Introduction To Text Mining And NLP (INF582)
Human Written Text vs. AI Generated Text

École Polytechnique

February 2023

1 Description of the Challenge

The goal of this project is to study and apply machine learning/artificial intelligence techniques to *predict whether a paragraph is written by human or generated by an AI*.

AI generative text is the process of creating natural language texts using artificial intelligence models. These models are trained on large amounts of text data and learn to generate coherent and fluent texts based on some input, such as a prompt, a keyword, or a context. AI generative text has many applications in various domains, such as content creation, summarization, dialogue systems, and natural language understanding. Lately, with the unprecedented advance of Deep Learning, some models are being taught to generate an excellent text that at times we cannot differentiate from the humanly written ones. However, this also raises ethical and social concerns about the potential misuse and abuse of such technologies. For example, AI-generated texts could be used to spread misinformation, manipulate opinions, impersonate others or create fake content. Therefore, it is important to develop and implement responsible and trustworthy practices for using and evaluating AI generative text systems. Given a labeled dataset with both versions, will you be able to extract the best features that differ between the two styles?

The pipeline that is typically followed to deal with the problem is similar to the one applied in any classification problem; the goal is to use information from the labeled text to learn the parameters of a classifier and then to use the classifier to predict whether a text in the test set is generated by an AI or written by human.

The challenge is hosted on Kaggle, a platform for predictive modelling on which companies, organizations and researchers post their data, and statisticians and data miners from all over the world compete to produce the best models. The challenge is available at the following link: `https://www.kaggle.com/c/inf582-2023`. To participate in the challenge, use the following link: `https://www.kaggle.com/t/babf7ea0930e40b584e5c83cdb1e0b40`.

2 Dataset Description

As mentioned above, you will evaluate your methods on a list of paragraphs of various subjects. The dataset contains various subjects description (written by humans) in to addition to machine generated descriptions using transformer based pretrained models (GPT, BART, T5, etc..). You are given the following files (which are available at: `https://1drv.ms/u/s!AhcBGHWGY2mukcpnlTBVIKshtAAtMQ?e=WE4aMz`).

1. **train_set.json**: This file contains $4,000$ paragraphs for various subjects (in the field `text` of the json file) of the json file) and labels (in the field `label` of the json file). The dataset is divided as follows: $2,016$ human written text and $1,984$ text generated from different text generation models.

2. **test_set.json**: This file contains $4,000$ text in total, divided as follows: $2,020$ human written text and $1,980$ text generated using the same models used in the `train_set`. This dataset is distributed equally between the public and private leaderboards on kaggle.

3 Evaluation

The performance of your models will be assessed using the accuracy metric. Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where $TP =$ True Positives, $TN =$ True Negatives, $FP =$ False Positives, and $FN =$ False Negatives.

4 Provided Source Code

You are given a script written in Python that will help you get started with the challenge. The script (`logistic_regression_baseline.py`) uses `TF-IDF` with a `Logistic Regression` classifier to make predictions.
As part of this challenge, you are asked to write your own code and build your own models to predict whether a text is generated or written by a human.

5 Useful Python Libraries

In this section, we briefly discuss some tools that can be useful in the challenge and you are encouraged to use.

- A very powerful machine learning library in Python: `scikit-learn`[1].
- A very popular deep learning library in Python is `PyTorch`[2]. The library provides a simple and user-friendly interface to build and train deep learning models.
- Since you will also deal with textual data, the Natural Language Toolkit (`NLTK`)[3] of Python can also be found useful.
- `Gensim`[4] is a Python library for unsupervised topic modeling and natural language processing, using modern statistical machine learning. The library provides all the necessary tools for learning word and document embeddings.
- `Hugging Face`[5] an immensely popular Python library providing pretrained models that are extraordinarily useful for a variety of natural language processing (NLP) tasks.

[1] `http://scikit-learn.org/`
[2] `https://pytorch.org/`
[3] `http://www.nltk.org/`
[4] `https://radimrehurek.com/gensim/`
[5] `https://huggingface.co/`

6 Rules and Details about the Submission of the Project

Rules. The following rules apply to this challenge: (i) one account is allowed per participant (ii) there is a limit in the size of each team (at most 3 members), (iii) privately sharing code outside of teams is not permitted, (iv) there is a limit in the number of submissions per day (at most 4 entries per day), (v) the use of external data is **not allowed** (except from word embeddings, e.g. BERT, GPT, BART, WordVec, etc..). For instance, you are not allowed to use external data to determine if a summary is generated by a machine or written by a human. (vi) your code must be **reproducible**.

Evaluation and Submission. Each team must fill this form before **14/03/2023**.
Note: without filling this form you won't be able to submit your files.
Your final evaluation for the project will be based on (1) the presentation you will give (**40**%), (2) on your position on the private leader-board and the accuracy that will be achieved (**30**%), and (3) on your total approach to the problem and the quality of the report (**30**%). As part of the project, you have to submit the following:

- A 4-5 pages report, in which you should describe the approach and the methods that you used in the project. Since this is a real classification task, we are interested to know how you dealt with each part of the pipeline, e.g., how you created your representation, which features did you use, which classification algorithms did you use and why, the performance of your methods (loss, accuracy and training time), approaches that finally didn't work but are interesting, and in general, whatever you think that is interesting to report.
- A directory with the code of your implementation (not the data, just the code).
- Create a `.zip` file containing the code and the report and submit it to Moodle.
- **Deadline: 19/03/2023 23:59**

Presentation: As mentioned above, you will be asked to present the approach you followed. Therefore, you will need to prepare some slides (using ppt or any other tool you like).
Date of presentation: TBA