Technical Report

1. Model Architecture

- **Sentiment Analysis Module (Fine-Tuned LLM):**
  - **Model:** A SimpleRNN model fine-tuned on the IMDB movie review dataset.
  - **Custom Component:** A custom SimpleRNN class (which removes the unsupported time_major parameter) is used to load the pre-trained model from simple_rnn_imdb.h5.
  - **Pipeline:** Input reviews are preprocessed (lowercased, tokenized, and padded to a fixed length of 500 tokens) before being fed into the model to predict sentiment.
- **Text Generation Explanation (RAG-Inspired):**
  - **Model:** GPT-2 from Hugging Face is used via the Transformers pipeline to generate explanations based on the provided movie review.
  - **Prompting Strategy:** The prompt is structured as "Review: [user review] \nSentiment:" so that GPT-2 generates a sentiment explanation.
  - **Generation Parameters:** The pipeline is configured with parameters (max_length, temperature, top_p, top_k, no_repeat_ngram_size, repetition_penalty) to control output length and reduce repetition.
- **Embedding Demo Module:**
  - **Approach:** Demonstrates one-hot encoding of a sample sentence followed by padding and embedding conversion using a simple Keras Embedding layer.
  - **Model:** A minimal Sequential model is built to convert one-hot encoded inputs into dense embedding vectors.

---

2. Dataset & Preprocessing

- **Dataset:**
  - The IMDB movie review dataset is used for training and testing the sentiment analysis model.
- **Preprocessing for Sentiment Analysis:**
  - **Text Normalization:** Reviews are converted to lowercase and split into words.

- **Encoding:** Words are encoded using the IMDb word index. Unknown words are assigned a default index (2), and indices are shifted by 3 (to account for reserved tokens).
- **Padding:** Encoded reviews are padded to a maximum length of 500 tokens.

- **Preprocessing for Embedding Demo:**
    - **One-Hot Encoding:** Sample sentences are converted into one-hot representations using a specified vocabulary size (10,000).
    - **Padding:** Sequences are padded to a fixed length (e.g., 8 tokens) before being processed by the embedding layer.

---

3. Design Decisions

- **Model Selection:**
    - The SimpleRNN model was selected for its simplicity and effective performance on the IMDB dataset, making it suitable for a domain-specific task.
    - GPT-2 was chosen for text generation because it can produce coherent explanations based on prompts even without fine-tuning on sentiment explanations.
- **Frameworks and Libraries:**
    - **TensorFlow/Keras:** Used for building and training the sentiment analysis and embedding models.
    - **Hugging Face Transformers:** Used for leveraging GPT-2 for text generation.
    - **Streamlit:** Selected as the web framework to build an interactive multi-page application that integrates all modules.
- **Integration Techniques:**
    - The project uses modular functions for each component (sentiment analysis, text generation, embedding demo) and integrates them into a single Streamlit app (app.py).
    - New caching decorators (st.cache_resource for resource-intensive objects and st.cache_data for serializable data) are implemented to optimize performance and resource management.

---

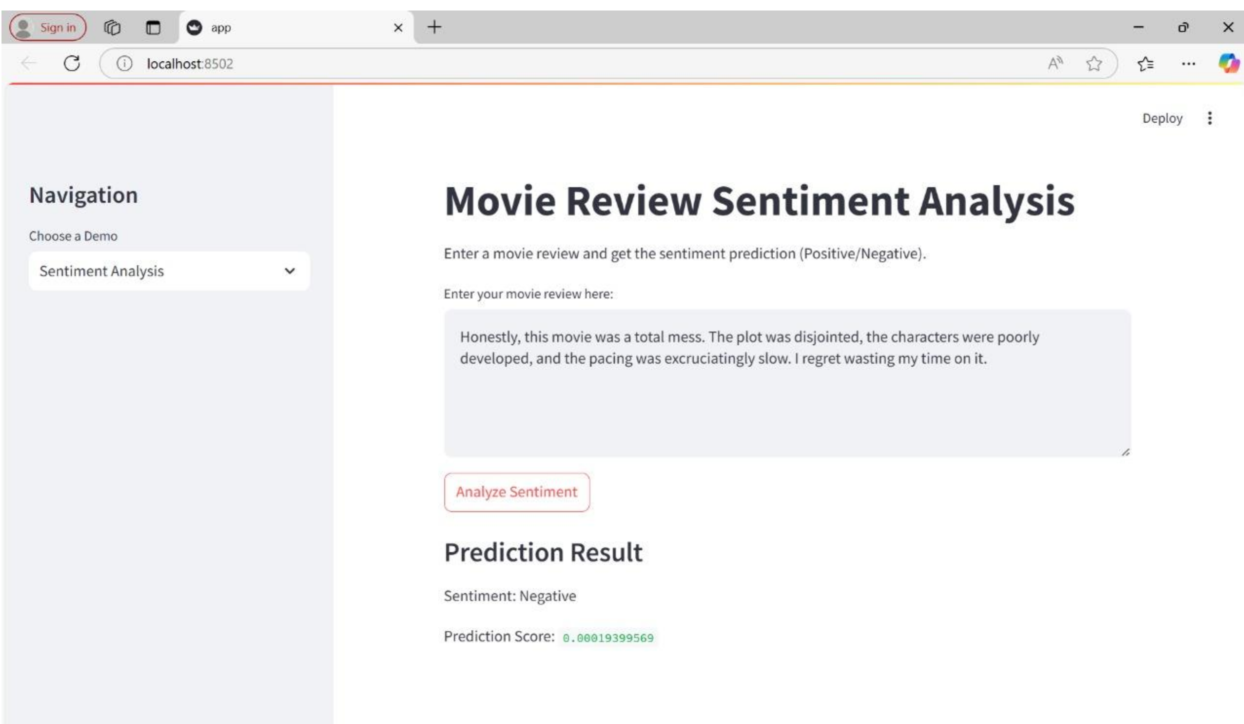
4. Performance Evaluation

- **Quantitative Evaluation:**
  - The SimpleRNN model, as shown in the training notebooks (simplernn.ipynb and prediction.ipynb), achieved a validation accuracy of approximately 81%.
  - The prediction function uses a threshold (set to 0.7 in the app) to classify reviews as Positive or Negative.
- **Qualitative Evaluation:**
  - **Sentiment Analysis:** For example, a review like "This movie was fantastic!" yields a prediction score (e.g., 0.81) and is classified as Positive.
  - **Text Generation:** The GPT-2-based module produces explanations that provide rationale for the predicted sentiment, though further fine-tuning may improve consistency.
  - **Embedding Demo:** The one-hot encoded inputs are transformed into dense embeddings and displayed in a clear tabulated format via Streamlit.

Deployment Link:

Streamlit

Results:



Deploy

Navigation

Choose a Demo

Text Generation Explanation

Text Generation Explanation (RAG-inspired)

Enter a movie review to generate a text explanation using GPT-2.

Enter your movie review here:

Honestly, this movie was a total mess. The plot was disjointed, the characters were poorly developed, and the pacing was excruciatingly slow. I regret wasting my time on it.

Generate Explanation

Generated Explanation

Review: Honestly, this movie was a total mess. The plot was disjointed, the characters were poorly developed, and the pacing was excruciatingly slow. I regret wasting my time on it. Sentiment: It's not that bad of an idea (though maybe some things can be better), but you never know what might happen next!

Sign in app

localhost:8502

Deploy

Navigation

Choose a Demo

Embedding Demo

Word Embedding Demo

Enter a sentence to view its one-hot padded representation and embedding vectors.

Enter a sentence

Honestly, this movie was a total mess. The plot was disjointed, the characters were poorly developed, ar

Show Embedding

Padded Sequence (One-hot Indices)

0	1	2	3	4	5	6	7
4,008	5,953	2,297	5,888	8,108	3,565	2,232	4,763

Embedding Vectors

0	1	2	3	4	5	6	7	8	9
-0.0408	-0.0468	0.0483	0.0107	-0.0385	0.0091	0.0272	0.0126	0.0068	0.0191
-0.0261	0.0337	-0.0319	0.0034	-0.0064	-0.0212	-0.0012	0.0009	-0.0365	0.0021
-0.0023	-0.0252	0.0337	0.0211	0.0398	0.0112	-0.0095	-0.0083	-0.0332	0.0112
-0.0397	-0.0149	0.0245	0.015	0.04	0.0443	-0.0217	-0.0169	-0.0327	-0.002
-0.0304	0.0078	-0.0462	0.0497	-0.0009	0.0441	0.0134	0.0188	-0.0284	-0.0051
0.0378	-0.0247	-0.0163	0.0345	-0.0055	-0.048	-0.0236	-0.0413	0.0079	0.0257
-0.0288	0.0443	-0.0443	-0.049	0.0476	0.0126	-0.0277	0.042	0.0176	-0.0261
0.0281	-0.0023	-0.004	-0.0275	-0.0337	0.0061	0.0066	-0.0204	0.0082	-0.0215