**Project Documentation: Impression Generation Using LLM**

**Author:** Karthikchidambaram Annamalai  
**Registration Number:** RA2111003010784  
**Affiliation:** 5C Network  
**Date:** 01/10/2024

**Table of Contents**

|  |
| --- |
| 1. [**Introduction**](#introduction) |
| 1. [**Objectives**](#objectives) |
| 1. [**Methodology**](#methodology) |
| * 1. [**Model Selection**](#model-selection) |
| * 1. [**Dataset**](#dataset) |
| * 1. [**Preprocessing**](#preprocessing) |
| * 1. [**Model Fine-tuning**](#model-fine-tuning) |
| * 1. [**Evaluation**](#evaluation) |
| * 1. [**Text Analysis**](#text-analysis) |
| * 1. [**Visualization**](#visualization) |
| 1. [**Results**](#results) |
| 1. [**Challenges and Improvements**](#challenges-and-improvements) |
| 1. [**Conclusion**](#conclusion) |
| 1. [**References**](#references) |

**1. Introduction**

The project aims to leverage Large Language Models (LLMs) to generate impressions based on the input reports provided in the dataset. This is crucial for enhancing automated report generation and improving operational efficiency within organizations like 5C Network.

**2. Objectives**

* Fine-tune a pre-trained LLM to generate impressions from a dataset containing reports.
* Evaluate the model's performance using perplexity and ROUGE scores.
* Conduct text analysis to preprocess the data, removing stop words and applying stemming and lemmatization.
* Visualize the top word pairs based on similarity scores derived from the processed text.

**3. Methodology**

**3.1 Model Selection**

For this project, the **Gemma-7b-it** model was selected based on its capabilities to generate high-quality text. Given the available hardware resources, this model was deemed suitable for fine-tuning tasks.

**3.2 Dataset**

The dataset used in this project consists of 330 reports, including fields such as **Report Name**, **History**, **Observation**, and the corresponding **Impression** to be generated.

**3.3 Preprocessing**

Data preprocessing is essential for preparing the input text for the model. The following steps were taken:

* Combined **Report Name**, **History**, and **Observation** into a single input string.
* Tokenized the input and target sentences using the appropriate tokenizer for the selected model.
* Padded and truncated the sequences to a maximum length of 128 tokens.

**3.4 Model Fine-tuning**

The model was fine-tuned using the Hugging Face Trainer API with the following parameters:

python

Copy code

training\_args = TrainingArguments(

output\_dir="./results",

per\_device\_train\_batch\_size=1,

per\_device\_eval\_batch\_size=1,

num\_train\_epochs=3,

learning\_rate=2e-5,

weight\_decay=0.01,

evaluation\_strategy="epoch",

save\_strategy="epoch",

load\_best\_model\_at\_end=True,

push\_to\_hub=False,

)

The fine-tuning process involved using a training set of 300 samples (250 for training, 30 for evaluation).

**3.5 Evaluation**

The model's performance was evaluated using the following metrics:

* **Perplexity:** Measures the model's uncertainty in predicting the next word.
* **ROUGE Score:** Evaluates the quality of generated impressions against reference texts.

**3.6 Text Analysis**

Text analysis was conducted on the entire dataset to derive insights and prepare embeddings for further exploration:

* Removed stop words using NLTK.
* Applied stemming and lemmatization to the remaining words.
* Generated embeddings using the Word2Vec model.

**3.7 Visualization**

The results of the text analysis were visualized to identify the top 100 word pairs based on their similarity. This involved creating scatter plots using Matplotlib.

python

Copy code

plt.figure(figsize=(10, 8))

plt.scatter(x, y)

for i, word in enumerate(word\_pairs):

plt.annotate(word[0], (x[i], y[i]))

plt.title('Top Word Pairs Based on Similarity')

plt.xlabel('Dimension 1')

plt.ylabel('Dimension 2')

plt.grid()

plt.show()

**4. Results**

The results of the evaluation metrics are summarized as follows:

* **Perplexity:** [Insert calculated perplexity]
* **ROUGE Score:** [Insert calculated ROUGE score]

Visualizations of the top word pairs identified during the analysis are included in this section.

**5. Challenges and Improvements**

Some challenges encountered during the project included:

* Model convergence issues due to limited training data.
* Difficulties in fine-tuning the model to generate coherent and contextually accurate impressions.

Potential improvements include increasing the training dataset size and experimenting with different model architectures or hyperparameters.

**6. Conclusion**

The project successfully demonstrated the capability of LLMs in generating impressions based on a structured dataset. The evaluation metrics indicate that the model performs adequately, and the text analysis provided valuable insights into the dataset.

**7. References**

* Hugging Face Transformers Documentation
* [NLTK Documentation](https://www.nltk.org/)
* Word2Vec Documentation