Final Project: Build a Transformer-Based Fake News Detection Chatbot

In this project, you are required to design and deploy an intelligent chatbot that detects fake news using a transformer-based model. The primary aim is to apply your skills in data preprocessing, model integration with a transformer model, and deployment to create a modern AI-driven conversational system for fake news detection.

Objective

The chatbot should process user-input news text and predict whether it is fake or true based on provided datasets (`true.csv` and `fake.csv`). The system will serve as a practical application of transformer models to classify news and deliver results conversationally.

Data

CSV files: `true.csv` and `fake.csv`, containing true and fake news articles, respectively. Unlike the previous assignment, which relied on keyword-based labeling without contextual information, this project requires preprocessing the data to create meaningful embeddings using a transformer model for classification.

Data Preprocessing

- Combine the `true.csv` and `fake.csv` datasets into a single dataset with labels (e.g., 1 for true, 0 for fake).
- Clean the text data by removing special characters, punctuation, and stopwords, and perform tokenization suitable for the selected transformer model.
- Ensure the dataset is balanced or address any imbalances during model training.
- Split the dataset into training, validation, and test sets (e.g., 80-10-10 split).
- For less model run time, use only top 100 rows of data from each csv file.(this is not mandatory but recommended)

Model Selection and Integration

You could use an open-source transformer model from Hugging Face's `transformers` library. For this project, we recommend using pre-trained models such as BERT (`bert-base-uncased`) or DistilBERT (`distilbert-base-uncased`), as they are well-suited for text classification tasks like fake news detection and are computationally efficient for training

on Kaggle's T4 free GPU. BERT provides robust performance, while DistilBERT offers a lighter, faster alternative with comparable accuracy. You may use the model in an inference-only manner after fine-tuning by embedding the input text and generating predictions dynamically. We are open to innovative approaches, such as experimenting with model configurations or optimization techniques.

Steps for Model Integration

1. Load the pre-trained BERT or DistilBERT model and tokenizer from Hugging Face.
2. Preprocess the text data to create input embeddings compatible with the selected model.
3. Fine-tune the model on the labeled dataset for binary classification (fake vs. true news).
4. Save the trained model for use in the chatbot.

Chatbot Development

Develop a chatbot interface capable of processing a single user-input news article or text snippet and generating a prediction (fake or true) with a confidence score. The chatbot should use the fine-tuned BERT or DistilBERT model to classify the input and return a conversational response (e.g., "This news is likely fake with a confidence of 85%").

Deployment with Streamlit

Deploy the chatbot using Streamlit. The interface should include:

- A text input box for users to paste or type a news article/snippet.
- An area to display the model's prediction (fake or true) and confidence score.

Video Demonstration

Record a short video (under 1 minute) demonstrating your chatbot in action. The video should clearly show:

- The Streamlit interface.
- A few example inputs (e.g., a fake news snippet and a true news snippet).
- The corresponding predictions and confidence scores generated by the chatbot.

Deliverables and Evaluation

Your submission must be pushed to your GitHub repository and include:

- Source code, including data preprocessing, model training, and chatbot implementation.
- The processed dataset (or scripts to preprocess the provided CSVs).

- Upload your streamlit deploy link to your github repository.

- A README file explaining how to run the project, including dependencies and instructions for running the Streamlit app locally or accessing a public/shareable link.
- A video file (max 1 minute) demonstrating the chatbot functionality.

Evaluation will be based on:

- Effectiveness of data preprocessing and model training.
- Accuracy and relevance of the chatbot's predictions.
- Quality and usability of the Streamlit interface.
- Completeness of deployment and adherence to submission guidelines.

Deadline

The deadline to submit the project is 6th July 2025.