Data Analysis Project: AI Text Detection

Jeremy Chan, Kevin Hu

1. Problem description and motivation

Amidst our digital age, the proliferation of misinformation has become one of the most pressing issues in media. With the rapid advancement and publicization of generative technology, the ability to generate human-like text has grown substantially, and has become more available than ever. With this comes the larger potential for misuse. This is especially problematic when these tools are leveraged to produce AI generated fake news. Fake news generated by these AI tools are not just grammatically and stylistically accurate, they also have the ability to mimic the tonality, structure and rhetorical devices used in reputable journalism. Therefore, to distinguish between AI and human is an increasingly difficult task. This raises an urgent question, **Can we accurately distinguish between AI generated and human written media using NLP techniques?** To answer the question, we use the publicly available Mirage-News dataset from Hugging Face (Huang, 2023) which consists of 15000 image caption pairs of AI generated and real news. Existing approaches to AI text detection commonly use large transformer classifiers, however this approach is lacking in transparency and replicability. In this analysis, we take a data driven approach utilising TF-IDF feature extraction combined with Naive Bayes (NB) and Random Forest (RF) classifiers, to tackle the task of classifying AI generated news using news captions. By using more interpretable models and a clean, balanced dataset, we hope to provide insight into the growing body of work that focuses on AI generated content detection.

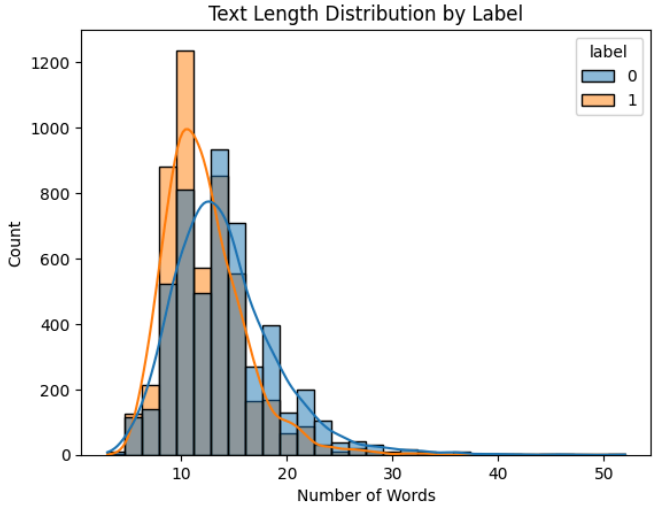
2. Describe the data

The dataset we used consists of 15000 records of news captions along with their label, with 1 being AI generated and 0 being human written. The data was split into 10000 for training, 2500 for validation, and 500 for 5 test sets (With differences in the news source and generative model used to obtain the news caption). We did not use an API to extract this data, instead, it was loaded on using the Hugging Face datasets library. This allowed us to take advantage of the predefined test subsets made by the creator of the dataset. We ended up utilizing only 3 of the test subsets (Test 1, 4 and 5) to avoid introducing too many different AI generation sources. Previous studies on AI content detection commonly analyze longer form texts such as reddit posts or essays using large transformer models. However, our dataset focuses on news captions, which is a short form journalistic text. A limitation of this is that news captions are usually self contained, lacking in contextual dependency. However, this enables a more direct

investigation on underlying lexical patterns in AI generated text. Another advantage of our dataset is it is perfectly balanced, with half of the news captions being AI and half of them being human, which avoids bias in the classification.

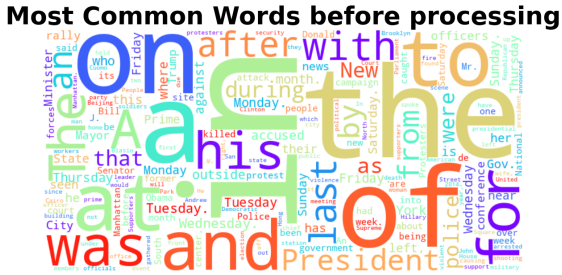
3. Exploratory data analysis

To start off the exploratory data analysis, we plotted the distribution of the text length for the two classifications.



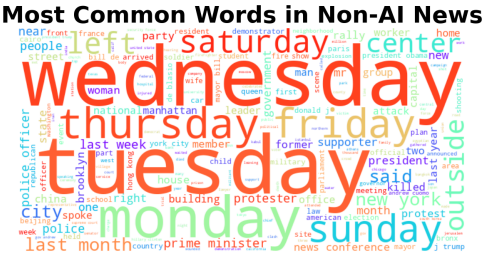
We can see that across most groups, the mode of the headline length is between 10 and 13, and both line plots exhibit a normal behaviour peaking at around similar values. This demonstrates that both AI and non AI generated headlines have comparable characteristics in text length.

A Wordcloud was then created to visually identify common recurring words across both classifications.



It is apparent that stopwords flood the Wordcloud. This further motivated data cleaning. Following the methodology similar to Part 1, we converted all texts to lowercase, removed punctuations, and removed all stop words. We then tokenized the texts, and then decided to lemmatize. We chose to do lemmatization over stemming since our investigation revolves around words, and we want to preserve clean and

meaningful words for our model. In some cases, stemming creates non real words which results in less interpretability.



Notably, the top words for the AI and non AI classification Wordclouds share similarities, both being days of the week which is understandable for news captions. Additionally, it is interesting to see that the AI captions like using “accuse”, possibly for clickbait purposes?

4. Describe the machine learning model

For the two models, we decided to go with the Naive Bayesian and the Random Forest. Both of these models predict a binary classification: predicting whether the text maps to an AI generated image or not. Additionally, both of these models are supervised learning algorithms, as the model learns from labelled data, where each data point comes with an assigned classification given by the dataset.

Firstly, we chose the Naive Bayesian model. The Naive Bayesian calculates the probability that a text belongs to each class based on words it contains using Bayes theorem. The Naive Bayesian model is extremely suitable for frequency data, and is computationally fast and lightweight. which applies particularly well to the purpose of this investigation. One glaring weakness that the Naive Bayes has is that it’s assumption being that the samples are independent from each other. This isn’t necessarily true in text labels. Due to grammar and sentence structures the prior word will influence the range of the next. However, the Naive Bayes provides a solid baseline model that we can compare the second model to.

Secondly, we chose the Random Forest model as it is capable of handling more complex, higher dimensional data. The model works by building many decision trees through bootstrapping on sample data to encourage variation, so that the collection of the trees becomes more robust and diverse. We believe it is appropriate for our investigation, as the nature of the random forest algorithm allows it to capture complex patterns in text. Furthermore, it is relatively robust to overfitting, as its implementation works by averaging across the majority of many trees.

To further transform our data to be usable as inputs to the two models, we vectorized the tokenized texts using TF-IDF. We chose to use TF-IDF over count vectors as TF-IDF introduces a weight component, which brings down the importance of common words, such as words that are often shared by both AI and non AI captions, and emphasizes rarer words that. As we want our model to focus on predicting a classification based on linguistic patterns, TF-IDF is more suitable since it helps our models to focus on more important and distinctive words, not just the common ones.

Thus, after processing the data, we were able to train our two models. As for the metric to evaluate how successful our model is, we will mainly focus on having a higher recall. This is because with a higher recall, there are less false negatives. We will also look at the overall accuracy as a metric.

We tested out different alpha parameters for our trained Bayes model and Random Forest model on the validation set. As a result, we found the Bayes model with alpha = 0.01 yielded the highest accuracy of 0.60 and highest recall of 0.20, and the Random Forest model with an accuracy of 0.57 and recall for AI of 0.146. Note that both of these recall values are extremely low, meaning our model that trained on the training data set struggled to detect most AI. However, both had an extraordinarily crazy recall value of 1. This means that our model is very likely to predict humans instead of AI given a text in the validation dataset. At this stage, we thought this could potentially be explained by the fact that the features used by AI and humans overlap significantly.

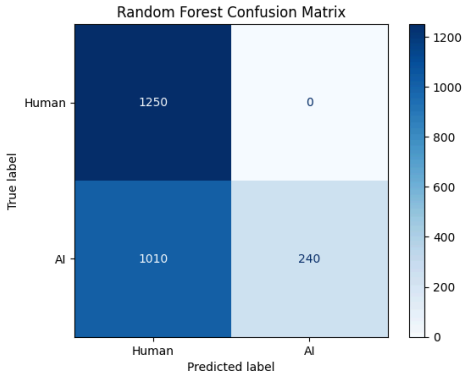
5. Results and Conclusions

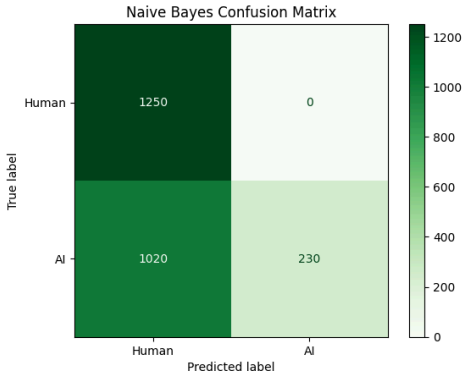
To evaluate our model, we tested the NB and RF classifiers on three test subsets. While both models performed reasonably well, it is apparent that NB outperforms RF across all three test subsets and all metrics.

Subset	Model	Accuracy	Precision	Recall	F1
Test 1	NB	0.864	0.90	0.82	0.86
	RF	0.808	0.82	0.79	0.80
Test 2	NB	0.792	0.76	0.86	0.80
	RF	0.718	0.69	0.78	0.73
Test 3	NB	0.80	0.83	0.76	0.79

	RF	0.746	0.77	0.71	0.74

One problem we ran into was the validation phase. During validation, our confusion





matrices revealed an unusual pattern.

Both models had 0 false positives. Furthermore, it predicted false much more than true. We were very confused by this result and attempted to investigate what caused this pattern. We believe that the AI model used to generate the validation set may have been stylistically closer to human text. However, this behaviour did not persist during testing, which is likely due to the test subsets utilizing different models.

Despite being a simpler model, Naive Bayes proved better suited for this task due to several lexical and structural advantages. Text classification, particularly with short-form inputs like news headlines, is primarily a lexical problem — it depends more on *which* words appear than on how those words interact or are structured. Naive Bayes, which directly models word-class probabilities, naturally aligns with this framework, especially when paired with TF-IDF vectorization, which emphasizes rare but class-informative terms. In contrast, Random Forest tends to perform better when richer feature interactions exist. However, with sparse and high-dimensional input like TF-IDF vectors, Random Forest struggles to form meaningful splits and may overfit to noise. This challenge is amplified in short texts, where interactions between words are minimal. Across all test sets, Naive Bayes maintained higher recall for AI-generated content, making it more reliable for identifying synthetic text — a key goal of our analysis. This outcome also reflects the stability of the model parameters: we used the standard Laplace smoothing parameter ($\alpha = 1$) for Naive Bayes with robust results, whereas Random Forest required careful tuning but still underperformed.

6. Citations

Huang, A. (2023). *Mirage-News Dataset*. Hugging Face.
https://huggingface.co/datasets/anson-huang/mirage-news