TruthTracker

A browser extension to detect fake news across the World Wide Web.

Anubhab Khanra - 2020A7PS2144H
Shyam N V - 2020A7SP2081H

What is Fake News?

- Fake News refers to false information that is intentionally spread online to mislead people.
- It is information that is fabricated and intended to deceive readers. The content can be designed to go viral and spread propaganda.
- Fake News stories are created to mislead, not report the truth. They are designed to misrepresent events for profit, views, or political purposes.

Our approach to combating it: A browser extension

- We developed a machine learning model that can identify Fake News with reasonable accuracy. We created an API based on this model to which a browser extension would pass news articles for classification.
- The browser extension would display the headline and article text to the API, which would return a prediction on whether the content is real or fake news along with the probability or confidence in the prediction. This helps users determine how confident they should be in the classification.
- By leveraging technology to automate Fake News detection, we aim to warn users before they view and unknowingly spread misinformation. A browser extension makes it simple to classify articles across the many websites where people consume news.

Improving Fake News Detection with Content-Aware Networks and Crafted Features

Part 1: An LSTM for Content Understanding

- We trained a Long Short-Term Memory network on a large dataset of real and fake news articles.
- LSTMs are ideal for analyzing sequential text data and uncovering subtle patterns that indicate deception.
- By learning directly from news content, this model develops an intuitive "feel" for the stylistic elements, omissions, or distorting rhetoric often found in Fake News.

Part 2: Targeted Feature Engineering and LightGBM

- Example features include: polarity/sentiment scores, word rarity/complexity, subjective/emotional language, lack of citations/evidence, inconsistent details, etc.
- These metrics provide quantitative measures of the attributes that often distinguish propaganda, conspiracy theories, satire, and intentional deception from truthful reporting.

Developing a Dataset for Training Fake News Detection Models

- The WELFake dataset was used, containing over 70,000 US news articles manually labeled as true or false. This dataset provides the ground truth data necessary to build accurate classification models.
- For each article, we extracted features from the headline and content that correlate with misleading information.
- This included metrics such as word/sentence length, vocabulary complexity, emotional/opinionated language, lack of evidence, inconsistent details, etc.
- To prepare the text for modeling, we first tokenized the headlines and content into individual words. We then lemmatized the words by identifying their canonical forms.
- This reduces vocabulary size, handles inflections, and increases regularization—all of which improves model performance, especially for scarce data.

Training and prediction pipeline + Results

Part 1: LSTM model

- The LSTM model was trained on the headlines and text bodies of the news articles. It achieved 99.8% accuracy on the validation set.

Part 2: LightGBM model

- 10 LightGBM models were trained on the features extracted from the news data. The predictions from these 10 models were combined to get the final prediction. This ensemble LightGBM model achieved 99.75% accuracy on the validation set.

Part 3: Model ensembling:

- The predictions from the LSTM model and LightGBM ensemble model were combined using majority voting. If the predictions of the two models didn’t match, the news article was classified as ‘fake’. We chose 'fake' over 'real' as classifying real news as fake is less harmful than misclassifying fake news as real.
- This model had an accuracy of 84% on a test set consisting of articles randomly chosen from the internet. This result is pretty good as the dataset consisted of mostly US articles of a narrow set of topics.

Creating a prediction endpoint

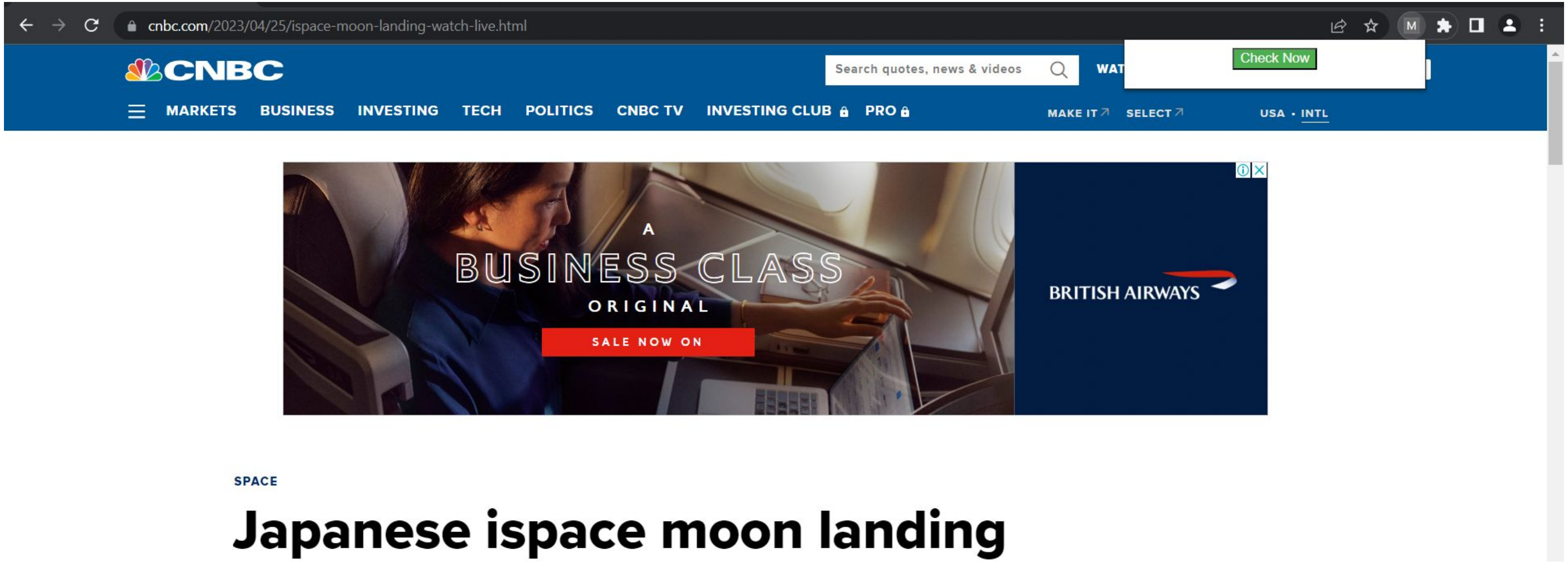
- We built a Python-FastAPI API and pipeline to deploy our fake news detection model.
- The API receives news headlines/text and returns fake/real predictions and probability scores in JSON.
- It enables integration with applications like our browser extension, sending predictions and confidence levels.
- Searches Google Fact Check API for public debunking results first. Only predicts articles without fact checks to maintain accuracy.
- Increases accuracy by applying an extra fact checking step.
- Returns probability scores, giving more context around predictions.

Browser Extension

- The next part of our project was to create a browser extension which could take the headline and body from news websites and pass it to the API.
- The chrome extension has a background service worker which sends a message to the content script whenever the required website is opened, which in turn takes the header and the body from the website by matching the HTML tags.
- This data is then passed as a JSON object to the API, which processes the information and gives its results back to the extension as a percentage.

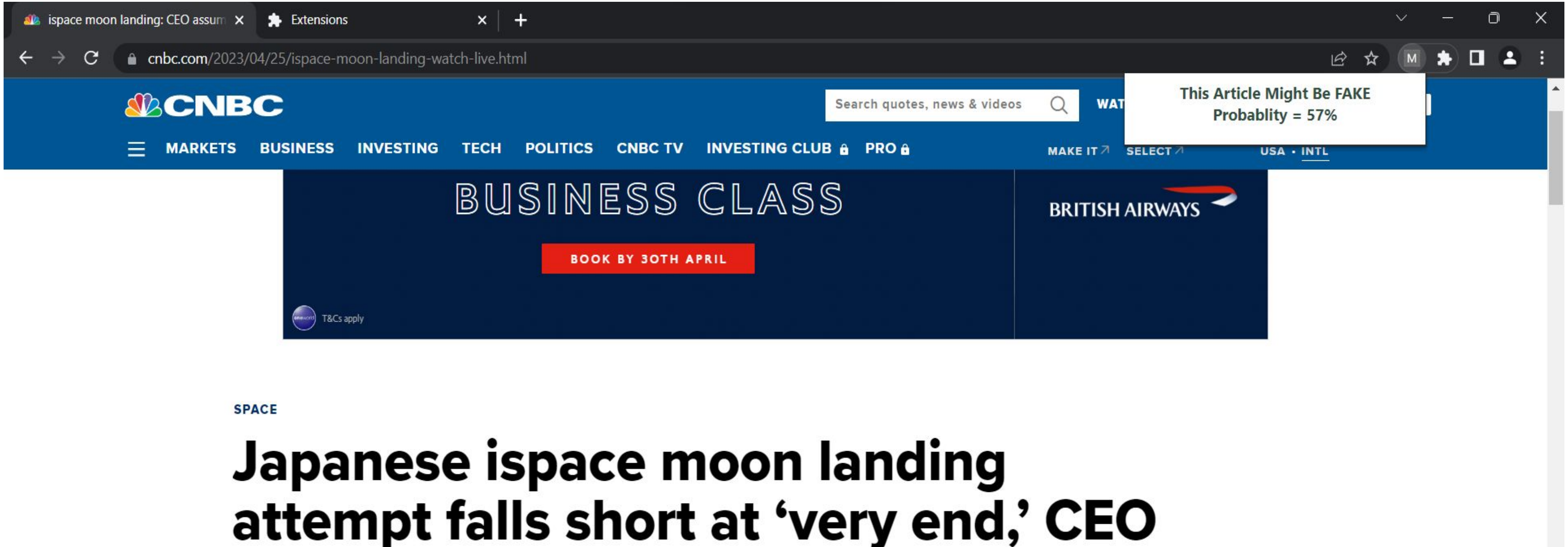
Working Samples

The user has to click on “Check Now” to get the results



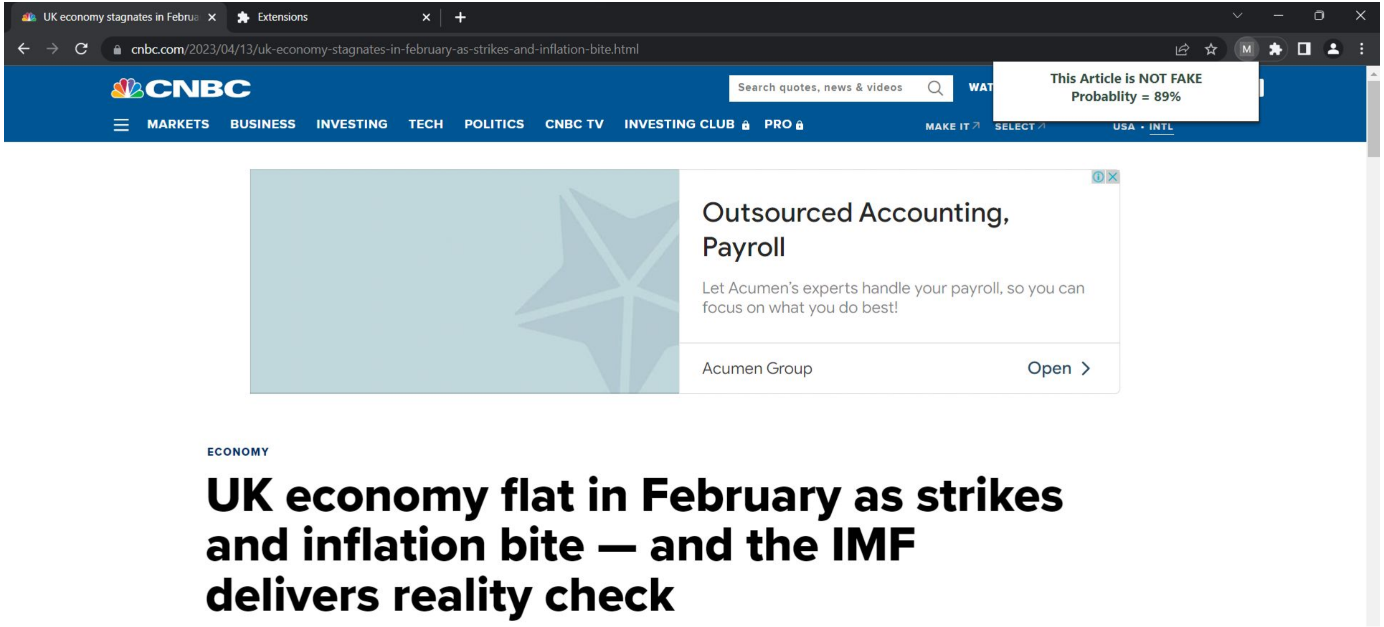
Working Samples

Example of a FAKE result



Working Samples

Example of a “NOT FAKE” result



Limitations

- Limited scope and coverage: The dataset used to train the models consisted only of US news articles from a few topics such as politics and world affairs. The models may not generalize well to a wider range of news sources, topics and regions.
- Monolingual support: As the dataset primarily contained English news articles, the detection models only work effectively for English text. They do not generalize to detect fake news in other languages.
- Narrow domain focus: The current models are optimized to detect misinformation in a limited set of news topics. They may not capture nuances required to identify fake news accurately across a much broader range of subject domains.
- ***Add extension limitations if any***

Scope of improvement

Despite the promising results, the detection pipeline still has some limitations regarding scope, coverage and generalizability that need to be considered for any real-world deployment. Some key ways to address these could be:

• Expanding the dataset to increase diversity in topics, sources, regions and languages.

• Retraining or fine-tuning the models on larger, more comprehensive datasets.

• Developing specialized models for different languages, topics or news domains based on their unique characteristics.

• Applying transfer learning to port the detection approach across domains and languages.

• Continuously evaluating new datasets and enhancing the models to expand capabilities over time