Prompts for Text Generation and Fine-Tuning

Task 1 : Translation of Code-Mixed Text to Original Language using LlaMa

Purpose :

To translate code-mixed text into a single target language while ensuring accuracy, contextual appropriateness, and preserving the original meaning.

Model: **meta-llama/Llama-3.1-8B**

Prompt :

```
“””
Translate the following text from code-mixed {source_language} to {target_language}.
Ensure the translation is valid, accurate, and remains in {target_language}.
Only provide the translation without any additional text.
Text: {input_text}
Translation: {generated output}
“””
```

Explanation :

This prompt directs the model to translate code-mixed text into a single, consistent language while maintaining accuracy. Instructing the model only to output the translated text without additional comments ensures a clean, straightforward result. The emphasis on validity and accuracy helps preserve the original context and meaning of the mixed-language content, ensuring the translation is coherent and faithful to the source. This approach provides a reliable translation of complex, code-mixed text.

Task 2 : Translation of Code-Mixed Text to Original Language using GPT

Purpose :

To translate code-mixed text into a single target language while ensuring accuracy, contextual appropriateness, and preserving the original meaning.

Model : **gpt-3.5-turbo-instruct**

Prompt :

```
“””
```

Translate the following {source_language} code-mixed sentence to {target_language}.
Ensure the translation is valid and in proper {target_language} Unicode characters :
Text: {input_text}
Translation: {generated output}
“””

Explanation:

This prompt directs the model to translate code-mixed text into a single target language while ensuring the translation is accurate and represented in proper Unicode characters. The focus on Unicode guarantees the output is displayed correctly, particularly for non-Latin scripts like Hindi. This ensures accurate, culturally appropriate, and readable translations that maintain the original meaning.

Task 3 : Fine Tuning - GPT models, LlaMa Models for predicting output label

Purpose :

For fine-tuning, we provide the model with sentence and completion pairs to help it learn from examples and generate accurate outputs.

Depending on the dataset, Offensive speech prediction tasks may use labels like Toxic/Not Toxic or Offensive/Not Offensive, and the prompt is adjusted for the language in use. Sentiment analysis typically uses labels such as positive, negative, and neutral to identify emotional tones.

The choice of labels and categories depends on the goals of the fine-tuning task and the dataset used for training.

Models : **GPT davinci-002 , GPT curie-001**

Prompts :

1. **Sentiment Analysis**

   Fine-tuning training Prompt :

   Input : “ Input: {input text}”

   Completion : “Sentiment: {label}”

   Fine-tuning testing Prompt :

   Input: “Input : {text}”

   Generated Output: “Sentiment : <generated encoded sentiment label>”

Examples (Sentiment Analysis: Hindi-English):

1. **Input**: The concert was amazing, sab log bohot excited the.
   Sentiment : 2 (Positive)
2. **Input**: The service was pathetic, bilkul bhi achi nahi thi aur staff rude tha.
   Sentiment : 0 (Negative)
3. **Input**: The weather is okay, zyada garmi nahi hai but thoda humid lag raha hai.
   Sentiment : 1 (Neutral)

2. **Offensive speech prediction**

Fine-tuning training Prompt :

Input : “Input: {input text}”

Completion : “Label : {label}”

Fine-tuning testing Prompt :

Input:“Input : {text}”

Generated Output:“Label : <generated encoded label>”

Examples (Offensive speech prediction: Hindi-English):

1. **Input**: The day was fine, subah walk pe gaya aur sab peaceful tha.
   Label: 0 (Non-Toxic)
2. **Input**: Yeh bohot bekar idea hai, samajh nahi aata kisne yeh karne ko kaha.
   Label: 1 (Toxic)

Model : Llama 3.1-8B

Prompts :

1. **Sentiment Analysis**

Fine-tuning training Prompt :

Input: “””

Classify the text into Positive, Negative, Neutral, and return the answer as the corresponding sentiment.

text: {tweet}

label: {sentiment}

“”””

Fine-tuning Testing Prompt :
Input : “”””
Classify the text into Positive, Negative, Neutral, and return the answer as the corresponding sentiment.
text: {tweet}
“”””
Output : “label: <generated label>”

2. **Offensive speech prediction**:

   Fine-tuning training Prompt :
   “”””
   Input:
   Classify the text into 0 (Not Toxic), 1 (Toxic), and return the answer as the corresponding toxicity label.
   text: {tweet}
   label: {label_en}
   “”””

   Fine-tuning Testing Prompt :
   “”””
   Classify the text into 0 (Not Toxic), 1 (Toxic), and return the answer as the corresponding toxicity label.
   text: {tweet}
   “”””
   Output : “label: <generated encoded label>”

Explanation:

In fine-tuning, the model learns to recognize patterns by processing structured input-completion pairs tailored to each task, such as sentiment analysis or offensive speech detection. Each prompt specifies the task and label format, ensuring clarity and minimizing the need for complex instructions. The model is trained on diverse examples in the training dataset and later applied to label the testing dataset.

Due to OpenAI's deprecation of the Curie model, we shifted to GPT-Davinci-002, following platform recommendations.