Université d’Ottawa
Faculté de génie

uOttawa
L’Université canadienne
Canada’s university

University of Ottawa
Faculty of Engineering

ELG5901 Electrical Engineering Project

Final Report:
Cover Page

Student name: Ahmed Ibraheem Ahmed Ali Badawy **ID 300389393**
Student name: Anas Ibrahim Ali Elbattra **ID 300389368**
Student name: Esmael Alahmady Ebrahim Ezz **ID 300389885**
Student name: Yousef Abdelhaleem Ahmed Shindy **ID 300389365**

Graduate Program:
Master of Engineering Electrical and Computer Engineering – Artificial Intelligence and Data Science Track (DEBI Program)

Semester to Register: 2023 Fall

Project Title: News Understanding

Table of Contents:

Table of Figures:

Table of Tables:

Acronyms

- AI - Artificial intelligence
- LLMs – Large Language Models
- NLP - Natural Language Processing
- CNN – Convolutional Neural Network
- LSTM – Long Short-Term Memory
- API – Application Programming Interface
- IDE – Integrated Development Environment

1. Introduction

1.1 Problem Definition

In the dynamic world of news consumption, challenges rising from misinformation, biased reporting, and the need for efficient content summarization presents a significant challenge that needs innovative solutions. For Microsoft, our “News Understanding” project aligns with the company’s commitment to innovation and cutting-edge solutions in Artificial Intelligence (AI) and Data Science. Addressing concerns related to misinformation and biased reporting reinforces Microsoft’s dedication to responsible AI and promoting trustworthy information sources. The broader benefits extend to media organizations, researchers, and the public, fostering and informed and discerning public. The project deploys Large Language Models (LLMs) to tackle diverse tasks, including distinguishing between authentic and deceptive news, identifying reporting bias, classifying topics, and extracting sentiment. The anticipated advantages include an enhanced understanding of news, offering valuable insights into the capabilities and limitations of LLMs, and contributing to trustworthy journalism in the digital era.

1.2 Background

In recent advancements in fake news detection within the academic landscape, Essa et al. [1] have notably introduced a hybrid model that combines the deep language understanding of BERT with the feature-learning power of LightGBM. This innovative approach outperformed traditional and deep learning methods, achieving an impressive 99.88% accuracy on the ISOT dataset and notable results on TI-CNN (96.94%) and FNC (99.06%). Emphasizing the importance of analyzing both headlines and full text, the hybrid architecture demonstrated its efficacy in combatting misinformation. Additionally, Alsini's research [2] on real-time Twitter sentiment analysis, employing CNNs and LSTMs, yielded promising results with CNNs achieving 93.7% accuracy and LSTMs achieving 94.2% on a custom topic-specific dataset. The integration of both models in a hybrid approach further elevated accuracy to 95.5% on custom data, showcasing the synergistic strengths of CNNs and LSTMs. Furthermore, Ansary's innovative summarization method [3], which combines TextRank, sentiment analysis, and category-wise keywords, has demonstrated superiority over existing methods on a Bengali dataset. This approach excels in capturing key points, identifying important sentences, and maintaining fluency, showcasing its potential for generating concise and relevant newspaper summaries.

Beyond these specific models, key contributions from industry players have also significantly impacted natural language processing (NLP). Hugging Face [4] has played a crucial role through the Transformers library and the Hugging Face Hub, providing pre-trained transformer-based models, tools, and a collaborative platform for sharing and utilizing NLP models. Additionally, Google AI's Gemini API [5] has emerged as a versatile text AI, empowering tasks such as content generation, translation, question answering, and creative writing. OpenAI's GPT-4 [6], as a multimodal AI, has pushed the boundaries of language processing, excelling in text and image analysis, long-form content mastery, and enhanced creativity. Furthermore, for those engaged in building, training, and deploying neural networks and machine learning models, ktrain [7] offers a lightweight wrapper for TensorFlow Keras, making deep learning and AI more accessible to both newcomers and experienced practitioners. Overall, these diverse contributions showcase the evolving landscape of fake news detection and sentiment analysis, fueled by innovative models and collaborative efforts within the NLP community.

1.3 Project Context

Various external systems and tools were integral to the successful implementation of our "News Understanding" project. We extensively utilized Language Models (LLMs), deploying them both on Google Colab Pro and locally through IDEs such as VS Code, PyCharm, and Android Studio. Python notebooks within these IDEs facilitated our development process, with essential libraries like numpy, pandas, sentence piece, ktrain enhancing our capabilities in data manipulation and model training. Furthermore, we incorporated Flutter and Flask frameworks, along with ngrok for secure tunneling, to build the user interface and handle web interactions. The project also involved the integration of external APIs, including Hugging Face, GPT-4 API, and the Gemini Pro API, to leverage advanced language processing capabilities.

To ensure the project's success, we maintained regular communication with key stakeholders. Our sponsor, Microsoft, played a pivotal role in guiding the project's direction, providing valuable feedback, and offering suggestions. Additionally, our engagement with uOttawa support involved ongoing meetings to seek guidance and support throughout the project lifecycle. These interactions were essential to align our efforts with project objectives and address any challenges that arose. Ensuring accessibility and availability, both Microsoft and uOttawa support confirmed their commitment to being actively involved in the project, emphasizing collaborative efforts to achieve desired outcomes.

2. Design Overview

The system involves a user-friendly mobile application allowing users to input news articles and choose between classification or summarization tasks. The backend, powered by Flask and accessible online through ngrok, handles the processing. Depending on the selected task, the input undergoes specific preprocessing before being fed into the appropriate NLP model—DistilBERT, RoBERTa-large, T5-small for classification, and BART-large for summarization.

Flask manages the flow between the mobile app and the backend, ensuring seamless communication. After processing, the results are sent back to the mobile app, where classification outcomes include bias, fake news, sentiment, and overall topic. Summarization results consist of a concise summary and the count of new words introduced.

This streamlined architecture enables users to effortlessly submit news articles, select tasks, and receive real-time, organized results on the mobile application. The use of Flask and ngrok optimizes data flow and accessibility, providing an efficient and user-friendly platform for news classification and summarization.

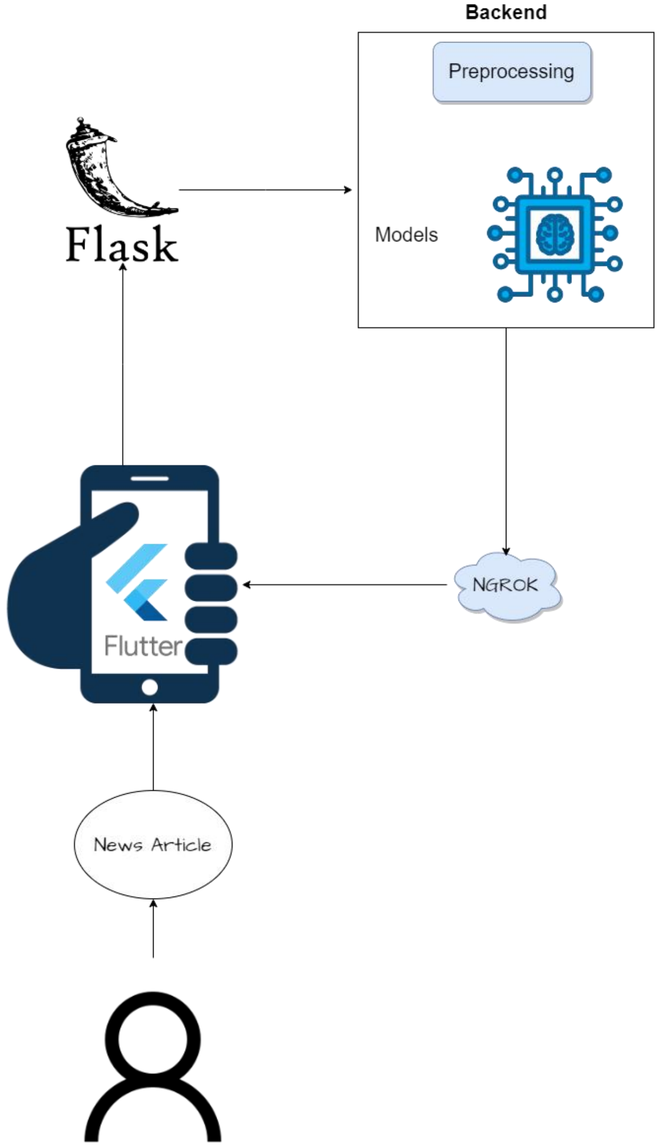

Figure 1: Architectural Diagram

2.1 Requirements

The overarching goal is to create a versatile system capable of classifying news articles across key dimensions:

Fake vs. Real News:
The system should effectively distinguish between fake and real news, employing advanced models to scrutinize the content and provide accurate classifications.

Bias Classification:
The system needs to categorize news articles based on political bias, distinguishing whether the content is Left-biased, Right-biased, or falls within the Center.

Sentiment Classification:
A critical requirement is the ability to perform sentiment classification, determining whether the sentiment of the news article is negative, positive, or neutral.

Breaking vs. Non-breaking News:
The system should identify and classify news articles as breaking or non-breaking, providing users with timely information on the significance of the news.

Topic Classification:
The system is expected to accurately classify the topic of each news article, allowing users to quickly understand the content's subject matter.

Summarization:
An integral part of the system is the capability to generate concise summaries for news articles, providing users with an efficient way to grasp the key information.

The end-user expectations are clear – a comprehensive and accurate news classification and summarization system that aids in understanding the authenticity, bias, sentiment, significance, and topic of news articles. The design and implementation of the system will revolve around meeting these requirements to deliver a valuable solution.

2.2 Detailed Design

The decision to utilize open-source LLMs and Transformers from Hugging Face forms the backbone of the system's detailed design. For distinct classification tasks, the following models were selected:

DistilBERT for Bias and Topic Classification:
DistilBERT was chosen for its efficiency in bias and topic classification. During training, the model was fine-tuned to return the same probabilities as the BERT base model. Its smaller size and faster execution made it the optimal choice for these tasks.

RoBERTa-large for Sentiment Classification:
RoBERTa-large was selected for its superior performance in sentiment classification. Its robust architecture proved effective in capturing nuanced sentiments within news articles.

T5-small for Fake Classification:
T5-small exhibited remarkable accuracy in classifying fake and real news during the initial stages of the project. Its successful performance in this task led to its adoption for fake news classification.

BART-large for Summarization:
The choice of BART-large for summarization was determined by an evaluation using BERT score. Among competing models (BART-large, T5-small, T5-base, Pegasus), Both BART-large and Pegasus achieved high metrics in BERT score. However, after manual evaluation, we chose BART-large as it consistently outperformed other models confirming its effectiveness in generating accurate and concise summaries.

Interaction and Interrelation:
The subsystems within the backend, each dedicated to a specific classification task, interact seamlessly. The Flask API orchestrates the flow of information between these subsystems, ensuring a coordinated execution of tasks, creating a cohesive and responsive user experience.

Trade-off Studies and Evaluation of Alternatives:
Trade-off studies informed the selection of models for each task. While T5-small excelled in fake news classification, its performance in other tasks was evaluated. DistilBERT emerged as the preferred choice for bias and topic classification, and RoBERTa-large, demonstrated superiority in sentiment analysis.

The deployment utilizes Flask to host the chosen models, establishing a robust backend. The connection between Flask and the mobile application, developed using Flutter, ensures seamless user interaction. Users can initiate the entire classification and summarization process with a single button press, enhancing the accessibility and user-friendliness of the system.

2.3 Implementation

The implementation of the "News Understanding" system involved a systematic and structured approach to ensure the efficient use of tools and technologies. The project utilized a variety of components, including LLMs from Hugging Face, Flask for backend development, Flutter for the frontend, and ngrok for secure tunneling. Here is an overview of the implementation:

2.3.1 Model Selection and Fine-Tuning

Topic classification:
After fine-tuning BERT, DistilBERT, and applying zero-shot classification with BART across three datasets, we observed comparable performance between BERT and DistilBERT, with the latter exhibiting a notable advantage in terms of model size and speed. However, when employing zero-shot classification with BART, a faster model that didn't require fine-tuning, initial results were unsatisfactory when selecting the single highest topic. Yet, a promising improvement emerged when considering the two highest probability topics. Despite these advancements, the overall performance of the zero-shot classification with BART still fell slightly short compared to both BERT and DistilBERT. Considering these findings, we ultimately selected the DistilBERT model fine-tuned on a dataset with six categories as our champion model. Its optimal balance between speed, results, and the number of categories made it the preferred choice for our application.

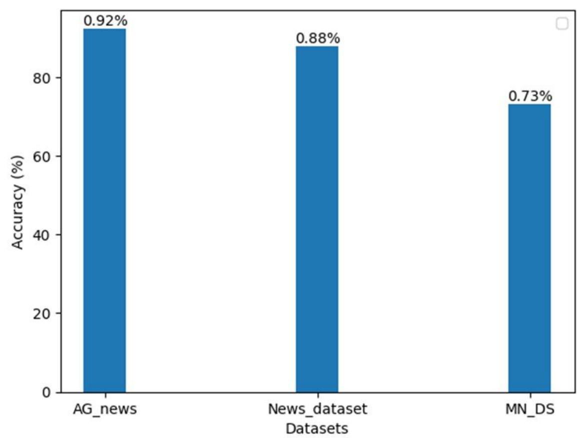

Figure 2: DistilBERT Accuracy for Topic Classification

Bias Classification:
We used datasets from allsides.com and mediabiasfactcheck.com. Initially employing the T5-small model, testing revealed suboptimal accuracy, leading to a transition to the more efficient DistilBERT model. The fine-tuning process involved training DistilBERT on the combined dataset, resulting in a model that accurately recognizes biases in news articles. Rigorous testing on distinct datasets from mediabiasfactcheck.com and allsides.com showcased impressive accuracy rates of 94% and 75%, respectively.

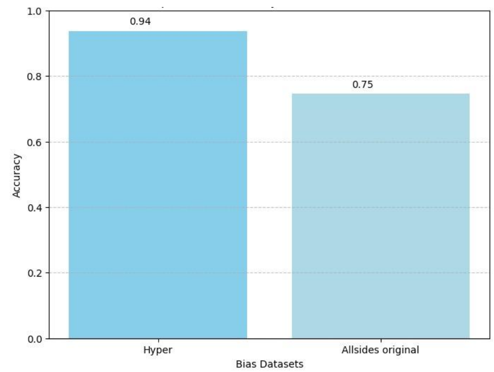

Figure 3: DistilBERT Accuracy for Bias Classification

Sentiment Analysis:
We explored various models, including Roberta-Large, T5-Base, Xlnet-Large, and Distilbert. After careful evaluation, Roberta-Large emerged as the most accurate, achieving the highest accuracy compared to other models. We fine-tuned the selected model, Roberta-Large, with Generated Data 1 and a combination of two Generated Datasets during experimentation. Testing results revealed impressive accuracies, with the model achieving 85% accuracy with Generated Data 1 and 78% accuracy with Generated Data 2.

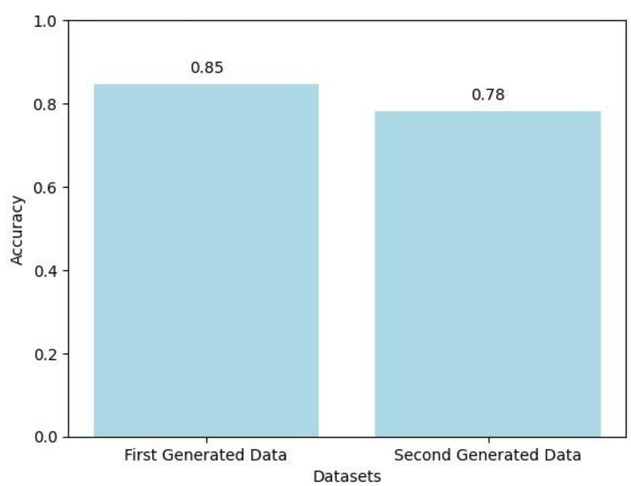

Figure 4: Roberta-Large Accuracy for Sentiment Analysis

Fake vs Real Classification:
We used two different models, Deberta and T5-small, for our classification task. The Deberta model showed poor performance, with accuracies of 60% across various datasets. Then we proceeded to fine-tune the T5-small model on mixed subsets of the data. Upon evaluating its performance on the remaining datasets, the T5-small model showed a significantly superior performance, achieving accuracies ranging from 87% to 98%.

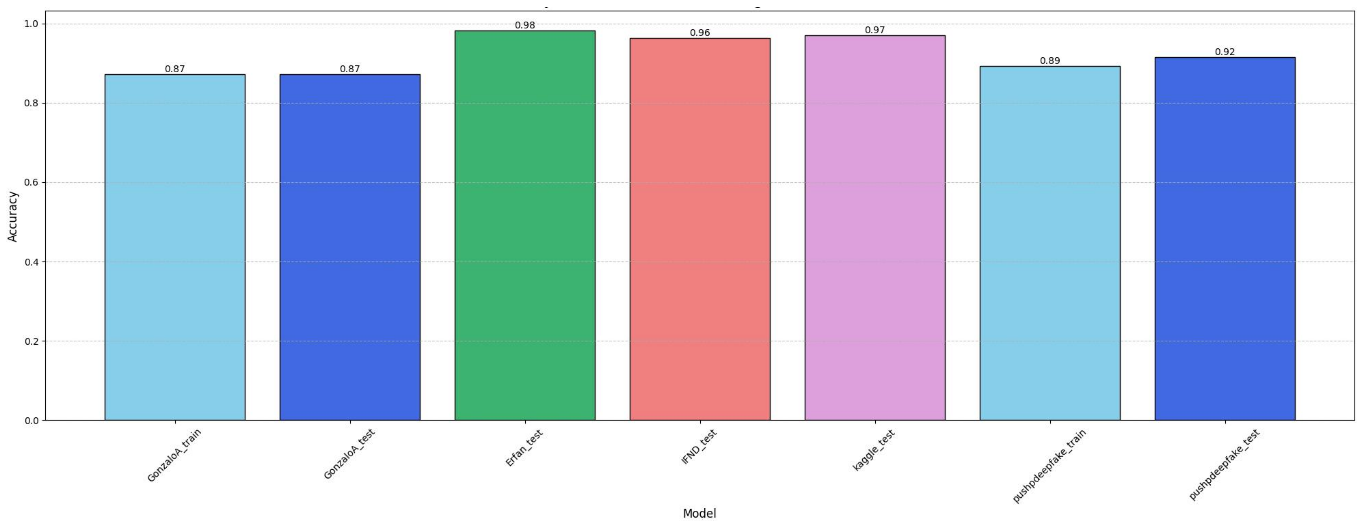

Figure 5: T5-Small Accuracy for Fake Classification

Summarization:
We experimented with T5-small, T5-base, BART, and Pegasus. Employing the BERT score metric for evaluation, BART and Pegasus consistently outperformed the other models, with Pegasus showing a slightly higher mean score across most metrics. Then we conducted a manual evaluation of the generated summaries, ultimately determining that the BART model displayed superior performance.

Metric / Model	Precision	Recall	F1-Score
T5-base	0.895370	0.823968	0.858023
BART	0.883448	0.910341	0.896641
Pegasus	0.902176	0.898644	0.900306

Table 1: Performance Metrics for Summarization Models

2.3.2 Backend Development with Flask

The backend was implemented using Flask, a Python web framework. Flask facilitated the integration of the chosen models and ensured smooth communication between the mobile app and the backend. The backend hosted distinct POST functions for classification and summarization tasks. Each classification task was handled by a specific model, and preprocessing and tokenization processes tailored to each model were implemented. The Flask API orchestrated the flow of information between subsystems, providing a cohesive user experience.

2.3.3 User Interface Design with Flutter

The user interface was designed using Flutter, a UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase. The mobile application featured a simple screen with a text field for entering news articles and two buttons for classification and summarization. The integration between the mobile app and the backend was established through the ngrok-generated API link.

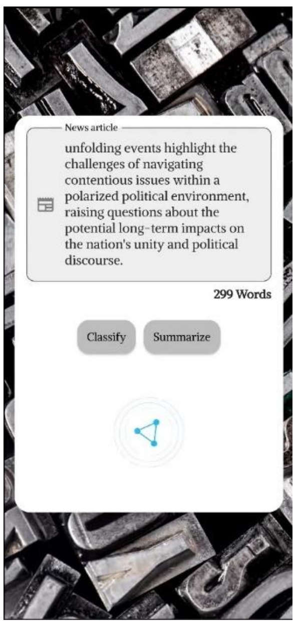

Figure 8: Classification and Summarization Request

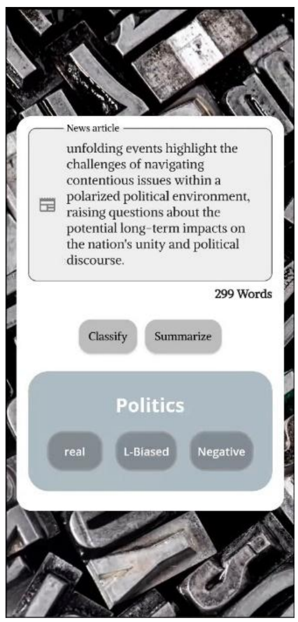

Figure 6: Classification Results

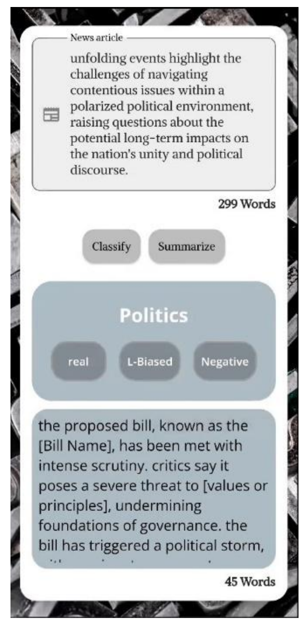

Figure 7: Summarization Results

2.3.4 Deployment and Online Accessibility

The deployment plan involved hosting the Flask API on a local server and using ngrok to make the API accessible online. This ensured that the application could be used on mobile devices, providing online accessibility. The clear and straightforward user interface contributed to a positive end-user experience.

2.4 Testing

2.4.1 Data Plan

Topic Classification:

1- Multilabeled News Dataset (MN_DS)

Dataset from zenodo of 10,917 news articles with hierarchical news categories collected between 1 January 2019 and 31 December 2019. The authors manually labeled the articles based on a hierarchical taxonomy with 17 first level and 109 second-level categories. [8]

2- AG_NEWS Dataset

AG is a collection of more than 1 million news articles. News articles have been gathered from more than 2000 news sources by ComeToMyHead in more than 1 year of activity. The AG's news topic classification dataset is constructed by Xiang Zhang
It consists of 4 categories world, sports, business, and sci/tech.

3- News Dataset

The News Dataset is an English-language dataset containing just over 4k unique news articles scrapped from AriseTv, One of the most popular news television in Nigeria.
It consists of 6 categories business, sports, politics, health, entertainment, and tech.

Bias Classification:

The data plan outlines the comprehensive approach taken for collecting and utilizing datasets to develop and test the bias classification component of the project. The primary goal is to assess news articles for bias or neutrality, categorizing them into three distinct classes: right bias, left bias, and center (neutral). For the first dataset, initial data comprising 21,747 news articles was gathered from AllSides balanced news headline roundups in November 2022, with an additional 4,038 rows acquired through data scraping from November 2022 to November 2023, resulting in a final dataset of 25,700 unique news headlines. The second dataset, Webis Bias Flipper 2018, includes 2,781 events from AllSides spanning June 1st, 2012, to February 10, 2018, with 6,458 news articles collected through crawling. The third dataset, Article Bias Prediction, consists of 37,554 articles stored as JSON files, combined with the second dataset to create a dataset of 47,444 rows after deduplication. The fourth dataset, Hyperpartisan News Detection, contains 750k rows labeled by overall bias, sourced from Hugging Face, with labels provided by BuzzFeed journalists or MediaBiasFactCheck.com. Originally having five labels, the dataset was converted to three for consistency. Additionally, a final dataset containing 45,000 rows without duplicated data was obtained from AllSides, and 100,000 rows were acquired from the data from MediaBiasFactCheck. For training purposes, 35,000 rows from each dataset were used, resulting in a total training set of 70,000 rows.

Sentiment Classification:
In addressing the challenge of limited datasets for sentiment analysis, our approach involved leveraging Knowledge Distillation. This method entails training a smaller model, known as the student model, to mimic the behavior of a larger model. To construct our datasets, we sourced information from two reputable websites. The first dataset, comprising 50,000 rows, was obtained from the MediaBiasFactCheck website. The second dataset, consisting of 40,000 rows, was extracted from the AllSides website. Both datasets were utilized not only for sentiment analysis but also for bias and unbiased classification, as mentioned earlier, focusing specifically on the news content. In the case of the first dataset, we utilized the entire news article, while for the second dataset, we selected the title and the first two paragraphs of the news articles. To label the news content, we employed the Gemini pre-trained model, ensuring robust and reliable dataset preparation for our sentiment analysis and bias classification tasks.

Fake Classification:
we collected diverse datasets from Kaggle and Hugging Face, undertaking thorough preprocessing. To handle duplication, we combined datasets and removed tens of thousands of duplicated articles. Utilizing both train and test sets from various sources, we created a comprehensive dataset. For model training, we employed three distinct train datasets, each containing 20,000 rows, from three different sources. To assess model performance, subsets of 400 rows were randomly selected in a stratified manner from the test datasets. This thorough approach ensures a diverse and representative evaluation of models across different datasets, enhancing the reliability of our fake vs real classification task.

Summarization:
We used a dataset from Hugging Face that contains 1800 rows for training and 425 rows for testing. The dataset contained news articles from BBC along with their corresponding summarizations. To measure the performance of each model, we employed the BERT score metric, comparing the generated summarizations with the reference summarizations provided in the dataset.

2.4.2 Validation & Verification

The validation and verification process were meticulous, aligning with the project's design specifications and ensuring the system's accuracy, adaptability, and adherence to initial requirements.

Fake News Classification:
For the task of fake news classification using T5-small, the expected output was binary (Fake or Real). Testing against various datasets demonstrated high accuracy, validating the model's effectiveness in discerning between authentic and deceptive news.

Topic Classification:
Two models, BART and DistilBERT, were employed for topic classification. BART underwent validation at two levels: identifying the highest probability topic and the two highest probability topics. DistilBERT was chosen as the champion model due to its optimal trade-off between speed, accuracy, and the number of categories in the dataset.

Sentiment Analysis:
XLNet_large and RoBERTa_large were used for sentiment analysis, aiming to categorize news articles as Negative, Positive, or Neutral. The models were fine-tuned with diverse, generative data. RoBERTa_large, with its higher accuracy, emerged as the preferred model.

Bias Classification:
For bias classification (Left-Biased, Right-Biased, Center), DistilBERT was tested against datasets from Allsides and MediaBiasFactCheck, ensuring robust results. The professional journalistic annotation on these sites provided a reliable benchmark, and DistilBERT demonstrated higher accuracy in one dataset and acceptable accuracy in the other, affirming its effectiveness.

End-to-End System Validation:
The validation process extended to the end-to-end system, where requests from the mobile frontend were sent to the backend hosting all classification tasks. The system returned accurate classifications for each task, confirming its seamless integration and functionality.

Summarization:
The summarization task underwent rigorous validation using two criteria: BERT score and manual evaluation. BART emerged as the preferred model for its superior performance, providing concise and accurate summaries.

Additional Validation with ChatGPT:
To further validate the system, news articles generated from ChatGPT containing all labels were utilized. A comparison between the results displayed on the mobile app and those generated by ChatGPT revealed overall alignment, affirming the system's accuracy.

Continuous Testing and Adaptability:
To ensure ongoing adherence to design specifications, continuous testing involved repeated evaluations with updated datasets. The system exhibited adaptability to evolving news articles and maintained consistent performance over time.

3. Overall Results and Analysis

3.1 Project Success and Challenges

The overall experience with the project demonstrated a successful implementation of the NLP-based news classification and summarization system. Despite achieving commendable results, several challenges were encountered throughout the project. Notably, data annotation posed a consistent issue, requiring meticulous effort and time to ensure alignment with diverse criteria for fake and biased news. Additionally, a shortage of data for sentiment classification, mainly centered around finance and stock news, prompted the team to employ GPT-4 and Gemini Pro for data generation. These challenges, however, became valuable learning experiences, showcasing the team's adaptability and problem-solving skills.

3.2 Learning Outcomes and Model Selection

The project significantly contributed to the team's learning outcomes in the graduate program and career objectives. Working with LLMs and transformers, the team gained hands-on experience in prompt engineering and fine-tuning, gaining a deep understanding of model customization for specific classification tasks. The iterative process of model selection emphasized the importance of speed and accuracy, leading to the adoption of the DistilBERT model for two key classification tasks. This practical exposure aligned closely with the team's career aspirations in AI and software development.

3.3 Future Considerations and Project Conclusion

Looking ahead, the project's success lays a foundation for continuous improvement. Future considerations may involve refining models, addressing data challenges, and exploring additional features to enhance the system's capabilities. Despite the challenges faced, the project's end-to-end implementation provided a comprehensive understanding of creating and deploying NLP projects. In conclusion, the experiences gained in this project have been instrumental in expanding the team's expertise, bridging theoretical knowledge from the graduate program with practical skills in the dynamic landscape of NLP and AI.

4. Deployment Plan

4.1 Backend Implementation

For the backend, Flask was employed to streamline the integration of models using Python. The application utilizes two distinct POST functions—one for classification and another for summarization. Each classification task is handled by a different model: DistilBERT for bias and topic classification, RoBERTa-large for sentiment classification, T5-small for fake classification, and BART-large for summarization. To ensure compatibility, preprocessing and tokenization processes tailored to each model were implemented. The classification function accepts text input, runs it through the four models, and returns a list of classified results. The API is hosted on a local server, and ngrok is employed to make the API accessible online, facilitating its use on mobile devices.

4.2 User Interface Design

Moving to the user interface, the system comprises a simple screen featuring a text field for entering news articles. Two buttons are available—one for classification and another for summarization. The integration between the mobile app and the backend is established through the ngrok-generated API link. To initiate classification, users input text and press the classify button, triggering a POST request to the backend. The backend processes the input through the four models, and the results are displayed in three containers representing bias, fake, sentiment classifications, and the overall topic. Additionally, a separate button for summarization prompts the backend to generate a summarized version of the text, with the system displaying both the summary and the count of new words introduced.

This design allows for a seamless interaction where users can easily submit news articles, receive real-time classification results, and access concise summarizations. The use of ngrok ensures online accessibility, enhancing the deployment of the application on mobile devices. The clear and straightforward user interface contributes to a positive end-user experience, allowing users to leverage the NLP models effectively for news classification and summarization.

5. Conclusions and Future Works

In conclusion, our project successfully leverages fine-tuned LLMs, such as Roberta Large for sentiment analysis achieving an accuracy of 85%, DistilBERT for topic classification with an accuracy of 88%, and DistilBERT for determining bias or unbiased news with an impressive accuracy of 95%. For the critical task of discerning fake news, we employ the T5 Small model, attaining a commendable accuracy of 97%. Furthermore, our summarization module utilizes BART, incorporating a high BERT Score metric. To ensure a seamless user experience, we employ Flutter for our frontend and Flask for our backend, combining powerful language models with an intuitive interface for effective news understanding and analysis.

In our future endeavors for the "News Understanding" project, we aim to expand our classifier repertoire by incorporating features such as distinguishing between breaking and non-breaking news. Additionally, we plan to implement advanced Named Entity Recognition to extract crucial details from articles, encompassing what, when, where, why, and how aspects of news events. To enhance user accessibility, we aspire to develop a browser extension enabling users to effortlessly assess the authenticity, topic, and sentiment of selected news text. Our objectives also include evaluating news reporting for bias or impartiality, along with implementing a summarization feature. Furthermore, recognizing the importance of user experience, we will prioritize the enhancement of our mobile application's interface to ensure a seamless and intuitive interaction for our users.

6. References

[1] Essa, Ehab, et al. “Fake News Detection Based on a Hybrid BERT and LightGBM Models.” Complex & Intelligent Systems., vol. 9, no. 6, 2023, pp. 6581–92, https://doi.org/10.1007/s40747-023-01098-0.

[2] Alsini, R. (2023). Analysis of Real Time Twitter Sentiments using Deep Learning Models. Journal of Applied Data Sciences, 4(4), 480-489. Doi: https://doi.org/10.47738/jads.v4i4.146

[3] Ansary, Md Siam. “Newspaper Article Summarization Using Combinational Method.” 2023 International Conference on Power, Instrumentation, Control and Computing (PICC), IEEE, 2023, pp. 1–4, https://doi.org/10.1109/PICC57976.2023.10142440.

[4] https://huggingface.co/

[5] https://github.com/amaiya/ktrain/tree/master

[6] https://www.gemini.com/

[7] https://openai.com/gpt-4

[8] Petukhova, A.; Fachada, N. MN-DS: A Multilabeled News Dataset for News Articles Hierarchical Classification. Data 2023, 8, 74. https://doi.org/10.3390/data805007