repository. It points out a few of the underlying assumptions and details that have not been provided in the problem statement. Suitable logical arguments regarding the same have also been attached.

7. Tips Follow Through

Below are some tips I would like to share:

1. Using Bert-large instead of Bert-base may cause time limit errors although the depth of analysis would be greater,
2. Do make sure while running the notebook you have the exact path of the Kaggle datasets, or else it will throw a submission error.
3. Although the additional datasets don't require their exact paths to be present, you can just download them locally and then upload them to start the analysis.
4. Using Bert Pre-Process instead of the usual NLP pipeline of preprocessing using NLTK would fetch better results because it's better trained.
5. We can train on even more data for better analysis, keeping in mind that the time-limit was 9hrs and my model took around 2.5hrs suggests we can play with more datasets

7.1. Variations and expanding the depth of analysis:

1. One may do fine-tuning for optimizing various hyper-parameters involved, like the number of epochs, optimizer being used, batch size, number of layers, and varying the activation function of the Bert layer.
2. I did a few of them and here is what I found, decreasing the batch size increased the accuracy in the offline run but the submission accuracy in the contest somehow decreased.
3. Adam did a far better job than compared to RMSE, but one may experiment with SGD as it generalizes well at the expense of accuracy.

8. Notable Points

A good paper on the topic of fake content detection as I had mentioned is this. Some points on differences between LLM vs Human-generated content from the above paper:

1. If words are ranked in terms of usage, human-written essays tend to have more rare words in terms of rank than LLM-generated essays which tend to have high frequency/probability words. Human essays are more diverse in vocabulary but shorter in length.
2. LLM-generated content is less emotional than human-generated essays - LLMs tend to use more punctuation and grammar to convey feelings.
3. LLMs are prone to repetitiveness.
4. LLM essays are more formal and structured. Human essays frequently use question marks, exclamation marks, ellipses, etc to express emotions.
5. LLM-generated essays could be more fabricated and could be less factual and may not always be accurate.

6. LLMs tend to concentrate on common patterns in the texts they were trained on leading to low perplexity scores.
7. LLMs use more determiners, conjunctions, auxiliary relations, etc in terms of Part of Speech tag analysis. There is the dominance of nouns implying argumentativeness and objectivity.
8. There is significantly fewer negative emotions and more neutrality in LLM-generated content.
9. LLMs can hallucinate at times to produce irrelevant, artificial, or non-sensical content.
10. Linguistic analyses show that machines produce sentences with more complex syntactic structures while human essays tend to be lexically more complex.

All these points along with many other are mentioned in the analysis which I did (refer to the PDF I attached in the analysis section)

9. CV vs LB scores fallacy & Strange Findings:

Many people, including me, have having 0.99+ CV score. I guess the reason is we are using a dataset different from the hidden test dataset. Maybe, the hidden test dataset was generated by prompting like: "Write an essay like a human writer, you must randomly add some typos in your essay to pretend like you are a human, not an AI. But the datasets we are using were prompted without like a human."

9.1. My take on why fallacy is occurring:

1. In a normal scenario, AI is almost unable to output typos, and this would be a very strong discriminator.
2. This competition's host made effort to hide easy wins from us - which is great. Hence, they did a special post-processing of the ai-generated essays, to introduce human-like noise (like typos).
3. The Key factor to win this competition will be the ability to build a training dataset resembling the one used by competition hosts (Hence extremely high scores can only be attained by the use of a dataset that is similar to the one being used to test our scores in leaderboard (which isn't the right thing to do as we are trying to overfit our model on the hidden test data which is being used in LB, so I feel extremely high scoring models won't be working that well on other datasets).
4. Test data has data that is highly grammatically incorrect, due to which word embeddings are meaningless as if the word doesn't have correct spelling then word embeddings would be highly skewed. And the motive of providing raw and clean data to our transformers would be lost.
5. Because hidden test data has errors like the errors we see in the train.csv, there is a reason why differences in scores are being produced in CV and LB scores.

10. How to improve LB score | Results

My submission using voting classifier of LogisticRegression(lr) and SGDClassifier(sgd) with basic preprocessing and tf-idf vectorizer, it easily scored a 0.932 LB score while consuming a runtime of mere 33 seconds!. So it's all about simple thinking

Well In my opinion what I understand and learned from the contest, is that it's fine to have a model with LB scores of around 0.85 if we would try to achieve high scores it simply isn't possible by the usage of deeper models like BERT, simpler models by the usage of TF-IDF and voting classifier seem to score better (like in my case, the score of 0.932 was achieved using tdidf), also we can reverse engineer the original essays upon which data preprocessing of some sort was applied and noises were introduced to make it seem more human-like, we can do this by using certain libraries of python (like language tool python).

But it would eventually kill the motive of having a good model in general for detection as we would be more focused on overfitting our model towards the hidden test data whose internals aren't even disclosed, it would be like hitting an arrow into the black box.

11. Result summary

1. A reliable CV-LB relationship has been very elusive in this competition. In spite of best efforts, the test datasets created could not mimic the public or private hidden test data
2. TFIDF features like max df, min df, and max features do not seem to make much difference owing to the nature of this use case. Similarly, 'modified Huber' is the only loss function that does well owing to outliers.
3. LLMs / Transformers are not good at detecting fake text when the test data is corrupted to the extent word embeddings layer becomes meaningless. Even otherwise, owing to their weights oriented towards other use cases, their efficiency does not seem to be good
4. TFIDF Vectorizer-based models are best at the moment for fake detection. This could be because phrases and sentences need to be compared to distinguish the style of LLM vs human.
5. Typos are major features that distinguish between human and LLM essays. However, since errors have been introduced into training and test datasets, this is no longer a useful feature.
6. NLP-based cleaning could be important as a few features due to character differences can make a huge difference in detecting fake texts.
7. TFIDF Vectorizers perform very differently with different tokenizers and tokenizing patterns. Scores vary significantly with different patterns - arriving at the right pattern could be crucial
8. Introducing typos into the training dataset is another way to model the problem and increase the score
9. Ensembling increases the score (only if the model accuracy of each model is greater than 50
10. Feature-based models are not found to be efficient for this use case even though some recent papers claim to have used XGBoost with features like frequent words, distinguishing words, etc to differentiate the essays

12. Author

Aman Behera, IIT Roorkee - if u have any suggestions do let me know!

Submitted by Aman Behera on 7 de enero de 2024.