Assignment 2

Shaima Abdi

Question1:

SDG 13 Text Generation Application

This project is a text generation application that uses the Bloom model to generate responses related to Sustainable Development Goals (SDGs), specifically SDG 13: Climate Action. The application allows users to input prompts and receive generated text responses that continue or elaborate on the given input.

Introduction

The SDG Text Generation Application aims to provide users with insights and information related to SDG 13: Climate Action by leveraging open-source generative AI models. The application uses the Bloom model to generate relevant, coherent, and creative text based on user prompts.

Example Prompts

- "What are effective ways to reduce carbon emissions?"
- "How can renewable energy help combat climate change?"

Model Evaluation

I evaluated the model using various prompts specific to climate action to ensure it generates responses that are coherent, creative, relevant, and grammatically correct.

Question2:

NLP Text Classification Project

Introduction

In this revised approach, we will use different methods to generate and evaluate text embeddings for a text classification task. This approach will focus on implementing different models and techniques for embedding extraction and model evaluation.

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 various embedding types using evaluation metrics such as accuracy, precision, recall, and F1-score.
4. Document the findings, including insights into the strengths and weaknesses of each embedding type.

Dataset

- **Source**: SDG-related dataset downloaded from Kaggle.
- **Description**: Contains text samples labeled with different SDG-related categories.
- **Preprocessing**: Clean the dataset by removing non-alphanumeric characters, converting text to lowercase, and tokenizing the sentences.

Steps to Complete the Project

Step 1: Data Preprocessing and Exploratory Data Analysis (EDA)

- **Data Cleaning**: Load the dataset and clean the text by removing special characters, numbers, converting text to lowercase, and tokenizing the text.
- **Exploratory Data Analysis (EDA)**: Perform basic EDA to understand the distribution of categories, check for class imbalance, and visualize text length.

Step 2: Splitting the Dataset

- Split the dataset into training and testing sets using an 80-20 ratio.

Step 3: Text Embedding and Feature Extraction

- **Word2Vec**: Use the pre-trained Word2Vec model from `gensim` to generate word vectors for each word. Compute sentence embeddings by averaging the word vectors.

GloVe: Use pre-trained GloVe embeddings (e.g., `glove.6B.100d.txt`). For words not in the GloVe vocabulary, use a zero vector.

Step 4: Train and Evaluate the Classifiers

- Train a machine learning model (e.g., Logistic Regression or RandomForest) using the different embeddings as features. Evaluate the model using standard metrics.

Conclusion

- **BERT** outperformed other embeddings due to its ability to capture contextual information. It is especially useful for tasks where understanding context is crucial.
- **Word2Vec** and **GloVe** are suitable for simpler tasks or when computational resources are limited.