LLMs Assignment 2

Ruweida Muzamil

Q1:

1. Project Overview

- **Project Title:**Text Generation for SDGs 4 Education
- **Objective:** To develop a text generation application using the Bloomy model to assist in generating content related to Sustainable Development Goals (SDGs).
- **Scope:** The application will focus on generating text based on user-provided prompts related to SDGs.

2. Model Selection

- **Rationale:** Explain why the Bloom model was chosen, considering as it's widely accessible and open-source.
- **Model Details:** Provide information about the specific Falcon model used (e.g., 7B, 40B).

3. User Interface Development

- **Framework:** Gradio
- **Components:** Describe the key components of the user interface, including the prompt input field, text generation button, and output display.
- **Functionality:** Explain how the user interface interacts with the Bloom model to generate text.

4. Model Integration

- **Loading:** Describe the steps involved in loading the Bloom model into the application.
- **Text Generation:** Explain how the model's `generate()` method is used to generate text based on user-provided prompts.
- **Parameters:** Discuss any relevant parameters used in the `generate()` method (e.g., `max_length`, `num_beams`).

5. Evaluation

- **Prompts:** List the prompts used to evaluate the model's performance.
- **Evaluation Criteria:** Describe the criteria used to assess the generated text (e.g., coherence, creativity, relevance, grammatical correctness).
- **Results:** Summarize the evaluation results, including quantitative and qualitative metrics.
- **Discussion:** Analyze the strengths and weaknesses of the generated text based on the evaluation results.

6. Conclusion

- **Summary:** Recap the key findings and achievements of the project.
- **Future Work:** Discuss potential areas for improvement or future research, such as exploring different models or expanding the application's capabilities.

Q2:

Natural Language Processing Project Documentation

Introduction

This project explores the use of different text embeddings—**Word2Vec**, **GloVe**, and **BERT**—for a text classification task. The aim is to understand how these embeddings impact the performance of classifiers in categorizing text data related to Sustainable Development Goals (SDGs), specifically using a dataset from Kaggle. The classification task will evaluate the embeddings based on metrics such as accuracy, precision, recall, and F1-score.

Objectives

1. Investigate different types of text embeddings: Word2Vec, GloVe, and BERT.
2. Apply these embeddings to a text classification task (e.g., sentiment analysis, topic categorization).
3. Compare the performance of the various embedding types using evaluation metrics.
4. Document the findings and provide a detailed analysis of each embedding type's strengths and weaknesses.

Dataset

- **Source**: An SDG-related dataset downloaded from Kaggle.

- **Description**: The dataset includes text samples labeled with different SDG-related categories, suitable for classification tasks.
- **Preprocessing**: The text data is cleaned by removing non-alphanumeric characters, converting to lowercase, and tokenizing the sentences.

Text Embeddings Explored

1. **Word2Vec**: A shallow neural network-based embedding technique that represents words in a continuous vector space based on their context within a fixed-size window.
2. **GloVe (Global Vectors for Word Representation)**: An unsupervised learning algorithm for obtaining vector representations for words. GloVe maps words into a meaningful space where the distance between words is related to semantic similarity.
3. **BERT (Bidirectional Encoder Representations from Transformers)**: A transformer-based model that provides contextual embeddings, capturing word meaning based on the surrounding context.

Conclusion

- BERT outperforms Word2Vec and GloVe on most metrics, demonstrating the value of contextualized embeddings in text classification tasks.
- Word2Vec and GloVe, while faster and less resource-intensive, do not match the performance of BERT in tasks that require nuanced understanding of language