NLP Project 8

Severi Raunama sraunama20@student.oulu.fi

Specifications achieved in this project: 1,2,4,5,6,7,8,10 and 11

# Introduction

Project 8 explores the connection between emoticon and emotions using lexicon and embedding based analysis. In this project I have explored how emoticons relate to emotions, specifically anger, anticipation, disgust, fear, joy, sadness, surprise and trust.

In this project I have used multiple different datasets and language processing libraries to quantify these connections. The [EmoTag](https://github.com/abushoeb/EmoTag/blob/master/data/EmoTag1200-scores.csv) dataset and the [Tweets with Emoji dataset](https://www.kaggle.com/datasets/ericwang1011/tweets-with-emoji) play a significant role in this analysis.

## EmoTag

The EmoTag dataset consists of a 150 emojis with each having a manually annotated score in the eight emotions (anger, anticipation, disgust, fear, joy, sadness, surprise and trust). These scores are used to evaluate how well different algorithms correlate with the scores given in the data.

Example of data:

Unicode emoji name anger anticipation disgust fear joy sadness surprise trust

1F308 🌈 rainbow 0.0 0.28 0.0 0.0 0.69 0.06 0.22 0.33

## Tweets with Emoji

The “tweets with emoji” dataset consist of a list of emojis, each having a dataset of 20 thousand tweets that include the emoji. From these emojis I have chosen six different emojis for my analysis: Egg 🥚, Enraged face 😡, fire 🔥, skull 💀, smiling face 😀, sun ☀️.

These tweets are used to model in which kinds of contexts these emojis often appear and are used to find interesting patterns.

# Specifications

## Specification 1

In the first specification I was tasked with finding the EmoTag dataset.

## Specification 2

In the second specification I used a Word2Vec model to make an embedding of the emojis in EmoTag dataset based on the name of the emoji. Then I created an embedding of each studied emotion with the same method.

With these embeddings I calculated the cosine similarity between the emoji name embedding and the emotion embedding. This returns a number between one and zero signifying the similarity between the emoji and the emotion.

Next, I measured the Pearson correlation between the scores given in the EmoTag dataset and the scores calculated using Word2Vec embedding. The average correlation I calculated was 0.348, which is not that large but still somewhat significant. This signifies that the Word2Vec model and the manually annotated date at least somewhat agree on the apparent emotion of the emoji.

## Specification 4

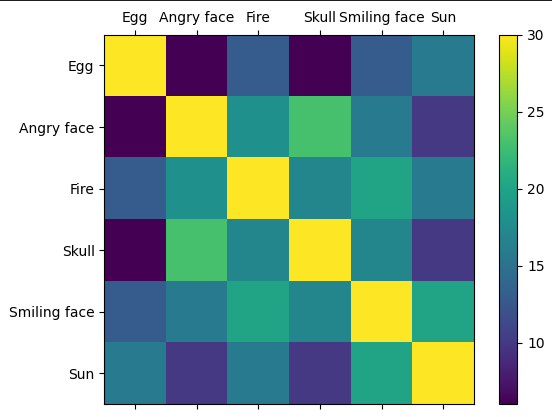
In the 4th specification I used the “Tweets with Emoji” dataset to generate a matrix in which the top 30 common words appearing in each tweet category are visualized for the six chosen emoji (🥚😡🔥💀😀☀️).

First, I extracted the top 30 most common words appearing in each tweet category. Here is an extract from those top 30 words.

A black background with white text

Description automatically generated

Then I used these top 30 words to find how many words each tweet category had in common and made a matrix visualizing that.



## Specification 5

In specification 5 I used a pre-trained “DistilBERT” model to create and embedding based on the tweets of each emoji. I then used dimensionality reduction to make a 2D representation of the embedding and drew a graph based on it.

A white sheet with red and blue dots

Description automatically generated

In the graph, each emoji is represented by a colored dot.

(Green = 🥚 red =😡 orange =🔥white =💀 blue =😀 yellow =☀️)

The distance between each dot should represent the similarity between each tweet group but it is difficult to say how well the model represents that. For example, in the graph the red and the yellow dots are quite close to each other, even though you would expect tweets containing an angry face and tweets containing a sun to be very different.

## Specification 6

The 6th specification was almost the same as the 5th but instead of using DistilBERT to create the embedding I used [empath categorization](https://github.com/Ejhfast/empath-client).

This created the following graph:

A graph with colored dots

Description automatically generated

(Green = 🥚 red =😡 orange =🔥white =💀 blue =😀 yellow =☀️)

This graph shows more expected results than the previous one. For example, tweets containing a smiling face emoji and tweets containing the sun emoji are quite close to each other, signifying their similarity.

# Specification 7

In the 7th specification I used LDA topic modeling to extract the 8 most significant keywords corresponding to each emoji. The language used in the specification was a bit unclear to me, so I am not certain this is what the specification asked me to do.

The most significant keywords for each emoji are as follows:

🥚 ['hope', 'happyeaster', 'day', 'eggs', 'egg', 'happy', 'https', 'easter']

😡 ['know', 'time', 'amp', 'don', 'people', 'like', 'just', 'https']

🔥 ['easter', 'don', 'time', 'amp', 'love', 'like', 'just', 'https']

💀 ['got', 'people', 'know', 'bro', 'don', 'just', 'https', 'like']

😀 ['happy', 'love', 'day', 'just', 'like', 'good', 'thank', 'https']

☀️ ['happy', 'tuesday', 'great', 'gm', 'morning', 'good', 'day', 'https']

## Specification 8

In specification 8 I used another topic modeling method to extract information from the tweets “[BerTopic](https://github.com/MaartenGr/BERTopic)”.

Using the BerTopic library I created a topic table for each category that includes the names of the topics and most significant keywords.

🥚 Tweets

A screenshot of a computer

Description automatically generated

😡 Tweets

A screenshot of a computer program

Description automatically generated

🔥 Tweets

A screenshot of a computer

Description automatically generated

💀 Tweets

A screenshot of a computer

Description automatically generated

😀 Tweets

A screenshot of a computer

Description automatically generated

☀️ Tweets

A screenshot of a computer

Description automatically generated

## Specification 9

In the 9th specification I was supposed to use a project called [torchMoji](https://github.com/huggingface/torchMoji), which can predict a suitable emoji based on a given text, to predict an emoji based on each of the chosen tweet categories in the “tweets with emojis” dataset and to see if the predicted emoji matched the category the tweet belonged in.

Unfortunately, I could not get the torchMoji model weights to load on my machine, so I had to skip this specification. I did take a look at the code behind torchMoji and noticed that some of the emojis I chose are not even possible to be predicted by the torchMoji model. For example, torchMoji could never predict the egg emoji as a suitable emoji for any kind of text.

## Specification 10

In this specification I used WordNet to find antonyms to the top 30 words extracted from specification 4 for two opposite kinds of emojis, in this case the angry face emoji 😡 and the smiling face emoji😀.

For this specification I used the wordnet “get\_antonyms” function to find antonyms for the common words.

The antonyms for the top 30 words appearing in tweets containing the angry face emoji:

['dislike', 'unlike', 'unalike', 'leave', 'take\_away', 'end', 'odd', 'uneven', 'ignore', 'stay\_in\_place', 'come', 'malfunction', 'be\_born', 'stop', 'no-go', 'ever', 'obviate', 'agitate', 'louden', 'moving', 'sparkling', 'no\_longer', 'continuant\_consonant', 'start', 'continue', 'begin', 'unmake', 'break']

The antonyms for the top 30 words appearing in tweets containing the angry face emoji:

['evil', 'evilness', 'bad', 'badness', 'ill', 'dislike', 'unlike', 'unalike', 'hate', 'night', 'unhappy', 'leave', 'take\_away', 'end', 'despair', 'little', 'ignore', 'never', 'sunset']

I used these antonyms to explore how opposite the tweet categories actually are, but surprisingly the antonyms of the top words for each tweet category do not appear in the top words of the other category very often. The only instance of this is the word “never” which is a common word in the angry face tweets and an antonym of a common word in the smiling face tweets.

Other observation was that both these categories have a few antonyms in common:

['dislike', 'unlike', 'unalike', 'leave', 'take\_away', 'end', 'ignore']

## Specification 11

The 11th and finial specification were to find and use appropriate literature to comment on and justify my findings, so here are a few of those

[Emoji, Text, and Sentiment Polarity Detection Using Natural Language Processing](https://www.mdpi.com/2078-2489/14/4/222)

[Semantics and Sentiment: Cross-lingual Variations in Emoji Use - ACL Anthology](https://aclanthology.org/2024.emnlp-main.1041/)

[The study of emoji linguistic behaviour: an examination of the theses raised (and not raised) in the academic literature | Communication & Society](https://revistas.unav.edu/index.php/communication-and-society/article/view/43412)