

Sentiment classification with context

By Tareq Ahrari
2024848253

Task

- For this task i do sentiment classification. I will use an pre-trained model for sentiment classification and fine-tune it for contextual sentiment classification. That is because usually models just focus on isolated pieces of text, but I will try to make it work on context as well.

Model and Datasets

- IMDB Dataset for movie reviews
- Yelp polarity dataset for restaurant reviews
- Bert Model (bert-base-uncased)

Codebase

- Bert-base-uncased
- Pre-trained BERT model from Huggingface trained on lowercase english text
- 12 layers (each with 12 attention heads), 768 hidden dimensions
- ~ 110M parameters
- Added classification head -> (positive, neutral, negative)
- Pre-trained on BookCorpus and Wikipedia

Datasets

- IMDB is a large movie review dataset
- Used for binary sentiment classification
- Yelp polarity dataset is a large yelp review dataset
- I added contextualization information by combining the given texts with adjacent passages.
- I also tokenized the datasets for training purposes

Fine-tuning

```
training_args = TrainingArguments(  
    output_dir="./results",  
    evaluation_strategy="epoch",  
    save_strategy="epoch",  
    learning_rate=2e-5,  
    per_device_train_batch_size=8,  
    num_train_epochs=3,  
    weight_decay=0.01,  
    logging_dir="./logs",  
    logging_steps=100,  
    save_total_limit=2,
```

Performance on IMDB

```
metrics_pretrained = evaluate_model_on_test(model_pretrained, test_dataset)
metrics_finetuned = evaluate_model_on_test(model_finetuned, test_dataset)
```

```
print(f"Pre-trained Model Metrics: {metrics_pretrained}")
print(f"Fine-Tuned Model Metrics: {metrics_finetuned}")
```

```
Pre-trained Model Metrics: {'accuracy': 0.5539821592863714, 'f1': 0.5529134069708699}
Fine-Tuned Model Metrics: {'accuracy': 0.890555622224889, 'f1': 0.890555622224889}
```

Performance on YELP

```
[45]: results = {
    "Dataset": ["Yelp"],
    "Pre-Trained Accuracy": [pretrained_metrics_yelp["accuracy"]],
    "Fine-Tuned Accuracy": [finetuned_metrics_yelp["accuracy"]],
    "Pre-Trained F1": [pretrained_metrics_yelp["f1"]],
    "Fine-Tuned F1": [finetuned_metrics_yelp["f1"]],
}

import pandas as pd
df_results = pd.DataFrame(results)
print(df_results)
```

	Dataset	Pre-Trained Accuracy	Fine-Tuned Accuracy	Pre-Trained
0	Yelp	0.514119	0.813837	0.513
	Fine-Tuned F1			
0		0.812995		

Improved the Model

- I improved the model for ~30% better F1 and Accuracy scores against the baseline model

Text: The service was excellent.

Context: The food was terrible.

Pre-trained Model Prediction: Neutral

Fine-Tuned Model Prediction: Neutral

Text: I enjoyed the experience.

Context: It was my first visit.

Pre-trained Model Prediction: Neutral

Fine-Tuned Model Prediction: Neutral

Text: The product quality is amazing.

Context: But the delivery was late.

Pre-trained Model Prediction: Neutral

Fine-Tuned Model Prediction: Negative

Example