

### ****Summary and Description of the Emo-SL Framework: Emoji Sentiment Lexicon Using Text-Based Features and Machine Learning for Sentiment Analysis****

The **Emo-SL Framework** leverages a combination of text-based and emoji-based features for sentiment analysis. The primary goal is to extract meaningful sentiment from social media posts (like tweets) by considering both the words in the text and the emojis present within it. The framework uses machine learning to compute sentiment scores based on these extracted features.

#### **Key Components:**

1. **Text-based Sentiment Analysis**:
   * The framework uses a **Sentiment Lexicon** to assign sentiment scores to words. Each word is checked for its sentiment value (positive or negative), and these scores are aggregated to determine the overall sentiment of the tweet.
2. **Emoji-based Sentiment Analysis**:
   * Emojis play a crucial role in social media sentiment. The framework extracts emojis from tweets and assigns sentiment scores based on an **Emoji Sentiment Lexicon**, which associates sentiment values with individual emojis.
3. **Feature Extraction**:
   * A **Feature Vector** is constructed for each tweet, capturing various features such as:
     + The count of positive and negative words.
     + The count of positive and negative emojis.
     + The total number of emojis in the tweet.
     + The overall sentiment (positive/negative) based on the aggregate sentiment scores.
4. **Emoji Occurrence Counting**:
   * The framework counts how many times each emoji appears in positive and negative tweets to determine the emoji's role in sentiment analysis.
5. **Sentiment Score Calculation**:
   * Emoji sentiment scores are calculated using the ratio of positive to negative occurrences of each emoji, which provides a numerical representation of the emoji's sentiment.

### ****Detailed Code Explanation and Flow:****

1. **Feature Extraction for Sentiment Analysis (Algorithm 2)**:
   * Extracts both **text-based features** (word sentiment scores from the sentiment lexicon) and **emoji-based features** (emoji sentiment scores from the emoji sentiment lexicon).
   * The **Feature Vector** stores:
     + Counts of positive/negative words and emojis.
     + Total number of emojis.
     + Sentiment scores for words and emojis.
   * The sentiment is then determined based on the aggregated positive/negative scores.
2. **Counting Emoji Occurrences (Algorithm 3)**:
   * Counts the number of times each emoji appears in positive and negative tweets.
   * The output is two dictionaries: one for positive emoji counts and another for negative emoji counts.
3. **Calculating Emoji Sentiment Scores (Algorithm 4)**:
   * Calculates the sentiment scores for each emoji based on its positive and negative occurrences using the formula: Score=p−np+n\text{Score} = \frac{p - n}{p + n} where:
     + pp is the number of occurrences in positive tweets.
     + nn is the number of occurrences in negative tweets.

### ****Input and Output Examples:****

#### **Input**:

The input consists of a dataset of **tweets**. Each tweet is a string that may contain words and emojis.

Example tweet:

tweets = [

"I love this product! 😊😍", # Positive sentiment

"This is the worst service ever 😡😞", # Negative sentiment

"Feeling neutral about this 😐" # Neutral sentiment

]

* **Sentiment Lexicon** (Example):
  + "love" → +0.5
  + "worst" → -0.7
* **Emoji Sentiment Lexicon** (Example):
  + "😊" → +0.8
  + "😍" → +0.9
  + "😡" → -0.9
  + "😞" → -0.7
  + "😐" → 0.0

#### **Process**:

1. **Feature Extraction**:
   * For each tweet, the framework extracts sentiment scores for words and emojis.
   * Example: For the tweet "I love this product! 😊😍", the extracted feature vector will include counts for positive words (e.g., "love") and positive emojis (e.g., "😊", "😍").
2. **Counting Emoji Occurrences**:
   * Count how many times each emoji appears in positive and negative tweets.
   * Example: "😊" appears 1 time in a positive tweet and 0 in a negative tweet.
3. **Calculating Sentiment Scores**:
   * Based on the counts of positive and negative occurrences, the sentiment scores for emojis are computed.
   * Example: For the emoji "😊", if it appears 3 times in positive tweets and 1 time in negative tweets, the score would be calculated as: Score=3−13+1=0.5\text{Score} = \frac{3 - 1}{3 + 1} = 0.5

#### **Output**:

* **Feature Vector Output**:
  + For each tweet, a feature vector is produced that includes sentiment scores for the words and emojis.

Example for the tweet "I love this product! 😊😍":

{

"positive\_words\_count": 1,

"negative\_words\_count": 0,

"positive\_emojis\_count": 2,

"negative\_emojis\_count": 0,

"total\_emojis\_count": 2,

"positive\_score": 1.4,

"negative\_score": 0.0,

"emojis": ["😊", "😍"],

"sentiment": "positive"

}

* **Aggregated Emoji Counts**:
  + The total positive and negative emoji counts across all tweets.

Example:

{

"positive\_emojis\_count": {"😊": 3, "😍": 2},

"negative\_emojis\_count": {"😡": 2, "😞": 1}

}

* **Emoji Sentiment Scores**:
  + For each emoji, the sentiment score is calculated and returned.

Example for emoji sentiment scores:

{

"😊": 0.5,

"😍": 1.0,

"😡": -0.9,

"😞": -0.7,

"😐": 0.0

}

### ****Summary****:

The Emo-SL framework combines **text-based sentiment analysis** and **emoji sentiment analysis** to provide a comprehensive method for understanding social media posts. The use of sentiment lexicons for both words and emojis allows for a detailed sentiment analysis that accounts for the emotional tone conveyed through both text and visual symbols (emojis). The system can be applied to analyze large datasets of social media content, providing insights into public sentiment based on both textual and visual cues.