SemEval - Fine-tuning LLM for sentiment classification

Andrey Sypachev
5126511
Module

Long Tran
5192922
M1101-CMS72

Abstract

Why do language models need to recognize human emotions? When we read messages, posts or emails, we can easily miss important nuances of the author's mood and attitude towards a situation. If we understand the emotional coloring of a text, we can more accurately interpret the meaning and establish a closer connection with the person who wrote the message. Without recognizing emotions, we lose some of the context that affects the depth of communication. Communicating via chat can be a source of stress or, on the contrary, support. Recognizing the emotional coloring of the text helps to notice in time that someone may need help, support or at least a kind word. For example, algorithms that detect signs of depression in posts can help to intervene in time and suggest a person to consult a specialist [1]. Thanks to recognizing human emotions, we become more attentive and sensitive to the world around us. This makes life richer, brighter and more meaningful. This report focuses on describing the fine-tuning process of existing language models for emotion recognition on an English text dataset.

See this Template on Overleaf.

1 Introduction

Human communication is inherently complex, and we frequently use language in indirect ways to express our emotions. We might say one thing while meaning another, particularly when dealing with sensitive topics, criticism and making jokes. This discrepancy between literal words and intended emotional meaning makes accurately guessing someone's emotional state a significant challenge. Oral communication's emotional interpretation is already challenging, interpretation of written text is even more problematic due to missing tones and facial expressions. Furthermore, individuals vary in how they express and perceive emotions, adding another layer of complexity. One person's innocent comment might be interpreted as offensive by another. Therefore, we can never be entirely certain about the true emotions of someone behind a written message.

While the task of understanding emotions expressed in texts can be framed as a multi-label classification problem, where multiple emotions might be present simultaneously, the real difficulty lies in the subtle difference between the sentiment expressed in the text itself and the actual emotion being felt by the speaker. Words may express one emotion, but the tone of voice or context might express another. To attempt to aid human in sentiment detection in texts, we fine-tune the large language model (LLM) BERT, in particular, BERT-Large to classify text snippets by emotions: joy, sadness, fear, anger, and surprise. In section 2, we will mention the different models involved in our experiments. In section 3, we describe in detail the different models and the training parameters we experimented on. Next, in section 4, we evaluate the results produced by different models and the effects of the parameters described in the previous section, judging by the F-score. After that, we will discuss what other methods can potentially be applied to further improve the score and fine-tune the model more efficiently for the multi-label text classification task.

2 Related Work

An example citation: (**?**).

3 Methodology

In this section, we will describe the fine-tuning process, technology and necessary components used during the fine-tuning process.

We apply Hugging Face framework for the process to achieve efficiency while maintaining simplicity. A fixed dataset was acquired from the from Codabench, an open-source platform allowing you to organize AI benchmarks. The dataset is then tokenised and split into smaller fractions for training and evaluation. A pre-trained LLM model is then selected from Hugging Face's library. Next, we conducted experiments on the training arguments and save the LLM model with the best F-score. After that, the fine-tuned model is used to perform test prediction on examples.

3.1 Dataset

3.2 Pre-trained Model

3.3 Training Arguments

3.4 Compute Metrics

3.5 Explainer

4 Evaluation

5 Discussion

6 Contribution statement

If you work in a team, describe here briefly who did what.

Instructions:

1. Clone/copy this file and fill it with your content.

2. Write no more than 8 pages (references do not count towards this).

3. In "Introduction", briefly lay out the problem you address, and your contribution. Include an overview/teaser image.

4. In "Related Work", reference both (1) a selection of previous works on the same/similar problems (and try to differentiate your approach from those), (2) a set of foundational literature relevant to the problem, and your methodology.

5. "Methodology" lays out your technical approach, high-level, as well in technical detail. Subsections are highly recommended here (and also elsewhere).

6. "Evaluation" should contain both a quantitative and a qualitative evaluation of your results. If in doubt about metrics and evaluation methodologies, talk to us.

7. "Discussion" on the one hand builds upon the evaluation, and should critically discuss strengths and weaknesses of your solution, and possible ways to improve it further. On the other hand, it should discuss relevant ethical questions related to the problem and/or your solution at hand.