202131111

Report

Introduction:

The provided code aims to fine-tune the GPT-2 model on a dataset of Amazon fine food reviews and then evaluate the model's performance by generating summaries and calculating ROUGE scores.

Data Preprocessing:

Data Preprocessing:

Loading the Data: The code starts by importing the dataset containing Amazon fine food reviews. It ensures that there are no missing values by removing rows with empty fields.

Selecting Relevant Columns: The focus is on two columns: 'Summary' and 'Text'. 'Summary' contains short summaries of the reviews, while 'Text' contains the full reviews.

Cleaning the Text: Before processing, the text in both columns is converted to lowercase for consistency. Special characters and HTML tags are removed using regular expressions and BeautifulSoup. This cleaning step is crucial to ensure the text is in a uniform format.

Removing Stopwords: The Natural Language Toolkit (NLTK) is used to remove common English stopwords like 'and', 'the', and 'is', which might not contribute much to meaningful analysis.

Splitting the Data: After preprocessing, the dataset is split into two parts: a training set and a testing set. The split is 75% training and 25% testing, a common ratio in machine learning, where the training set is used to fine-tune the model, and the testing set is used to evaluate it.

Model Fine-tuning:

GPT-2 Initialization: The code initializes the GPT-2 tokenizer and model, essential components for processing text and generating new content.

Custom Dataset: A custom dataset class is defined to format the dataset appropriately for training with GPT-2.

Training Arguments: Various parameters for training the model are set, such as the output directory for saving model checkpoints, batch size, number of training epochs, and learning rate. These settings control the behavior of the training process.

Training the Model: A Trainer object is initialized with the GPT-2 model, training arguments, and a data collator to ensure data is processed in the correct format. The model is then fine-tuned using the training dataset.

CustomDataset class is defined to prepare the dataset for training.

TrainingArguments are set up, including output directory, batch size, number of epochs, and learning rate.

Trainer is initialized with the model, training arguments, and data collator.

The model is fine-tuned on the training dataset.


```
{'input_ids': tensor([50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
        50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
        50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
        50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
        50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
        50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
        50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
        50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
        50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
        50256, 50256, 50256, 50256, 16370,  1266,   875,  1878,   220,   425,
         1683,  3088, 18548,  9059,  1588,  6508,  4634,  9059, 15649,  4634,
        17666,  1949,  7539, 15649,  4939,  2158, 40163, 15649,  7209, 25103,
        18221,   588,  6891, 12922, 18548,  1560,   875,  1878]), 'attention_mask': tensor([0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1]), 'labels': tensor([50256, 50256, 50256, 50256, 50256, 50256, 50256, 502
0256, 50256,
        50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
        50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256, 50256,
  warnings.warn(
```


[9360/9360 48:24, Epoch 20/20]

Step	Training Loss
500	6.265600
1000	6.235900
1500	6.216700
2000	6.229800
2500	6.242900
3000	6.250500
3500	6.230800
4000	6.239700
4500	6.237600
5000	6.231700
5500	6.228100

Evaluation Metrics:

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores are used to evaluate the quality of generated summaries.

The compute_rouge_scores() function calculates ROUGE-1 , ROUGE-2 and ROUGE-L scores between a reference summary and a generated summary.

The calculate_rcs() function computes ROUGE scores for all summaries in the testing set.

Average ROUGE scores are calculated from the obtained scores.

```
 rouge-1:
Precision: 0.05, Recall: 0.44, F1-Score: 0.08
 rouge-2:
Precision: 0.01, Recall: 0.13, F1-Score: 0.02
 rouge-l:
Precision: 0.04, Recall: 0.41, F1-Score: 0.08
```

```
Average rouge-1 score: 0.0826138011637614
Average rouge-2 score: 0.017829062615820387
Average rouge-l score: 0.0774726200968081
```

Summary Generation:

The generate_summary() function generates a summary for a given input text using the fine-tuned GPT-2 model.

Example usage demonstrates generating a summary for a specific input text.

ROUGE Score Calculation:

ROUGE scores are calculated for each generated summary compared to its corresponding reference summary.

ROUGE scores are stored in a pickle file for later analysis.

Average ROUGE scores are calculated across all summaries in the testing set.

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Generated Summary: bought several Vitality canned dog food products found good quality product looks like ste
w processed meat smells better Labrador finicky appreciates product better dog food dog food dog food dog foo
d dog food
```

Conclusion:

The provided code successfully fine-tunes the GPT-2 model on Amazon fine food reviews data, generates summaries, and evaluates their quality using ROUGE scores. The average ROUGE scores provide insights into the model's performance in generating coherent and informative summaries.

Thankyou!