Introduction to NLP - Final project report

Authors : Tania ADMANE and Giovanni MANCHE

Professor : Yair Lakretz

Table of contents

I. Introduction

The goal of this project is to train a NLP classifier capable of distinguishing whether a given economic text (article, report, speech, etc.) originates from a real economic figure or instance such as the ECB, the FED, economists and academic papers, or from an AI-generated source. This project lies at the intersection of language analysis and economic reasoning, and directly relates to current concerns regarding disinformation and the authenticity of sources. It will also be a good example of usage of NLP on a concrete subject.

To do so, we first scrap raw text data from various sources, then we use an AI-model to generate texts about roughly the same topics. Sources are in English and French. A simple, *baseline* model is first trained and tested, before implementing more sophisticated neural network models that answer the main challenges our topic arise. The results we obtain are quite mixed, as we faced several computational problems (to generate and scrap data as well as to train powerful models). Nevertheless, we try to explore these different issues and how to resolve them if possible.

II. Data

Firstly, we scrapped raw text data from several real economic sources using RSS feeds. These sources are, for instance, ECB press releases, VoxEU columns, CNBC articles, and French economic newspapers such as LeMonde – Economie. To ensure relevance and quality, articles are filtered by keywords (if applicable) and a minimum word count threshold. The content is cleaned by removing irrelevant elements (scripts, ads, footers, etc.) as much as possible, and metadata such as title, source, date, and language is stored. The data is then saved into a structured JSON database for further processing and modeling.

To generate the texts using AI, we relied on the **Groq API**. We produced texts in both French and English, each limited to approximately 300 words for performance reasons. We first generated a broad range of economic topics, and then, using a simple prompt, we asked the AI to produce texts based on these topics.

We created two types of content: economic speeches and economic articles. Below are the prompts we used:

- *"Professional economic speech of 280 words about: {topic}. Natural oral style, no formatting."*
- *"Economic article of 280 words about: {topic}. Natural journalistic style, no formatting."*

We quickly were limited by different generation and scrapping problems : the sources were limited, often not working without some sort of subscription, and when it worked, the “cleaning” work was immense, as the raw text often contained elements like ads, asks for subscription, etc. As for the AI generation, the computation takes a rather long time as well, as our laptops are not efficient enough for such tasks to be quicker. That’s why we have a rather small dataset, which can be problematic in terms of results, but at least it is a unique dataset we created from scratch.

III. Baseline model (SVM, RF, Logistic Regression)

To begin our analysis, we decided to apply classical supervised classification methods: Support Vector Machine (SVM), Random Forest, and Logistic Regression.

Before building the models, we performed text data preprocessing. We first removed stopwords, those very frequent words (such as *the*, *of*, or *and*) that provide little discriminative information.

We then built features likely to reveal stylistic differences between texts generated by AI and those written by humans. Among these features were sentence structure : AI-generated texts often tend to have more regular sentences, the variety and use of punctuation, the frequency of logical connectors such as *therefore*, *nevertheless*, or *however*, as well as broader measures such as text entropy.

Double TF-IDF Vectorization

We then applied a **double TF-IDF (Term Frequency – Inverse Document Frequency) vectorization**, a classical method for transforming text into a numerical vector. The TF-IDF principle is based on two components. First, the term frequency (TF) measures how often a word appears in a document relative to the total number of words. Then, the inverse document frequency (IDF) weights this score according to the rarity of the word in the entire corpus. Thus, a word that is very frequent in all documents, such as *the*, will have a low weight, while a rarer word will be given greater importance.

Our approach combined two levels of analysis. The first vectorization, performed by words, aimed to capture vocabulary and characteristic expressions. AI-generated texts, for example, often contain stereotypical phrases such as *therefore, we can observe that*. These expressions stand out with a high TF (frequently used in AI texts) and a medium IDF because they appear in many AI-generated documents. The second vectorization, performed by characters, allowed us to detect finer orthographic and stylistic patterns. It focuses on character sequences (or character n-grams) present in words. For example *nseq* in *consequently* or *verth* in *nevertheless*.

Model Evaluation

Once the texts were vectorized, we tested the three machine learning algorithms while optimizing their hyperparameters using **GridSearchCV**. The results were identical for all three methods: each achieved an **accuracy of 100%,** even when keeping only the most relevant features. This clearly indicated significant overfitting. Our dataset contained only 600 texts, which is far too small to ensure generalization. The models were therefore most likely memorizing the training examples rather than being able to generalize to new data.

These limitations led us to consider more robust approaches. In the next phase of our work, we decided to turn to more complex models such as **BERT** and its derivatives.

IV. BERT Models

Bidirectional Encoder Representations from Transformers (BERT) is a neural network model specifically designed for NLP tasks. It was introduced by Google in 2018 in the paper “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”[1]. Here’s

[1] This paper itself heavily relies on a previous paper, also from the Google research team, that introduced transformers : “Attention is all you need” (2017).

a breakdown of how BERT works (details can be found in the 2017 and 2018 papers, which are open access) :

- Bidirectional context : BERT reads text in “both directions”, meaning that it looks at previous and next words in a sentence, making it good at understanding context. In our case, this is indispensable as AI might have full sentence patterns and economic subjects can be hard to understand without the whole picture.
- Transformer architecture : BERT uses the Transformer architecture and the mechanism of “self-attention”, which allows the model to weigh the importance of different words in a sentence relative to each other (these models are called **contextual models).** For instance, the word “bank” can refer to a financial institution or the side of a river depending on the context. With older models like GloVe, we wouldn’t capture these nuances, which is, however, indispensable for our work.
- Pre-training tasks : BERT is already pre-trained on two unsupervised tasks, the first being “Masked Language Model” (some words are masked, and the model has to predict them) and the second being “Next Sentence Prediction” (the model learns to understand the relationship between pairs of sentences).
- Fine-tuning : After pre-training, the model can be fine-tuned on specific tasks. For classification tasks like ours, BERT uses a special [CLS] token (classification token) that is prepended to every input sequence. During pre-training, this token learns to aggregate information from the entire sequence. For fine-tuning, we add a feedforward classifier on top of the [CLS] token's final hidden state, which serves as a fixed-size representation of the entire input text. This approach is particularly well-suited for our binary classification task (human vs AI-generated economic texts), as the [CLS] token captures the global semantic and stylistic features of the entire document.
- Input Representation : BERT takes as input a sequence of tokens, which are first converted to token embeddings, position embeddings (to retain word order), and segment embeddings (to distinguish between sentence pairs). These embeddings are summed and passed through multiple Transformer layers.

Because of these capabilities and advanced mechanism, we argue that this model should provide a significant improvement from the baseline model. We however faced several challenges, and we will now go into greater detail about the implementations we proposed.

IV.1. Standard (base) BERT Implementation

Firstly, we implemented the `bert-base-uncased` with 110 million parameters, consisting of 12 transformer layers, 768 hidden units, and 12 attention heads. The classification head we added consists of a dropout layer of 0.1 (meaning that 10% of the neurons are “muted” at each learning step) and a linear layer that maps BERT’s [CLS] token representation to 2 output classes, human VS AI-generated. We employed a standard fine-tuning approach where all BERT parameters are trainable. We trained the model for 3 epochs (it’s like retraining the model 3 times with it remembering what it learned) and a batch size of 16. We used the classical loss function for classification (CrossEntropyLoss).

Our model achieved 100% accuracy on both validation and test sets, which indicates severe overfitting and the presence of exploitable biases in our dataset. This near-perfect performance suggests the model is likely learning superficial patterns rather than genuine linguistic differences between human and AI-generated economic texts.

IV.2. Bias detection

We investigated deeper our dataset and here are the main results :

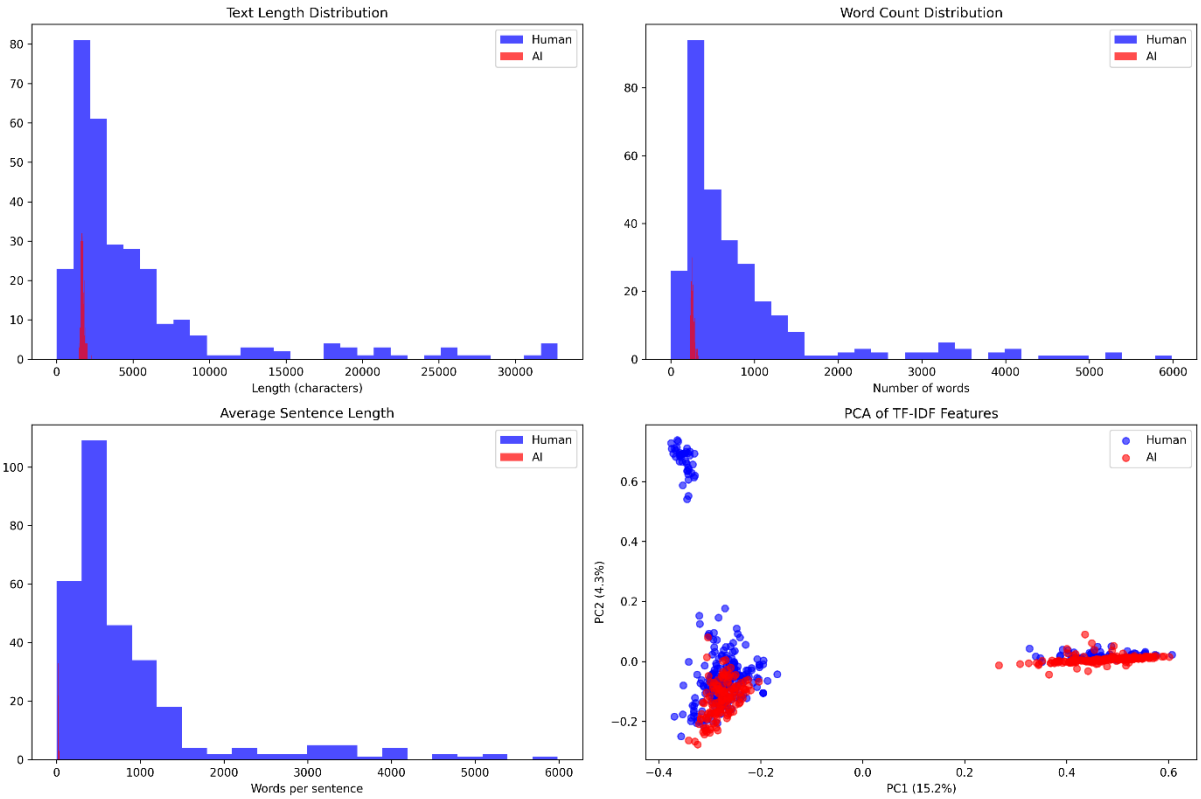

Figure 1 : Graphical representation of our dataset characteristics

From this, we identified the following possible bias sources :

- Size of the dataset compared to the number of parameters : we know that our dataset is too small for the model to train efficiently. We have around 600 texts for 110 million parameters.
- Length bias : human texts are systematically longer than AI-generated texts, as shown in the top graphics.
- Formatting patterns: there might still be consistent structural differences in punctuation and paragraph breaks even with cleaning. As shown by the bottom left graphic, AI tends to make far shorter sentences than humans.
- Vocabulary bias : the bottom right graphic represents the PCA of Term Frequency-Inverse Document Frequency features. It projects the vocabulary usage patterns into a 2D visualization where each point is one text document based on its word frequencies. Here, the PCA plot reveals that human and AI texts cluster in different regions rather than being randomly mixed, which indicates systematic differences in the choice of words between AI and human texts. It is possible that the model learns these vocabulary patterns rather than genuine stylistic differences.

That's why we tried different approaches to develop more robust, bias-resistant models.

IV.3. DistilBERT with and without truncature

The first problem we will try to handle is the parameter problem. For that, we implemented a distilled version of BERT that uses approximately 66 million parameters while retaining 97% of BERT's performance[2]. We also employed more aggressive regularization techniques (high dropout rates, limited fine-tuning and label smoothing) to prevent overfitting as much as we can given our dataset. Applying this model to our dataset still gave perfect results (**99.4% test accuracy**), which is still suspicious and reveals that parameter reduction and regularization alone are insufficient to address the biases.

That's why we implemented **text truncation** to directly address the length bias identified in our analysis, which appears to be the main bias (at least visible). We limited all texts to a maximum of 300 words. We ran a usual Student-test which revealed that, on average, length differences are **still statistically significant**. However, the metrics we obtain are far less "perfect", which, in our case, is less suspicious (we want the accuracy to be maximal, but not perfect, obviously). With that, our model has **an accuracy of 69.4%,** which seems more reasonable**.** The model shows balanced but asymmetric performance across both classes (cf. confusion matrix below), with an AI recall of 91.7% VS 77.0% for human recall.

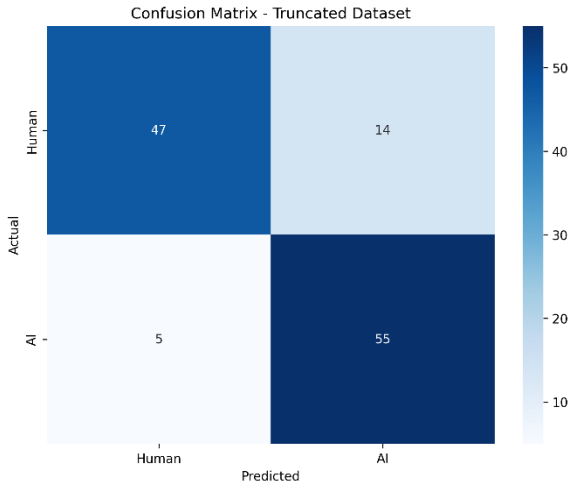

Figure 2 : Confusion matrix – DistilBERT with truncated texts

V. Conclusion

In this work, we applied both simple methods such as **SVC**, **Random Forest**, and **Logistic Regression**, as well as more complex models including **BERT** and **DistilBERT**. Throughout the project, we consistently faced issues related to overfitting.

Only the **DistilBERT** model provided a satisfactory performance, achieving an **accuracy of 69.4%**. This is the model we ultimately selected and validated for the task of distinguishing between real economic texts and AI-generated content.

To further improve performance, particularly with more advanced machine learning approaches, this analysis could be repeated using a much larger dataset, significantly beyond the current 600 texts.

[2] Source : *DistBERT, a distilled version of BERT : smaller, faster, cheaper and lighter*, Hugging Face.

VI. References

A. Vaswani et al., "Attention Is All You Need," 2017.

J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018.

V. Sanh et al., "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 2019.