Progress Report 2

Propaganda Detection Application

Team Members:

Aly Farrag Elmandouh

Amira Ali Mohamed

Aya Adel Saied

Overview:

Our application is a tool designed to help users detect propaganda in textual content. The application uses two different AI models: a binary classification model to detect whether a given text contains propaganda and a multilabel classification model to identify specific persuasion techniques if propaganda is detected. The application is built using Tkinter for the GUI interface and the Hugging Face Transformers library for natural language processing.

Progress:

1. Initial Development:

GUI Interface Setup:

- Implemented a Tkinter interface with options for users to input text and select the model type (binary or multilabel).
- Added labels, buttons, and a text input field to allow user interaction with the application.

Model Integration:

- Integrated our transformers models using the Hugging Face Transformers library:

1. A binary classification model for detecting propaganda.
2. A multilabel classification model for identifying specific persuasion techniques.

- Successfully loaded and tested both models in the application.

2. Key Features Implemented:

Text Input and Analysis:

Users can input a paragraph of text directly into the application, and the selected model will analyze the content to determine the presence of propaganda or specific persuasion techniques.

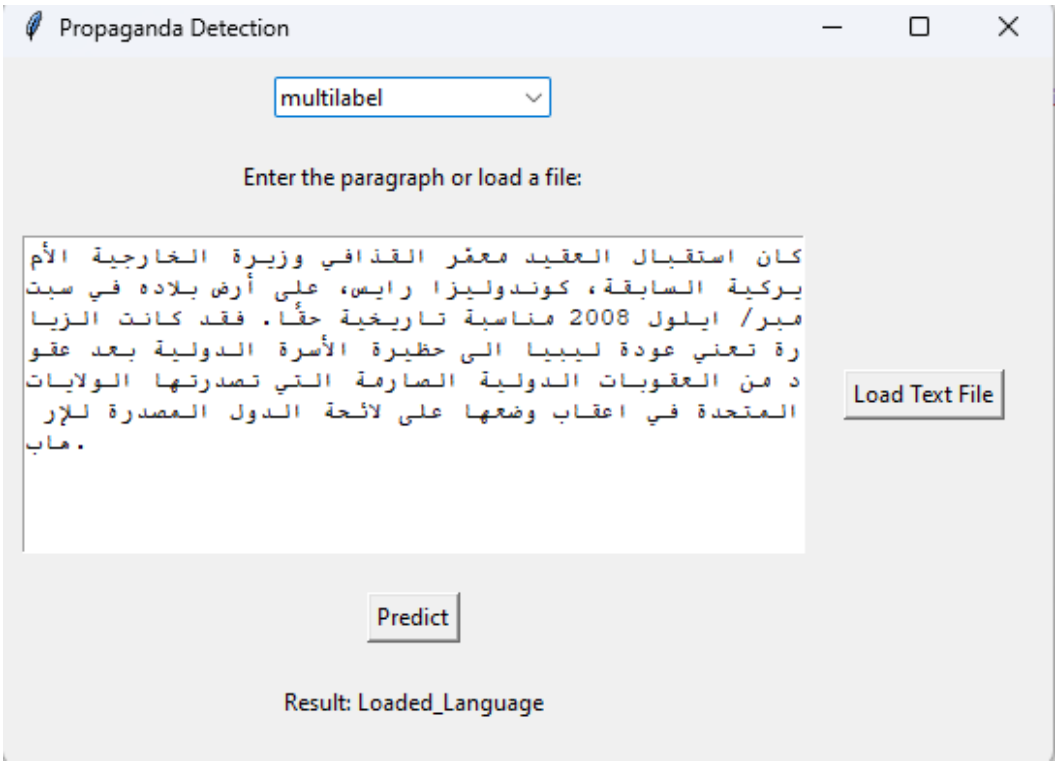
Prediction Logic:

Implemented logic to display the predicted labels, including handling the "no technique" case, ensuring that if no techniques are detected, this is clearly communicated to the user.

File Upload Feature:

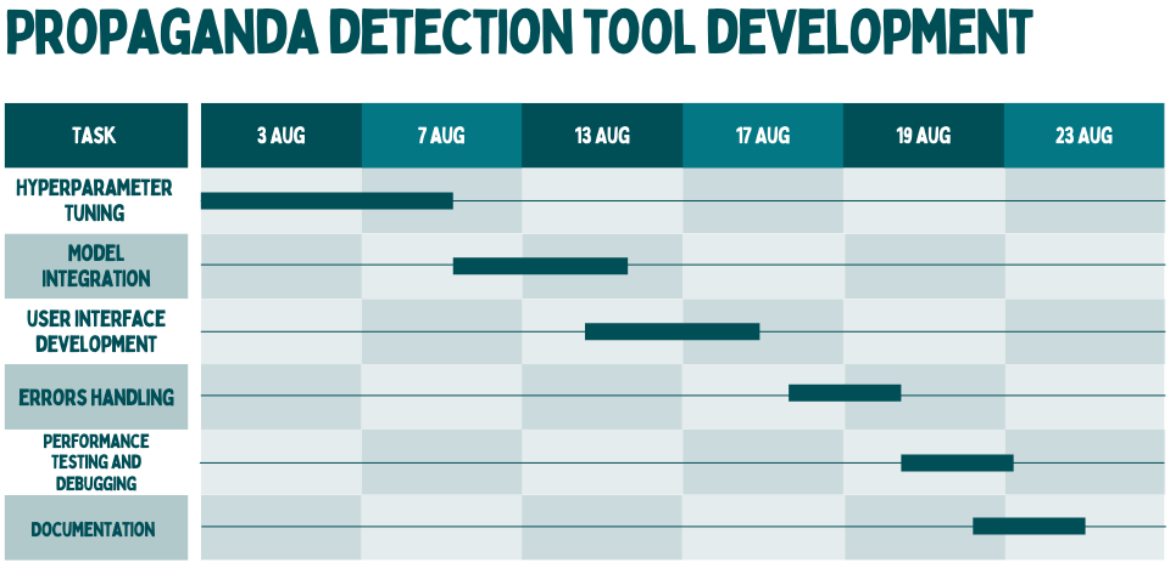
Added a feature allowing users to upload a .txt file for analysis. The content of the file is read and displayed in the text input field, enabling further analysis by the user.

Propaganda detection tool interface



3. Bug Fixes and Improvements:

- Resolved an issue where the application was incorrectly displaying "no technique" alongside other detected techniques. Adjusted the prediction logic to ensure "no technique" is only shown when it is the sole prediction.
- Improved the handling of model predictions by refining the threshold for label detection, which enhanced the overall performance of the multilabel model.



Challenges Encountered

1. **Model Performance:**

- The multilabel model sometimes struggles to accurately detect multiple persuasion techniques, particularly when the "no technique" label is prevalent. This required fine-tuning the prediction threshold and logic to ensure meaningful results.

2. **Balancing Accuracy and Usability:**

- Striking a balance between model accuracy and usability within the application was challenging, especially given the low overall accuracy of predictions. This happened due to the unbalancing in number labels.

3. **Hyperparameter Tuning with Optuna:**

- We attempted to optimize the model's hyperparameters using Optuna, an advanced hyperparameter optimization framework. However, due to the extensive training times and lacking the computational resources the tuning process was not completed, and as a result, the tuned model could not be integrated into the final application.

Next Steps:

Feature Enhancements:

- Add functionality to save analysis results to a file for future reference.
- Implement support for of multiple text files.

User Interface Improvements:

- Enhance the GUI by adding more intuitive controls and improving the layout.
- Consider adding tooltips and help documentation within the application to guide users.