Overview
- **Motivation**: Emojis play a crucial role in conveying emotions, making accurate emoji prediction a useful task to explore.
- **Goals**: Predict emojis for messages in supervised setting and generalize to new emojis in zero-shot setting.

Problem Setup
- Denote the set of emoji labels by $E$ and the dataset by $D = \lbrace \epsilon_i, \epsilon_j, \ldots \rbrace_n$ where $\epsilon_i = \lbrace \epsilon_{i1}, \epsilon_{i2}, \ldots \epsilon_{in} \rbrace$ represents a text sequence with $n$ tokens and $\epsilon_i$ refers to a single emoji in the label set $E$.
- Given a tweet $\mathbf{t}$, the task is to predict the $\epsilon \in E$ that best associates with $\mathbf{t}$.
- In the supervised setting, dataset $D$ can be randomly split into $D_{\text{train}}$, $D_{\text{valid}}$, $D_{\text{test}}$.
- In the zero-shot setting, we ensure that the test label set is disjoint from the training label set, i.e., $E_{\text{train}} \cap E_{\text{test}} = \emptyset$, so that the labels predicted at test time are unseen in training.

Data
- **emoji-100k-49**: 100,000 tweets with a single label from 49 emoji classes.
- **emoji-100k-20**: select the 20 most used emojis in emoji-100k-49, 75,087 tweets remaining.
- **emoji-1m-49**: 1,000,000 tweets with a single label from 49 emoji classes.
- **emoji-1m-20**: select the 20 most used emojis in emoji-1m-49, 749,570 tweets remaining.
- **Split data**: 80% Train, 10% Validation, 10% Test

Experiments
- **Supervised setting**
  - Use bert-base-cased and gpt_small as the base models for finetuning.
  - Stack a classification head on top and finetune all layers.
  - Predict the label with the highest probability.
- **Zero-shot setting**
  - Use gpt_small as the base model.
  - Stack a language modeling head on top and instruction tune all layers.
  - Prediction
  - Given a prompt, generate the next token.
  - Compute a score for each emoji label using chain rule. Denote a tweet as $\mathbf{t}$ and an emoji label $\epsilon = \lbrace \epsilon_1, \epsilon_2, \ldots \epsilon_n \rbrace$.
  - $s(\epsilon_\mathbf{t}) = \prod_{i=1}^{n} \mathbb{P}(\epsilon_i | \epsilon_{i-1}, \ldots, \epsilon_1)$

Methods
- **(A) Pretrain-finetune**
- **(B) Instruction Tuning**

Results
- **Unfinetuned**

References