Dataset Report 20

Mario Reda Matricola: 1937116

1 Dataset Description

The second task I worked on is task 20 "ITAmoji". This task aims to predict the most likely emoji that can be associated with the text of a tweet. In this case, unlike the previously analyzed task 22, we do not have a binary file, but we have a very extensive set of emojis. Additionally, I used the "counter.py" code to calculate the number of occurrences of emojis. Here are some of them: red heart: 50680, face with tears of joy: 49655, smiling face with heart eyes: 23627, winking face: 13371

2 Dataset Format

This dataset is structured into two parts:

- The training set contains a series of tweets from different users on social media. Each tweet is represented as a JSON object with the following fields:
  - uid: a unique identifier for the sample
  - text no emoji: the text of the tweet without emojis
  - label: the label indicating the correct emoji associated with the tweet
  - tid: the tweet ID
- The test dataset comprises a collection of tweets from various sources. Each tweet is structured as a JSON object with the same fields as the training dataset.

Each sample in the dataset must be represented as a JSON object, with the following fields:

- sentence: the text of the tweet
- choices: a list of emojis to choose from
- label: the index of the correct emoji in the list of choices

3 Methodology for Dataset Reframing

The training and test data are loaded from the provided files and processed to create the training and test datasets. During this process, class labels are replaced with their corresponding Italian translations to enhance understanding and interpretation of the results.

4 Methodology for distractor generation

The output of the TF-IDF vectorizer is a matrix where each row represents a document and each column represents a term, and the cell values represent the TF-IDF score of each term in each document. The Naive Bayes classifier is a probabilistic classifier based on Bayes' theorem with the "naive" assumption of independence between features given the class. Let X be a feature vector representing the input data, and C_k be a class label from the set of possible classes $\{C_1, C_2, \ldots, C_K\}$. The probability of class C_k given the input X can be calculated using Bayes' theorem:

$$P(C_k|X) = \frac{P(X|C_k) \times P(C_k)}{P(X)}$$

In practice, the denominator $P(X)$ is constant for all classes, so it can be ignored when comparing the probabilities for different classes. Under the "naive"

assumption of feature independence, the conditional probability $P(X|C_k)$ can be factorized as:

$$P(X|C_k) = P(x_1|C_k) \times P(x_2|C_k) \times \ldots \times P(x_n|C_k)$$

where $x_1, x_2, \ldots, x_n$ are the TF-IDF scores of X. Since the dataset is highly imbalanced, the model would predict always the classes with the more number of samples, therefore a straight approach would be to oversample the less represented labels to have the same numbers. Although it decreases test accuracy, it is necessary in a task like this where it is more important to not have always the same possible choices. Also the distractors are sampled according to the probabilities output by the model by making it more flexible.

5 List of Suitable Prompts

- "Prevedi quale emoji sia più probabile associare al testo del tweet: 'sentence'. Scegli tra le opzioni: choices."
- "Quale emoji pensi si adatti meglio al contenuto del tweet: 'sentence'? Seleziona tra: choices."
- "Valuta quale emoji sia più adatta al testo del tweet: 'sentence'. Opzioni disponibili: choices."
- "Scegli l'emoji che ritieni più appropriata per il testo del tweet: 'sentence'. Scegli tra le seguenti opzioni: choices."
- "Considera il testo del tweet: 'sentence'. Quale emoji credi meglio lo rappresenti? Scegli tra: choices."

6 Instructions for Running Code

To run the code, simply run the files named "save train" and "save test", which will subsequently generate all the required JSON files.