Title: Fake Political News Detections Using BERT

1. Introduction

1.1 Background

The widespread distribution of misinformation has become an increasing issue in the age of digital media, especially in political contexts where it can alter public perception or even affect election results (Bovet, 2019). Fake news is fraudulent information spread over social media platforms and the internet.
A phenomenon occurred during the presidential election of 2016 in the United States. The analysis of 171 million articles suggested that 25% of news-related articles mentioned fake or highly biased sources, indicating the severity of misinformation (Bovet, 2019). This is driven by the low cost and rapid transmission capabilities of social media, allowing false information to reach massive audiences with speed.

Detecting false information is a challenging task due to its intentional goal to mislead and mix it with real data. Although manual fact-checking is effective, it cannot match the enormous number of articles generated online. Automated solutions utilizing NLP and machine learning are necessary for effectively dealing with this issue. This project seeks to create a system that addresses political news by employing NLP techniques to effectively reduce misinformation.

1.2 Problem Statement

The unregulated spread of fake political news threatens the integrity of information. Current detection methods frequently depend on language metrics or basic machine learning, which cannot fully capture the complicated context and social dynamics of misinformation on social media. Furthermore, biases in datasets and ethical concerns, including privacy and justice, restrict the development of reliable systems. This research seeks to address these problems by developing a precise, accurate detection system built for political misinformation, utilizing advanced natural language processing to enhance conventional techniques.

1.3 Aim and Objectives

Aim:

Use advanced NLP techniques and the pre-trained BERT model, to develop a fake political news detection system with high accuracy and low bias.

Objectives:

1. Collect a labelled dataset with a close 50:50 fake and real political news ratio from diverse sources.
2. Preprocess and clean the dataset to produce high-quality input for training.
3. Implement NER to extract political entities for contextual analysis.
4. Implement BERT's contextual embeddings to increase classification accuracy.

5. Train and evaluate the system to identify real and fake news, and exceed the performance of baseline models.
6. Consider ethical and legal concerns, such as dataset balance and privacy, in the system design.

1.4 Product Overview

The project develops a Python-based software program that classifies political news articles as either real or fake, providing a confidence score. The system, built on BERT for text classification, initially achieves an accuracy of 82.55% on the FakeNewsNet dataset. This tool attempts to efficiently decrease the impact of political misinformation.

2. Literature Review

The rapid spread of misinformation is a significant threat to democratic discussion and public trust in media. The literature analyzes the difficulties in identifying fake news, evaluates conventional and modern approaches, and looks into how modern NLP techniques—such as BERT, NER, and propagation analysis can solve these challenges, providing a framework for the project's methodology.

2.1 The Characteristics and Challenges of Fake News

Fake news is defined as its aim to mislead and exploit the individual vulnerabilities of people (Shu, 2017), such as confirmation bias or social phenomena like echo chambers. On social media, misinformation spreads rapidly due to fraudulent accounts, artificial bots, and rapidly changing trends, making detection a complex task. The textual similarities between fake and real news limit reliance on content alone. These problems require a comprehensive strategy that combines linguistic analysis with social and contextual features.

2.2 Traditional Detection Methods

Fake news detection mainly relied on linguistic and stylistic examination. Methods focused on detecting warning indicators such as exaggerated language or grammar mistakes (Shu, 2017). These systems were able to detect certain fake content, but they had problems detecting politically related misinformation that was written to appear to be real. These methods rely on manually created features, limiting their scalability and flexibility to changing false information methods. It brings attention to the need for a data-driven, automated solution that is reusable across different datasets.

2.3 Developments in Machine Learning and Early Neural Models

Machine learning showed a transition towards improved detection methods. Early models, such as Support Vector Machines (SVM) and Naive Bayes classifiers, included variables like n-grams, sentiment polarity, and term frequency-inverse document frequency scores (Shu, 2017). These developments improved traditional approaches but remained restricted by their dependence on pre-defined features. The development of deep learning such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which can extract patterns from text. Convolutional Neural Networks are efficient in understanding specific patterns, while Recurrent Neural Networks can lay out sequential relationships (Shu, 2017)., their unidirectional processing limits contextual understanding is a limitation for political news which depends on a broader context.

2.4 Transformer Models and BERT

Transformer-based models addressed the above limitations by introducing bidirectional context (Devlin, 2019) introducing BERT (Bidirectional Encoder Representations from Transformers), a pre-trained model that uses Masked Language Modelling (MLM) and Next Sentence Prediction (NSP) to gain deeper contextual embeddings from unlabelled text. BERT allows a deeper understanding of linguistic nuances (Devlin, 2019). It showed outstanding results on many NLP tasks, achieving an F1 score of 92.8 on the CoNLL-2003 NER dataset and significant improvements in text classification.

In this project, I used BERT as the primary model for classifying political news, reaching an initial accuracy of 82.55% on the FakeNewsNet dataset. Its ability to recognise slight differences in tone and purpose makes it suitable for detecting authentic material from false information. However, it is high in computational complexity and requires extensive pre-training showing optimisation is needed in development.

2.5 Named Entity Recognition (NER) for Contextual Analysis

Named Entity Recognition (NER) improves fake news detection by identifying key political entities—such as politicians, organisations, or events, that provide context to evaluate reliability (Lample, 2016). Two neural architectures are introduced, Named Entity Recognition: a bidirectional LSTM integrated with a Conditional Random Field layer and a transition-based Stack-LSTM. It uses character and word embeddings to achieve excellent accuracy across multiple languages. The LSTM-CRF model, modelling both token-level and sequence-level dependencies, achieves outstanding results on benchmark datasets.

In this project, I used NER to extract political entities from news articles, cross-referencing with reliable sources to detect inconsistencies. The combination of NER with BERT embeddings strengthens the feature set, but at the cost of increased computational requirements, it is an issue that will be resolved in further development.

2.6 Ethical and Practical Considerations

Detecting fake news raises moral questions of bias, transparency, and privacy. Social biases in training data can cause misclassification of real information from minority groups. Using social media data causes privacy problems, requiring compliance to GDPR.
In this project, I used a balanced dataset and avoid sensitive user information, to comply with legal requirements.

3. Methodology

3.1 Research and Software Development Process

Step 1: Literature Review and Dataset Research

I started by reading papers on detecting fake news in order to set up my strategy with proven techniques and to focus on fake news misinformation. I chose transformer models like BERT, because of its amazing ability to understand text context. It is essential for identifying small variations between fake and real news. I was convinced that a transformer-based model would work best for capturing the complex nature of political articles.

The next step was to choose the right dataset. I considered a number of choices, FakeNewsNet dataset seems like a better choice. It provides political news articles with labels, which will save me time manually labelling data and trying to find fake news examples. Additionally, its 60% real news and 40% fake news ratio provided a balanced dataset for training. There are multiple sources from the dataset, which included news websites, Facebook, and Twitter, since it reflected how fake information spread on different platforms in the real world.

After choosing the dataset, I had to clean it and prepare for training. To ensure consistency, I removed duplicates and filling missing data. After that, I normalised the text by removing URLs, special characters, and punctuation and changed everything to lowercase. This helped the model focus on the text rather than formatting inconsistencies. Then I moved on to tokenisation, by dividing text into subword units using the BERT tokeniser. This step is important because it preserves the contextual meaning of political terms.

Step 2: Choose a Model and Start with Training

I decided to start with bert-base-uncased from Hugging Face's BertForSequenceClassification since it is practical and reasonable on my M1 MacBook with limited Graphics power. My laptop is powerful enough to complete the task effectively. Since case differences do not really matter in news classification it is a good idea to simplify the input.

I wanted to see how the model performed and tried to add more details to my progress report. I started preliminary training using the FakeNewsNet dataset. The initial result was 82.55% accuracy. I fixed the model initialisation problem. I ran into problems with weights, which I solved by fine-tuning BERT's pre-trained weights. Because of GPU memory limitations, I also had to be creative in order to maintain steady training without overloading my system I adjusted the batch size and used gradient accumulation. I used accuracy, precision, recall, and F1-score to assess the model, which provided me with a solid baseline to work on.

Step 3: Enhancing the Model with Features and Optimization

I wanted the model to pick up small differences better between fake and real political news. I looked to feature engineering. In order to gather political entities, like the names of politicians, organisations, and places, I first used SpaCy's en_core_web_sm to create Named Entity Recognition (NER). The model gained additional context about the main subject of the article by counting the occurrences. Then I made a list of terms like "election," "policy," and "congress" to make political jargon. I was able to keep track of how often these terms were used. The output of the model was improved by combining these features.

Then I make optimisation for the model. I increased the training epochs from three to five. Extra time was needed to train but accuracy increased. I want to find a sweet spot that keeps training stability. I also tested with different learning rates and batch sizes. After some testing, I switched to the Adam with weight decay optimiser, which increased the model's learning process. In order to prevent overfitting and make sure the model could generalise well. I have validation sets to keep track of the results.

Step 4: Evaluating, Analyzing, and Polishing the Model

I did 5-fold cross-validation, to see how the model performs. Then I examined all the false positives and negatives to see what went wrong. I observed trends: the model was tricked by unclear wording or overlapping entities. To fix this, I added data augmentation by switching synonyms to make the training data more diverse.

3.2 Technology

Dataset description

The project uses the FakeNewsNet dataset, a trusted resource for fake news research, containing political news articles labelled as real news and fake news.

Content of dataset:

• Real News: 569 articles sourced from news websites (91.1%), 53 from Twitter (8.4%), and 2 from Facebook (0.32%).

• Fake News: 390 articles from news websites (90.3%), 36 from Twitter (8.3%), and 6 from Facebook (1.38%).

• Total: 959 articles from news websites (90.8%), 89 from Twitter (8.4%), and 8 from Facebook (0.75%).

This distribution has a ratio of 60% real news and 40% fake news, for balanced model training. The inclusion of sources like social media and news websites allows cross-platform analysis. FakeNewsNet political dataset fits well for the use case of the project.

Data Preprocessing

• Text Cleaning: All text to lowercase, remove special characters, punctuation, and URLs using regular expressions to minimise noise.

• Tokenization: Use the BERT tokeniser to split text into tokens in BERT's input format, providing contextual information.

It simplifies input, reducing differences that could affect feature extraction and classification.

Customized features

•Political jargon was developed to enhance the model’s ability to detect political terms in news articles, which can increase the performance in classifying fake from real articles. The political consists of frequently used political terms for example "election,", "congress," "campaign," " nominee," "vote," " pivot," " markup," and "bill"—the list was added by sourcing additional political terms from online sources such as Political Dictionary. The political jargon function processes and scans through each article, calculating a jargon count feature that is combined with the NER feature, and then passed into the classification layer.

• Named Entity Recognition (NER) was developed to extract contextual features from the news article, enhancing the model’s ability to identify entities related to political news, such as names of politicians, organizations, and locations. Using the SpaCy library, I used the en_core_web_sm model because it is a lightweight English language model.
NER function iterates news articles, applying SpaCy’s NLP pipeline to detect entities, and returns the count of entities identified for example “Kier Stamer” as a person. This count is used as a numerical feature capturing the presence of named entities, which is then combined with the RoBERTa output layer and then passed into the classification layer.

Model Architecture and Training

I originally used bert-base-uncased, then I changed to custom RoBERTa-base. It is because Roberta is richer in pre-trained knowledge, it used 30 billion words to pre-train compared to 3.3 billion words in Bert base. RoBERTa has seen more political terms, jargon and patterns. I believe RoBERTa can generalise better than Bert base.

Using RoBERTa's tokeniser, which is based on a 50,000-token Byte-Pair Encoding vocabulary, I started with input processing, tokenising each news article from fake and real news. RoBERTa has a better context understanding, so I limit it to 128 tokens, less than BERT's 512. This process was efficient on my M1 MacBook and kept enough context. Padding and truncating are needed to ensure efficiency. But I keep all the key information.

Then, I put tokenised sequences through RoBERTa-base for encoding. The 12 transformer layers handle the input by combining each token's embedding with a position embedding. The multi-head self-attention mechanism, keeps track of every token in relation to other tokens, so it can capture context by knowing the relationship. After passing the layers, I got a set of 768-dimensional state vectors for every token.

Classification layer, for the model to make predictions, I have to add an output layer for the classification result. I added a final layer on top of the pooled output. To improve the performance, I also added two scores generated from NER and political jargon features and then combined them with the output layer.

Training, The FakeNewsNet dataset has 1,000 training data, 60% real and 40% fake. Since it is a small dataset I used 5-fold cross-validation to better make use of the dataset and prevent it from overfitting. I used a fixed seed random_state=42 to make it so I could reproduce the result and perform further debugging and development. I split each fold into 80% training data and 20% testing data, with 10% of training as validation. I also tested different hyperparameters. I tested batch sizes of 16 and 32, and learning rates of 1e-5 and 2e-5. The Adam with weight decay was really effective. It provides adaptive updates, with weight decay, keeps the training stable and works well with small dataset.

Since there is a 60:40 split between real and fake data. To reduce imbalance, I used computed class weight. After balancing, the weight of the fake is higher than real. This can make it fair when making predictions, I set training to run up to 10 epochs per fold but added early stopping after 2 patient epochs if validation loss doesn't improve. This is necessary because my dataset was small, and it can effectively reduce overfitting. Executing this on my M1 MacBook with Metal Performance Shaders(MPS) acceleration improved the computation speed. It reduced the 5-fold runtime to about 3-4 hours. It is way faster than using CPU alone.

The classifier was based on Python. I used Hugging Face RobertaModel and RobertaTokenizer from the library to handle text encoding and contextual embeddings. My earlier BERT baseline model achieved 82.55% accuracy without NER or jargon features.

For tensor operations and model training, you used PyTorch, including torch.nn for CustomRobertaClassifier, torch.optim for the Adam with weight decay optimizer. torch.utils.data for the DataLoader. Scikit learn for data splitting with train test split and 5-fold cross-validation. Result metrics like accuracy score, confusion matrix, and roc score. I used pandas to load and manage my datasets politifact_real.csv and politifact_fake.csv. NumPy for array operations.

3.3 Version management

Google share drive(source code):
https://drive.google.com/drive/u/1/folders/1zwqYoSSP32XGlFu3LLiB8EYS2yyGoknv

4. Results

The baseline model using BERT-base-uncased was trained using 80 to 20 train and test split with a batch size of 16 and a learning rate of 5e-5 for 3 epochs. The model achieved an accuracy of 82.55%.

	Fake	Real
Precision	0.72	0.91
Recall	0.87	0.8
F1-Score	0.79	0.85

Confusion Matrix for baseline model

	Predicted Fake	Predicted Real
Actual Fake	68	10
Actual Real	27	107

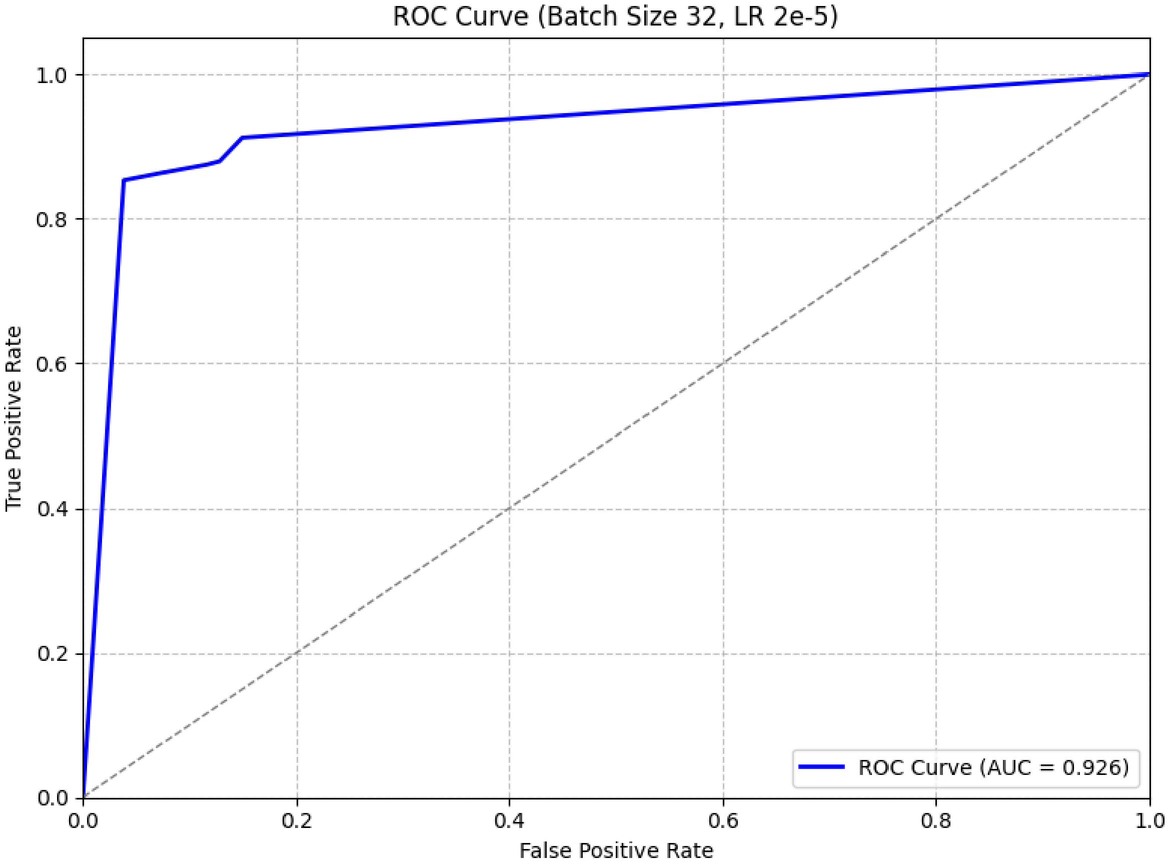
The experiments tested four combinations of hyperparameters across 5-fold cross-validation, specifically batch sizes of 16 and 32 paired with learning rates of 1e-5 and 2e-5. I used average accuracy for each combination to show the difference in performance. With a batch size of 16 and a learning rate of 1e-5, the model achieved an average accuracy of 87.69%. The same batch size with a learning rate of 2e-5 achieved the same average accuracy of 87.69%. Increasing the batch size to 32 with a learning rate of 1e-5 achieved a higher average accuracy of 88.16%. The best performance was with a batch size of 32 and a learning rate of 2e-5, achieving an average accuracy of 88.45, which is the best combination.

RoBERTa model (Batch Size 32, LR 2e-5)

Fold	Accuracy	Precision (Fake)	Recall (Fake)	F1-Score (Fake)	Precision (Real)	Recall (Real)	F1-Score (Real)
1	90.09%	0.81	0.96	0.88	0.97	0.87	0.92
2	91.47%	0.85	0.92	0.88	0.95	0.91	0.93
3	87.68%	0.81	0.87	0.84	0.92	0.88	0.9
4	86.26%	0.88	0.85	0.86	0.85	0.88	0.86
5	86.73%	0.83	0.88	0.86	0.9	0.85	0.88
Average	**88.45%**	**0.84**	**0.9**	**0.86**	**0.92**	**0.88**	**0.9**

Confusion Matrix for Roberta model Hype (Batch Size 32, LR 2e-5)

	Predicted Fake	Predicted Real
Actual Fake	77	9
Actual Real	15	109



Using Batch Size of 32, learning rate of 2e-5 the accuracy achieved 86.26% to 91.47%, it outperformed the baseline model with an accuracy of 82.55%. The Roberta model achieved an average precision of 0.84, recall of 0.9, and F1 score of 0.86 for fake news and 0.92, recall of 0.88, and F1-score of 0.90 for real news. Correctly classifying 77 fake articles and 109 real articles, with only 9 false positives and 15 false negatives. This performance, is supported by a high ROC of 0.926 as shown. Showing significant improvement by adding NER, political jargon features and performing hyperparameters tuning.

Critical evaluation

The 5.9% accuracy difference between RoBERTa and baseline BERT model are from multiple factors. NER and political jargon features are one of the major factors. In addition to RoBERTa advanced pre-training and cross-validation, it surpasses baseline BERT in terms of contextual understanding. RoBERTa has a better ability to classify fake and real news. It has a higher ROC-AUC 0.9260 compared to 0.8957 for baseline BERT model, which is reinforced by entities and jargon understanding that BERT lacks.

NER and Political Jargon Features effectiveness

In order to verify the factual consistency of news articles, NER detects entities such as individuals, groups, and places. When combined with RoBERTa's contextual understanding, NER could help identify inconsistencies if a fake news article incorrectly links a politician to an incident. Political jargon features, capture terms unique to the political domain, enhancing the model's understanding of political context.

These features are combined with RoBERTa output before feeding into the classification layer, improving the input with linguistic signals. RoBERTa uses this extra information to reduce misclassifications, which leads to better recall score(0.9 compare to 0.87 with baseline BERT model for fake news) and precision score(0.84 compare to 0.72 with baseline BERT model for fake news).

False detection Analysis using fold 4 of RoBERTa model

False Positives: Real News Predicted as Fake

- "Hillary Clinton says guns exceed next nine categories as leading cause of death for young black men"
- "To young people who are undocumented: This is your country, too."
- "Who Benefits from President Trump’s Child Care Proposals"
- "No MidOct Paycheck for Troops if Government Shuts Down Lawmaker says"
- "China's Space Threat: How Missiles Could Target U.S. Satellites"
- "Missing Teen-Ager Found in New Jersey"
- "Democratic debate: Bernie Sanders really isn't as "socialist" as President Eisenhower"
- "Statement from FDA Commissioner Scott Gottlieb, M.D., on FDA’s ongoing efforts to help improve effectiveness of influenza vaccines"
- "H1N1 vaccinations to be offered to Guantanamo Bay detainees"
- "Writers May Have Pact. Huckabee Crosses Picket Line—Again"
- "Palin: Pioneer, maverick -- and now game-changer"
- "New York Officials Welcome Immigrants, Legal or Illegal"
- "Donald Trump featured in new jihadist recruitment video"
- "Peanut Product Recall Took Company Approval"
- "Obama Promises Abortion in Public Plan"
- "Israel should welcome Rouhani's election victory"
- "Rand Paul: Will Donald Trump betray voters by hiring John Bolton?"
- "Vladimir Putin's Approval Rating Hits All-Time High, Boosted by Syria Airstrikes"

- "As Americans Save Money, their Government Spends with Reckless Abandon"
- "The page you're looking for may have been moved or renamed"
- "Mike Huckabee: Fried Squirrel Out of a Popcorn Popper"
- "Schneiderman suing Trump over DACA cites 'discriminatory animus'"
- "Terror Suspects Are Buying Guns - and the FBI Can't Stop Them"
- "U.S. considers 3,000 more troops for Afghanistan"
- "Sen. Sanders warns of ‘frightening trend’ towards oligarchy"

False Negatives: Fake News Predicted as Real

- "LAW PASSED: All Child Support in the United States Will End by Beginning of 2018"
- "Democrat Maxine Waters Has Shown Up To Only 10% Of Congressional Meetings For 35 YEARS"
- "US Representatives Promise Implement Of UN Gun-Control Plans"

The first reason for false positives is the model over emphasis on emotional or compelling language, which is frequently linked with fake news, particularly in political or crisis reporting. Real news outlets often use attention grabbing headlines to draw attention, which is similar to fake news. "Terror Suspects Are Buying Guns - and the FBI Can't Stop Them," It uses alarming words to draw attention to a major security issue. These real articles were classified as fake news by the model, this suggests that the model is sensitive to emotional and dramatic tone. This highlight only relies on textual features struggle to identify some of the edge cases, since it lacks true lack true context awareness.

Secondly, training data bias is one of the reasons for misclassifications, since the training data may lack real news with emotional or political charged content, causing it to recognize as fake by default. For example, “China's Space Threat: How Missiles Could Target U.S. Satellites" It is using a threatening tone, so the model classifies it as fake news since there is no coverage from the real news training data.

Insights and Future development Directions

BERT-based models typically range from 85 to 90% accuracy in detecting false news;.RoBERTa model's performance aligns with the range. However, it lack propagation data such as article spread speed, share count, like count, which could further improve accuracy. However, RoBERTa has high computational complexity and relies highly on dataset quality. It reached the bottleneck. To further improve accuracy and reduce false detection, future work could use multiple datasets which include propagation data. Also, a more powerful system can allow the model to use full 512 token size compared to 128 token size used in current RoBERTa model could potentially increase the accuracy of the model.

Constraints and Limitations

Computational constraints affected the development of the project. My M1 MacBook used Metal Performance Shaders (MPS) acceleration. It made training faster but with limited memory on my device I am with restricted batch size and number of epochs. I encountered multiple problems during training since the laptop was out of memory. I tried Google Colab for training but with hardware limitations. It took way longer to run the training compared to my local device. It prevented me from testing larger models. The model's lack of true context awareness was one of the limitations. Since classification was only based on textual features. It did not integrate with external fact checking databases or historical credibility data from different sources. It is not reliable enough where deeper verification is required.

5. Professionalism

5.1 Project Management and planning

September: Project Initiation & Research

- Defined research problem and objectives.
- Conducted preliminary literature review on fake news detection models and NLP techniques.
- Explored datasets, specifically FakeNewsNet, and determined suitability.
- Initial project planning and setup of Git repository for version control.

Reflection: The month focused on gaining a deeper understanding of the research topic and defining clear objectives. Early literature review helped shape the project’s direction, though narrowing down methodologies was challenging.

October: Literature Review and model selection

- Conducted an in-depth review of different transformer models.
- Studied political jargon and feature engineering techniques.
- Experimented with different preprocessing techniques and tokenization methods.
- Submitted project proposal and received feedback.

Reflection: Progress is challenging, it is difficult to perform effective feature extraction for political context. To address this, I improved keyword selection techniques for political jargon.

November: Model Development & Dataset Preparation

- Finished dataset preprocessing strategies
- Began training baseline BERT model for classification.
- Initial testing and recorded accuracy metrics.

Reflection: The dataset required more cleaning than initially expected, particularly in handling phrases. Feature engineering and jargon-based filtering proved effective in improving model understanding.

December: Model Optimization & Experimentation

- Fine-tuned BERT model for improved classification accuracy.
- Performed hyperparameter tuning on learning rate, batch size, and dropout.
- Compared accuracy metrics between different hyperparameters
- Evaluated false positives and false negatives to refine custom features.

Reflection: Model accuracy improved after hyperparameter fine tuning, but there are false negative and false positive predictions. Future improvements should focus on contextual word embeddings and additional validation set, in order to decrease false classficaiton.

January: Evaluation & Error Analysis

- Performed testing and cross-validation.
- Analysed misclassified samples to understand model limitations.
- Addressed bias in dataset distribution.
- Added additional regularization techniques for training.

Reflection: Error analysis revealed that political satire and exaggerated statements led to misclassifications. Additional training on different sources may help to solve this issue.

February: Documentation & Finalization

- Wrote initial drafts of the dissertation.
- Created visualizations and results analysis.
- Refined model explanations and methodology descriptions..

Reflection: Focus on explaining feature selection and preprocessing approaches.

March: Submission and Presentation Preparation

- Finalized dissertation
- Make presentation slides
- Prepared for presentation

Reflection**:** The project journey was rewarding and challenging. A better initial planning with clear tasks and measurable goals for each week could have reduced some delays, especially in dataset selection and processing.

5.2 Risk

One of the risks during the project came from the quality and balance of the dataset used to train the model. Although the FakeNewsNet dataset is a reliable source for classifying political news, the 60 to 40 real and fake news split. This could impact the model's performance by favouring the majority class, so balancing dataset distribution is important. For example, Fold 4 included 107 fake news pieces and 104 real news articles, which was almost equal. By making this change, the model avoided overfitting to one class and proper weight for each class.

Another risk came from the project relying on high computational power. It is restricted by both hardware limitations and the time it takes to perform training. Google Colab, a cloud-based platform that provided free GPU access, was a starting point of the development. However, due to usage restrictions can constant disconnection, I have to use my local device M1 MacBook for training using the slower Metal Performance Shaders (MPS) for acceleration. During the hyperparameter tuning and testing stage, I have to perform multiple iterations to optimize the RoBERTa model. To speed up workflow, the training was changed to use early stopping that would end epochs when the validation loss stopped improving. I explored larger models like RoBERTa Large to further increase the performance of the classification. However, the computational limitations, I redirected back to RoberTa base and focused on optimizing the features and parameters.

Model underfitting is the third risk, with only 1,000 articles for the dataset. Roberta consists of millions of parameters, there is a risk of underfitting due to small dataset. There is a possibility that RoBERTa could not generalize well with limited training data. I tried to solve it by Learning Rate Scheduler. The scheduler adjusts the learning rate dynamically. It allows the model to take smaller steps to converge, which helps it to learn efficiently from the small dataset.

Looking to the future, several risks appear when the project grows. Scalability is a major challenge because computational requirements can exceed the capabilities of existing hardware if the dataset expands or includes real-time social media propagation data. Distributed training frameworks or powerful GPUs may be required. Domain adaptation is another challenge, as the model's training on political news may not translate completely to other domains like science and finance, which might require retraining or transfer learning to maintain accuracy.

5.3 Professionalism

Ethically, the risk of misuse, and misinterpreting the model's outputs to weaken credible journalism. It will weaken public confidence in the media. For example, a user could misuse the model to label a real article as fake. Then use the prediction to discredit the news website and suppress opposing reports and articles, which could mislead the public. A transparent explanation of the model's limitations and usage instructions is necessary to prevent this define it just as a reference and acknowledge users for potential misclassification.

Legally, the project followed data usage and intellectual property regulations during its development. The FakeNewsNet dataset was used under non-commercial academic conditions to comply with intellectual property regulations as required by BCS part 2d. However, a real-world deployment could create additional legal challenges, evaluating news articles from copyrighted sources needs permission to avoid violating intellectual property rights. The current dataset does not contain personal information. Future versions adding social media metadata such as user interactions, comments, share and like counts could violate privacy regulations like the General Data Protection Regulation (GDPR). To ensure that the project meets legal requirements for privacy and accountability. It requires anonymization and professional data-handling techniques to ensure user privacy.

Socially, the project has the chance to improve information integrity by accurately identifying fake news, which will improve public access to reliable information. It is a benefit correlated with BCS part 1d focus on equal access to IT advantages. However, the risk of false positives incorrectly identifying genuine reports as fake news may damage public confidence in journalism. In order to address this, the project puts a high priority on being truthful and open about the uncertain nature of the model and its limitations and encourages users to fact check the result.

The project shows professional competence and integrity. By developing a fake political news classifier using NLP techniques, I advanced my skills in AI ethics and machine learning. I also illustrate transparent documentation of methods, limitations and results.

6. Conclusion:

This project successfully developed RoBERTa Base model and Natural Language Processing (NLP) to create an accurate fake political news detection model. The model performed better than baseline model by combining contextual features like Named Entity Recognition (NER), political jargon and hyperparameter tuning. The RoBERTa model outperformed the baseline BERT model, which had an accuracy of 82.55%, with an average accuracy of 88.45% across 5-fold cross-validation. The result shows the ability of transformer based contextual embeddings and custom features combined to recognise small differences between real and fake political news. Ethically, the project addressed transparency by analysing limitations and providing usage guidelines to prevent misuse. Legally, it complied with the research license for the FakeNewsNet dataset, and socially, it aimed to improve information integrity.
In the future, there are several ways to enhance the model's capabilities. Adding propagation parameters, such as(spread speed, share count, like count), could improve accuracy by taking social media engagement data into consideration. Accuracy could be improved by using more efficient architecture like DistilBERT or larger transformer models like RoBERTa-Large. Adding external real time fact checking databases, allows the model to have true context awareness. It will compare and use historical data and compare numeric data from different sources. Expanding the dataset to multiple sources could solve training data bias and improve generalisation.

7. Bibliography

1. Bovet, A., Makse, H. (2019). Influence of fake news in Twitter during the 2016 US presidential election. Available at: https://scholar.google.ch/citations?view_op=view_citation&hl=fr&user=rbHfk1EAAAAJ&citation_for_view=rbHfk1EAAAAJ:wMgC3FpKEyYC

2. Devlin, J., Chang, M., Lee, K., Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Available at: https://aclanthology.org/N19-1423.pdf

3. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C. (2016). Neural Architectures for Named Entity Recognition. Available at: https://www.researchgate.net/publication/305334469_Neural_Architectures_for_Named_Entity_Recognition

4. Shu, K., Sliva, A., Wang, S., Tang, J. and Liu, H. (2017). Fake news detection on social media: A data mining perspective. Available at: https://www.semanticscholar.org/paper/Fake-News-Detection-on-Social-Media%3A-A-Data-Mining-Shu-Sliva/cb40a5e6d4fc0290452345791bb91040aed76961

8. Appendices

Google share drive(source code):
https://drive.google.com/drive/u/1/folders/1zwqYoSSP32XGlFu3LLiB8EYS2yyGoknv