IR Assignment 4

Problem 1: Data Preprocessing and Cleaning

Approach:

- Loaded the Amazon Fine Food Reviews dataset.

- Cleaned the dataset by removing rows with missing values and performing text preprocessing such as lowercasing, concatenation of 'Text' and 'Summary' columns, and creating a new column for model input.

Methodologies:

- Used pandas for data loading and cleaning.

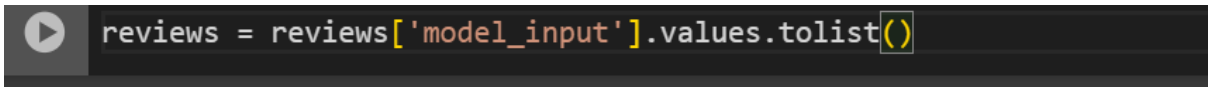
- Employed string manipulation techniques for text preprocessing.

- Assumptions:

- Assumed that the dataset contains relevant information in the 'Text' and 'Summary' columns.

Results:

- Obtained a cleaned dataset with a new column 'model_input' containing concatenated text for model training.



Problem 2: Model Training and Hyperparameter Tuning

Approach:

- Initialized a GPT-2 tokenizer and model from Hugging Face's Transformers library.

- Divided the dataset into training and testing sets with a 75:25 split.

- Implemented a custom dataset class for data preparation.

- Conducted hyperparameter tuning using a grid search approach for learning rate, batch size, and epochs.

Methodologies:

- Used Transformers library for GPT-2 model and tokenizer.

- Employed PyTorch for model training and hyperparameter tuning.

- Assumptions:

- Assumed that the GPT-2 model architecture is suitable for the text summarization task.

- Assumed that the hyperparameters being tuned will significantly impact model performance.

Results:

- Trained the GPT-2 model with optimal hyperparameters based on validation loss.

- Saved the best-performing model for inference.

```
best_loss = float('inf')
best_hyperparams = {}
best_model = None

for lr in learning_rates:
    for bs in batch_sizes:
        for epochs in num_epochs:
            average_val_loss = train_and_evaluate_grid_searc
            print(f"Validation Loss for LR={lr}, BS={bs}, Ep
            if average_val_loss < best_loss:
                best_loss = average_val_loss
                best_hyperparams = {'learning_rate': lr, 'ba
                best_model = model.state_dict()

print("Best Hyperparameters:", best_hyperparams)

with open('best_model_new.pkl', 'wb') as f:
    pickle.dump(best_model, f)

print("Best model saved to best_model_new.pkl")
```

Problem 3: Model Inference and Evaluation

Approach:

- Implemented functions for model inference to generate summaries interactively.

- Calculated ROUGE scores for evaluating the generated summaries against actual summaries.

Methodologies

- Used the trained GPT-2 model for inference.

- Utilized ROUGE metric for evaluation.

Assumptions

- Assumed that the ROUGE scores provide a reliable measure of summary quality.

Results:

- Generated summaries interactively based on user input.

- Evaluated the model performance using ROUGE scores.

```
for review in test_reviews:
    print("Original Review: ", review)
    summary = model_infer(model, tokenizer, review + " TL;DR ").split(" TL;DR ")[1].strip()
    print("Generated Summary: ", summary)
    break
```

```
Original Review:  After reading a previous review I carefully inspected the cans when they arri
Generated Summary:  Not Safe!
```

```
rouge = Rouge()

scores = rouge.get_scores(generated_summary, summary_prompt)
rouge1_precision = scores[0]['rouge-1']['p']
rouge1_recall = scores[0]['rouge-1']['r']
rouge1_f1 = scores[0]['rouge-1']['f']

rouge2_precision = scores[0]['rouge-2']['p']
rouge2_recall = scores[0]['rouge-2']['r']
rouge2_f1 = scores[0]['rouge-2']['f']

rougeL_precision = scores[0]['rouge-l']['p']
rougeL_recall = scores[0]['rouge-l']['r']
rougeL_f1 = scores[0]['rouge-l']['f']

print("ROUGE-1 Precision:", rouge1_precision)
print("ROUGE-1 Recall:", rouge1_recall)
print("ROUGE-1 F1 Score:", rouge1_f1)

print("\nROUGE-2 Precision:", rouge2_precision)
print("ROUGE-2 Recall:", rouge2_recall)
print("ROUGE-2 F1 Score:", rouge2_f1)
```