CSE508 Information Retrieval
Winter 2024
Assignment - 4

Report

Siddharth
2021424

Preprocessing:

I have used the Amazon review dataset. Cleaned and preprocessed the ‘Text’ and ‘Summary’ column of the Reviews file.

Assumption: I have removed the rows having ‘NaN’ values from the dataset as the rows were significantly smaller than the whole dataset, so it won’t have any effect on the model training.

Model Training:

After preprocessing the dataset and saving it to a new csv. I moved to Kaggle for better computation in model training and testing.

Then I followed the below-given steps:

Initialize GPT-2 Tokenizer and Model:

Utilize the Hugging Face transformers library to initialize a GPT-2 tokenizer and model.
Load a pre-trained GPT-2 model and tokenizer, such as GPT2LMHeadModel and GPT2Tokenizer.
Split Dataset into Training and Testing Sets:
Divide the dataset into training and testing sets, usually with a split ratio like 75:25 or 80:20.
The training set will be used to train the model, while the testing set will be used to evaluate its performance.

Implement Custom Dataset Class:
Create a custom dataset class to prepare the data for training.
This class should handle tasks like tokenizing the text, padding sequences, and converting them into tensors.
You can use PyTorch's Dataset class and DataLoader to efficiently load and preprocess the data.

Fine-Tune the GPT-2 Model:
Fine-tuning involves training the pre-trained GPT-2 model on your specific dataset.
During training, the model's parameters are updated to minimize a loss function, typically cross-entropy loss.
Train the model on the training set using techniques like backpropagation and gradient descent.
Monitor the model's performance on the testing set to prevent overfitting.

Experiment with Hyperparameters:
Hyperparameters such as learning rate, batch size, and number of epochs significantly impact the model's performance.
Experiment with different hyperparameter values to optimize the model's performance.
Techniques like grid search or random search can be used to find the best combination of hyperparameters.
Monitor metrics such as loss and evaluation scores (e.g., ROUGE scores for summarization tasks) to assess the model's performance under different hyperparameter settings.

By following the above steps I was able to fine-tune my GPT-2 model for generating summaries.

Evaluation/Testing:

After training the model, I generated summaries using the .pth file of the output of trained model. Using the generated summaries, then calculated the Rouge scores to evaluate the quality of the generated summaries.

I have followed the following steps:

Model Initialization:
I initialize a GPT-2 model and tokenizer from the Hugging Face transformers library. The model is loaded with pre-trained weights from the "gpt2" model variant.
Additionally, loaded the fine-tuned weights for the model from a saved state dictionary file.

Summary Generation:
The generate_summary function takes a review text as input and generates a summary using the fine-tuned GPT-2 model. It tokenizes the input text, generates the summary using beam search, and decodes the generated summary into human-readable text.

ROUGE Score Calculation:
The compute_rouge_scores function computes ROUGE scores (ROUGE-1, ROUGE-2, and ROUGE-L) between the actual summary provided by the user and the generated summary.
We use the rouge_scorer module from the rouge_score package to compute ROUGE scores.

User Interaction:
The code prompts the user to input the review text and the actual summary text for evaluation.

Evaluation and Reporting:
After generating the summary, the code computes ROUGE scores between the actual and generated summaries.
Finally, it prints out the generated summary along with the ROUGE scores (precision, recall, and F1-score) for each ROUGE metric.
In summary, the code demonstrates a practical implementation of summarization using a fine-tuned GPT-2 model and evaluates the quality of the generated summaries using ROUGE scores. This system can be used to automate the process of summarizing text and assess the performance of the summarization model.