9/5/2024

Homework 7
Emotion Detection Spring2024
Report

Patil, Praful Venkatesh
PVP22001

Task:

The task is to develop a model that accurately identifies the presence of multiple emotions within a tweet. Given the intricate nature of emotional expression, this task moves beyond traditional binary or single-label classification, requiring a more sophisticated, nuanced approach. A tweet often contains a blend of emotions, making it essential to capture their subtle interplay. By leveraging multi-label classification techniques, the model can identify various emotions concurrently, providing a comprehensive understanding of the underlying sentiment. The primary emotions considered include anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, and trust, each of which can co-occur or stand alone in varying degrees. The goal is to ensure that the model detects each emotion's presence and intensity with high accuracy, ultimately enabling a more realistic and insightful sentiment analysis of social media data.

Data:

The dataset used in this model is from the "Emotion Detection Spring2014" competition, comprising the following files:

- **train.csv**: Contains training data with annotated tweets. Below are the columns
  (1) **ID:** Unique identifier for each tweet.
  (2) **Tweet:** Text of the tweet.
  (3) **Emotion Labels**: Binary indicators (0 or 1) for the presence or absence of emotions like anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, and trust.
- **test.csv:** Contains the test data, providing only the tweet text and IDs without emotion labels. Below are the columns
  (1) **ID:** Unique identifier for each tweet.
  (2) **Tweet:** Text of the tweet.
- **sample_submission.csv:** Offers a template for submitting predicted emotion labels in the required format.

Results and Performances of Models used:

1) google/gemma-1.1-2b-it model fine-tuned using LoRA

Evaluation Results:

eval/steps_per_second	9.942
eval_accuracy	0.19094
eval_f1_macro	0.60799
eval_f1_micro	0.68651
eval_loss	0.50574

Kaggle F1 Macro Score:

F1_Macro	**0.57448**

Link to WandB for this model: https://api.wandb.ai/links/utd659/qdc34slx

2) google/gemma-1.1-2b-it model fine-tuned using IA3

Evaluation Results:

eval/steps_per_second	6.455
eval_accuracy	0.16764
eval_f1_macro	0.55669
eval_f1_micro	0.6624
eval_loss	0.53524

Kaggle F1 Macro Score:

F1_Macro	**0.48294**

Link to WandB for this model: https://api.wandb.ai/links/utd659/77jn6ih0

3) e5-mistral-7b-instruct model fine-tuned using QloRA

Evaluation Results:

eval/steps_per_second	1.665
eval_accuracy	0.1257
eval_f1_macro	0.39739
eval_f1_micro	0.57765
eval_loss	0.6512

Kaggle F1 Macro Score:

F1_Macro	**0.4034**

Link to WandB for this model: https://api.wandb.ai/links/utd659/7gj1z5ux

4) Qwen1_5_7B model fine-tuned using QloRA

Evaluation Results:

eval/steps_per_second	0.476
eval_accuracy	0.19871
eval_f1_macro	0.6055
eval_f1_micro	0.70209
eval_loss	0.49164

Kaggle F1 Macro Score:

F1_Macro	**0.5380**

Link to WandB for this model: https://api.wandb.ai/links/utd659/f6z1qp39

5) Gecko-7B-v0.1 fine-tuned using QloRA

Evaluation Results:

eval/steps_per_second	1.689
eval_accuracy	0.21294
eval_f1_macro	0.59385
eval_f1_micro	0.688

eval_loss	0.49095

Kaggle F1 Macro Score:

F1_Macro	**0.6040**

Link to WandB for this model: https://api.wandb.ai/links/utd659/rgpf6847

6) gte-Qwen1.5-7B-instruct fine-tuned using QloRA
Evaluation Results:

eval/steps_per_second	1.961
eval_accuracy	0.24466
eval_f1_macro	0.60325
eval_f1_micro	0.6989
eval_loss	0.55151

Kaggle F1 Macro Score:

F1_Macro	**0.5787**

Link to WandB for this model: https://api.wandb.ai/links/utd659/dcsortue

Analysis:

Among these models, the Gecko-7B-v0.1 model with QLoRA fine-tuning and the gte-Qwen1.5-7B-instruct model stand out for their high F1 Macro scores, demonstrating their ability to generalize well across various classes. While gemma (LoRA) performed well, it didn't surpass the Gecko model in accuracy or F1 Macro, suggesting fine-tuning strategies greatly influence outcomes. The e5-mistral-7b-instruct model requires further analysis due to its relatively low scores, which might point to issues like overfitting.

Hyperparameter Selection Strategies for Fine-Tuning Models:

- **LoRA (Low-Rank Adaptation):**

1) Introduces trainable low-rank matrices to improve adaptability with fewer parameters.
2) Achieved a Kaggle F1 Macro score of 0.57448 for google/gemma-1.1-2b-it.
3) Gecko-7B-v0.1, fine-tuned with QLoRA, obtained an F1 Macro score of 0.6040.

- **IA3 (Intrinsic Attention Adaptation):**

1) Adjusts attention weights using a smaller set of trainable vectors.
2) Scored a Kaggle F1 Macro of 0.48294 for google/gemma-1.1-2b-it.
3) Provides computational efficiency but requires further optimization for better generalization.

- **Instruction Fine-Tuned Models:**

1) gte-Qwen1.5-7B-instruct achieved a Kaggle F1 Macro score of 0.5787.
2) Demonstrated competitive performance in multi-class tasks due to optimized fine-tuning.

Conclusion: Challenges Faced and Learnings Gained

- Even with Colab Pro and access to high-end GPUs like A100 or V100, memory errors were persistent, indicating a demand for resources beyond what the available hardware could provide.
- Longer training times were also a challenge, particularly when fine-tuning larger models.
- LoRA provided higher adaptability and generalization, while IA3 delivered lower performance despite being computationally efficient.
- Instruction fine-tuning produced competitive results, though computational requirements were significant.
- Selecting the appropriate fine-tuning strategy proved critical, as the choice greatly impacted model performance.
- Some models were private and not accessible, limiting the ability to compare or leverage them directly.
- Despite trying several models, it was difficult to achieve an F1 score higher than 61%, highlighting the challenge of reaching peak performance across diverse classes.