

End Evaluation Report

Detect AI Generated Text

Siddhant Rohila

January 2024

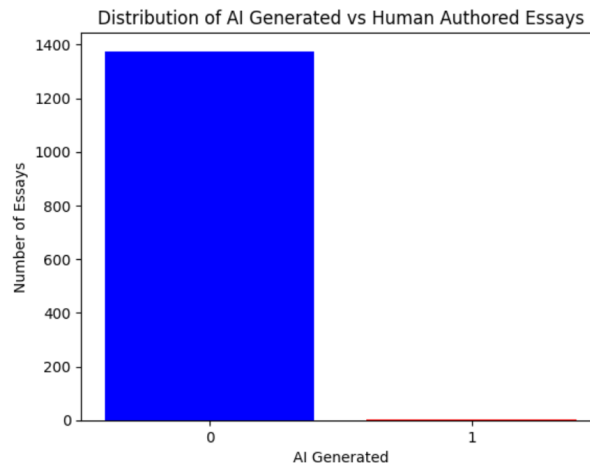
1 Introduction

The objective of this project is to develop a machine learning model capable of detecting whether an essay was written by a student or was developed by a large language model. The approach involves utilizing a Byte-Pair Encoding tokenizer from the Hugging Face tokenizer library for pre-processing, feature extraction with TF-IDF(term frequency-inverse document frequency), ensemble model for classification which combines logistic regression and stochastic gradient descent classifier.

2 Methodology

2.1 Data Collection

The dataset used for training and evaluation consists of a diverse set of text samples, including both human-generated and AI-generated text. The original training dataset is quite uneven, following bar chart is an evidence for that.



I've imported an external train dataset 'daigt-v2-train-dataset' using the pandas library, it provides a good distribution of essays for the model training since it has generated : student written ratio of 45.6 : 54.4.

2.2 Pre-Processing

For the natural language processing tasks, the given essays cannot be used directly, they must be changed into suitable format including a sequence of subword units. Using the hugging face tokenizer library, a Byte-Pair encoding tokenizer is employed to break down text into meaningful subword units. This involves normalization and pre-tokenization and further, training the tokenizer. The training process involves iterating through the dataset in chunks of 1000 samples.

2.3 Feature Extraction

Term Frequency-Inverse Document Frequency (TF-IDF) vectorization is a technique used for representing text data in a numerical format and capturing the significance of terms in the context of each document. Now since the test data is tokenized, it is ready for TF-IDF vectorization.

$$w_{x,y} = \text{tf}_{x,y} \times \log \left(\frac{N}{df_x} \right)$$

TF-IDF

Term x within document y $\text{tf}_{x,y}$ = frequency of x in y
 df_x = number of documents containing x
 N = total number of documents

Following are the tasks implemented in order to achieve this:

- Creating a vectorizer object involving certain parameters like ngram_range, lowercase (for preserving the case of the text), token_pattern etc.
- The vectorizer is then fitted to the tokenized test data and then the vocabulary learnt from the test data is obtained.
- Memory cleanup and resource management using garbage collector.

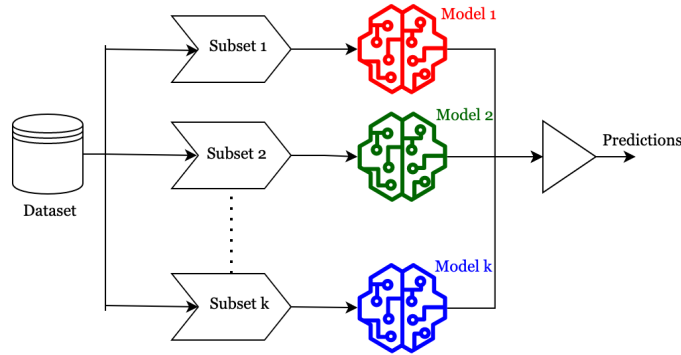
2.4 Model Training

This part is basically implemented by training of an ensemble model, i.e. multiple diverse models are used to predict the outcome. The ensemble consists of a Logistic Regression model and a Stochastic Gradient Descent (SGD) Classifier. It is obvious that the given data requires a binary classification. There are multiple algorithms like logistic regression, support vector machines, naive bayes, K nearest neighbours and many more. Finally I have used the following for model training :

- Logistic Regression : A process of modeling the probability of a discrete outcome given an input variable. The logistic regression model uses the sigmoid function (also known as the logistic function) to map the output to a value between 0 and 1. The formula for the sigmoid function is:

$$S(z) = \frac{1}{(1 + e^{-z})}$$

- Stochastic Gradient Descent Classifier : SGD is an optimization algorithm used to minimize the cost or loss function during the training of a machine learning model. It is a variant of the gradient descent algorithm. It is suitable for both convex and non-convex optimization problems and thus I employed it in my ensemble model.
- The ensemble model is trained using the fit method with the TF-IDF transformed training data and corresponding labels.
- Memory cleanup and resource management using a garbage collector.



3 Challenges Faced and Results

My kaggle submission has shown an accuracy of 93.6%. The final submission.csv looks like this :


submission.csv (97 B)

id	generated
0000aaaa	0.3898678890763593
1111bbbb	0.3898678890763593
2222cccc	0.3898678890763593

There were several challenges that I faced during making this machine learning model :

- I am a beginner in this field of machine learning and data science, it took me a bit of time to grasp some natural language processing concepts. Initially I did not find it intuitive, but after several days of consistent efforts, things became clear.
- Skewed Dataset : The dataset given on kaggle had a biased distribution and hence the training dataset was not appropriate to train the model. I explored several other datasets and found 'daigt-v2-train-dataset' that had an unbiased distribution of the essays.
- Training the Model : There were several model training methods like SVMs,KNN,logistic regression and many more. Since I was using an ensembling model, there were several combinations I tried which was a vigorous task but ultimately the one I used came out to be fruitful.

Here is a display of my kaggle submission, along with the accuracy.

Submission and Description		Public Score 
 <u>VLG_Project_LLM-DAIGText - Version 7</u>		0.936
Succeeded · 34m ago · Notebook VLG_Project_LLM-DAIGText Version 7		

In conclusion, this project has been an exciting journey of discovery and innovation. As we move forward, let's carry the momentum of our achievements and continue to push the boundaries of knowledge in this field.