

Assignment 2 Report

Student Name: Faisal Popalzai

Date: August 31, 2024

Introduction

This report outlines the development of a text generation application using open-source generative AI models and a natural language processing project focused on text embeddings and classification tasks. Additionally, an optional semantic search engine was implemented to demonstrate the practical application of text embeddings.

Q1: Text Generation Application

1. Model Selection and Setup

For the text generation application, the BLOOM model from Hugging Face was chosen due to its effectiveness in generating coherent and contextually relevant text. The model and tokenizer were loaded using the transformers library.

2. User Interface

A user-friendly interface was developed using streamlit to allow users to input prompts related to the Sustainable Development Goals (SDGs). The model generates text continuations based on these prompts, showcasing its creativity and relevance.

3. Performance Evaluation

The generated texts were evaluated based on coherence, creativity, relevance, and grammatical correctness. The BLOOM model performed well across these criteria, providing outputs that were

Assignment 2 Report

both creative and relevant to the given prompts.

Q2: Natural Language Processing Project

Part 1: Explore and Compare Text Embeddings

1. Embedding Types

Three types of text embeddings were explored: Word2Vec, GloVe, and BERT. Each embedding type was applied to a text classification task using a dataset related to the 20 Newsgroups.

2. Text Classification Task

A text classification task was performed using the embeddings to classify documents into categories such as "rec.autos" and "sci.space." The embeddings were evaluated using accuracy, precision, recall, and F1-score metrics.

3. Comparison of Embeddings

- Word2Vec: Fast to train and requires less computational resources, but lacks contextual understanding.
- GloVe: Similar to Word2Vec but provides slightly better results in our classification task.
- BERT: Provides the best performance due to its ability to understand context, but is more resource-intensive.

Part 2: Build a Semantic Search Engine (Optional Bonus)

A semantic search engine was built using the FAISS library. This engine leverages text embeddings to search and retrieve relevant documents based on user queries. Case studies demonstrated the

Assignment 2 Report

engine's effectiveness in returning accurate results for various queries.

Conclusion

This assignment successfully demonstrated the capabilities of generative AI models and text embeddings in practical applications. The text generation app provided valuable insights into model performance across different prompts, while the NLP project highlighted the strengths and weaknesses of various embeddings. The optional semantic search engine showcased a real-world application of these technologies.

References

- Hugging Face Transformers Library
- FAISS Library
- 20 Newsgroups Dataset