Emoticons Generation Analysis

Emanuele Poceli, Luigi Manieri, Prosper Playoust, Chau Nguyen

Master's Program in Computer Science and Engineering

University of Oulu, Finland

Github: https://github.com/manuu1311/NLP/blob/main/project/project.ipynb

***Keywords*— Emoticon, Emotion, Embedding, Similarity, Antonymy**

I. Introduction

Today, emoticons have become a huge part of digital communication. They are useful for visual representations of sentiments and emotions and a great tool for others to understand your feelings. This project aims to explore the connection between emoticons and emotion using existing embedding-based analysis. Specifically, we focus on eight fundamental emotions: anger, anticipation, disgust, fear, joy, sadness, surprise and trust.

Some research in NLP has shown that emoticons provide additional emotional cues that are missing in a full textual text and can help to understand better the text (Derks et al., 2008) [1].

Some research also showed that messages containing emoticons are more engaging and memorable (Walter and D'Addario, 2001) [2].

Our research focuses on the link between text, emoticons and emoticons' meaning. We want to investigate the relation between emoticons and the semantic part of text.

In this perspective, we first decided to analyze the proportion of each of the 8 emotions described before within the meaning of the emoticons. That means, using the textual description of each emoticon and estimating their similarity with different embedding methods with the emotions meaning.

To investigate the link between texts and emoticons, we decided to analyze a corpus of tweets from Kaggle. This dataset counts thousands of tweets containing emojis.

This analysis goes through selection of antonym emoticons and the selection of the most used words linked to each of them. We also generate the embedding of each tweet and visualize them in a 2D vectorial space to observe clustering patterns and assess similarity in emoji usage.

After that, we investigate how emojis are linked to specific emotional or thematic content by analyzing the association between emojis and the topics they represent. By examining how well the topics align with the emoji labels, we aim to gain insights into the emotional themes conveyed through emoji usage in tweets. This is done using topic modeling methods.

We also wanted to try the reverse process, transforming a text to an emoji. This work has been implemented in the project with SentenceToEmoji. The results will be compared to the original classification of the tweets.

Finally, we investigate the antagonism relationship between the most frequent words used for antagonists emojis. Antagonist emojis should express the opposite feelings, such as happiness and sadness, and we would like to gain insight into the contrasting emotional theme between them.

II. Methodology

A. Relationship between emojis description and emotion

First of all, to see the relationship between emoji and emotion, we use the textual description of an emoji given in the emoTag1200[4][5] dataset. We selected, 5 different emojis (🌈, 🌙, 🌚, 🌞, 🌟) and collected their textual description (*rainbow*, *crescent moon, new moon face, sun with face, glowing star*). This textual description is saved, and we generate an embedded vector from it using different embedding methods (DistillBERT, word2vec, GLOVE). We use the same embedding method for the 8 emotions and calculate the cosine similarity between all the emotions and each emoticon's description.

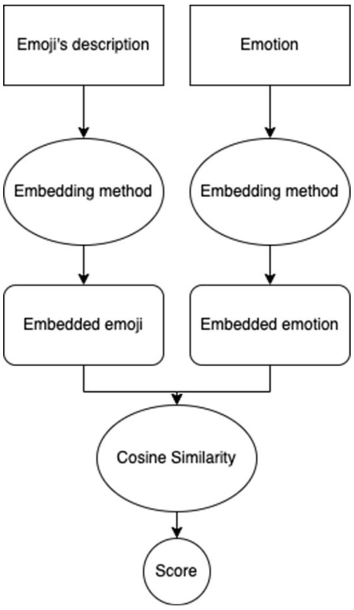

Figure 1 - Emotion score for emoji with embedding system

Finally, we compute the Pearson's correlation to compare our results against the ones obtained in the emoTag project.

B. Semantic link between antonym emoticons and text

To study the semantic link between emoticons and text, we used Tweets with Emoji dataset from Kaggle[6]. We selected 3 pairs of antonym emojis and 20K tweets related to each of them. From these tweets, we selected the 30 most frequent terms linked to each of these emojis. We then have a glossary specific to our emoticons. These different glossaries are compared together to see their common components. This way we see if similar emoticons share similar context.

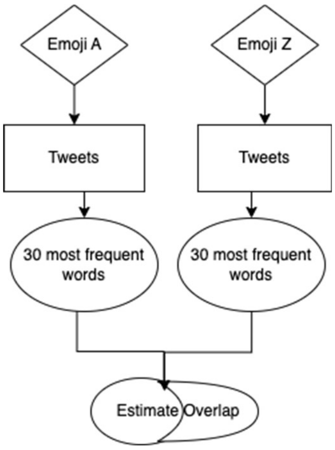

Figure 2 - Relation between lexicon linked to antonym emojis

C. Semantic meaning of emojis used in tweets

We use DistilBERT to embed each of these 20K tweets and TSNE to project these vectors into a 2D space. This way, we expect to see some clusters showing us the similarity between the glossaries of these emoticons.

We repeat the same process using Empath[8] as embedding model, TSNE to project them of the 2D plan and take the mean value to draw the emoticons on the graph.

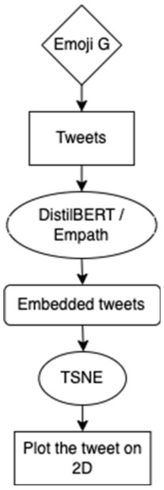

Figure 3 -Semantic meaning of emojis on 2D plot

D. Relationship between topics and emojis

To investigate the relationship between emojis and topics they could represent, we used LDA on the different groups of tweets selected before. LDA is generating a topic through the analysis of tweets and 8 different keywords linked to this topic. We then count the number of times these keywords appear in the corpus of tweets.
For example, if LDA identifies a set of eight keywords related to celebration, such as "celebrate," "fun," and "party," we can assess how closely these keywords align with a party emoji. Then, by calculating the number of matching keywords, we can assign a weight to the emoji based on its relevance to the given theme.

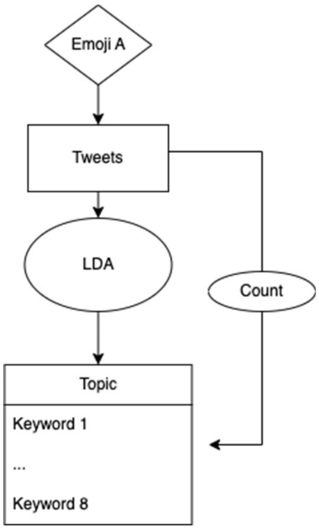

Figure 4 - Topic modeling of tweets with LDA

We try to do the same using BerTopic[7] to see the effects of using transformers.

E. Generation of emoji from text

To translate text to emoji, we use the SentenceToEmoji project, where for a given textual post, an automatically generated emoji is output. To evaluate the results obtained we proceed to use 20K tweets associated with a certain emoji G from the Tweets with Emoji dataset. We predict emoticons on these tweets and count how many emoticons G we get.

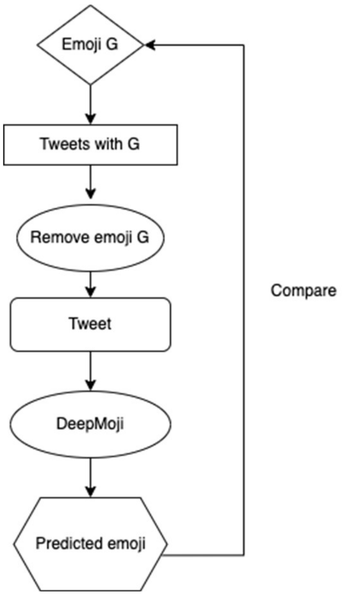

Figure 5 - Emoji prediction from text

F. Antonymy relationship between most frequent words used for antonym emoji

To study the antonymy relationship between most frequent words used for antonym emojis, we use wordNet. From emoji G, we go through the most frequent words and use wordNet to give antonym. We then check if they are present in the most frequent words of the antonym emoji J and count the ratio.

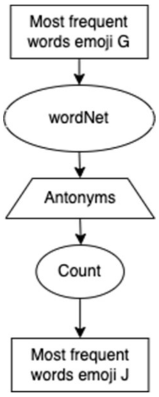

Figure 6 - Antonym relationship between most frequent words of antonym emojis

III. Results and Discussion

A. Relationship between emojis description and emotion

Here are the results obtained from the analysis between emoticons and emotions, using the Word2Vec.

Emoji	**Description**	**Anger**	**Anticipation**	**Disgust**	**Fear**
🌈	Rainbow	0.00	0.28	0.00	0.00
🌙	Crescent Moon	0.00	0.31	0.00	0.00
🌚	New moon face	0.06	0.08	0.17	0.06
🌞	Sun with face	0.00	0.22	0.00	0.00
🌟	Glowing star	0.00	0.28	0.00	0.00
Emoji	Description	Joy	Sadness	Surprise	Trust
🌈	Rainbow	0.69	0.06	0.22	0.33
🌙	Crescent Moon	0.25	0.00	0.06	0.25
🌚	New moon face	0.42	0.19	0.06	0.11
🌞	Sun with face	0.78	0.00	0.11	0.22
🌟	Glowing star	0.53	0.00	0.25	0.31

Figure 7 - Score of emotion for each emoji with Word2Vec

The results obtained seems coherent. Though these emojis are not the most used in written expression, certain emojis (e.g., “🌞” and “🌈”) consistently reflected positive emotions like joy, while more nuanced relationships existed for other emojis.

We decided to use other embedding methods and compare the results obtained with EmoTag1200. Here is a comparative table between Word2Vec, Glove and DistilBERT.

Emotion	Word2Vec	Glove	DistilBERT
Anger	0.36	0.27	0.08
Anticipation	0.22	0.17	-0.03
Disgust	0.36	0.33	0.04
Fear	0.62	0.47	0.11
Joy	0.42	0.38	0.03
Sadness	0.18	0.36	0.15
Surprise	0.22	0.10	-0.10
Trust	0.28	0.21	0.05

Figure 8 - Pearson correlation coefficient with EmoTag1200 results

We considered that a Pearson correlation coefficient superior to 0.2 is enough to show some similarities with EmoTag1200 dataset. We can see that the Word2Vec and Glove results are correlated with the EmoTag1200 scores. On the other hand, DistilBERT prediction does not show good correlation.

We understood from the result that Static embeddings, like Word2Vec and Glove, capture general associations better for short, stable definitions. Meanwhile, BERT-based models focus on syntax and nuanced context, which might not align as well with simple emoticon definitions. Thus, for this task, the first two models are better suited.

B. *Semantic link between antonym emoticons and text*

Here is the table showing how many common words are shared between most frequent words linked to these emojis.

	😡	❤️	😂	😭	🥳	🙂
😡	30	10	20	22	7	15
❤️	10	30	22	11	18	20
😂	20	10	30	22	8	14
😭	22	11	22	30	8	17
🥳	7	18	8	8	30	15
🙂	15	20	14	17	15	30

Figure 9 - Common most frequent words between each emoji

Surprisingly, this study doesn't show clear patterns. As expected, some antonym emojis share fewer common words like partying face and enraged face.

But we can see that joy emoji shares around the same number of words with all the different emojis involved. Maybe that can be explained because joy emoji is quite neutral, it doesn't represent strong emotions.

Although 😂 and 😡 represents antonymic emotions, they share a lot of common words, more than two times the number between 😂 and 🥳 whose relation seems closer.

The data source could be involved in these ambiguous results. Indeed, twitter is a platform where its users tend to use more 2nd degree or sarcasm. In this case, emojis involved may not clearly represent the text content of the tweet.

Consequently, analyzing emoticons in social media texts requires sophisticated handling of context, especially to detect sarcasm accurately.

We also detected that the most frequent words for each emoji are not strongly correlated to the emojis. It is often a general word like "go" or "like".

C. *Semantic meaning of emojis used in tweets (3)*

The following graphs show the embedded tweets linked to certain emoticons on a 2D vectorial plan. We used only a subset of the tweets, because of computing limitations. We tried using different methodologies. First of all, without any text preprocessing, then we applied the same stopwords removal process to the tweets, using the nltk corpus with some additional expressions like 'youre', 'im', 'kinda'. We also tried using different parameters for t-SNE.

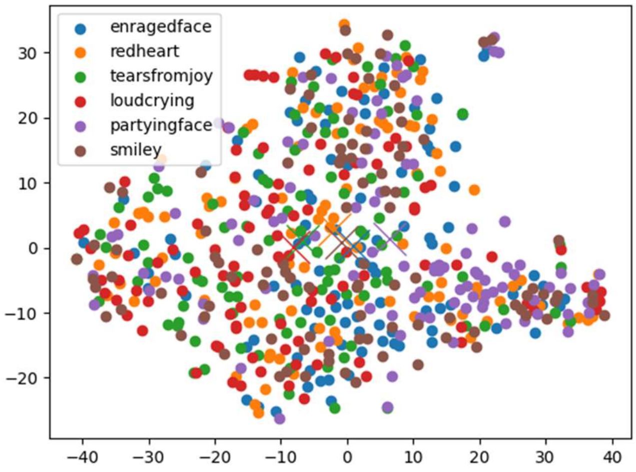

Figure 10 - 2D plot of embedded tweets with DistilBERT

The results are always similar and do not show a clear correlation between the embeddings and the emoticon related to the tweet. We can see that the centroids are on different positions, and that antagonist emoticons are on opposite sides, but we don't get any clear cluster. Given a datapoint, it is very hard to predict the category.

This could be a problem related to DistilBERT or t-SNE. We think that the problem is likely DistilBERT as this would confirm the bad performance it achieved in point A.

The following plot shows the results using Empath embedding.

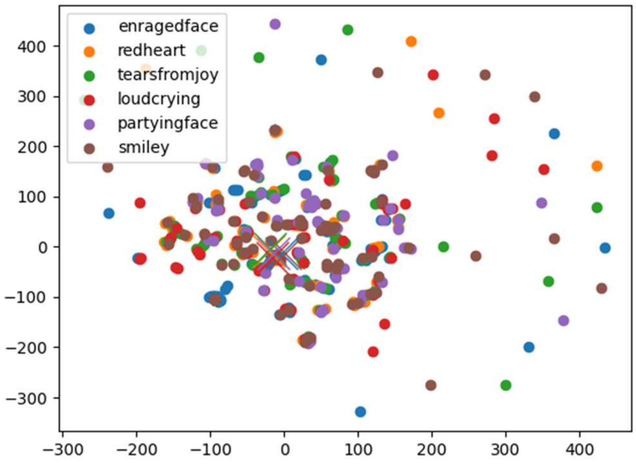

Figure 11 - 2D plot of embedded tweets with Empath

We are using the same number of tweets as before; however, it contains very few distinct data points compared to the previous one. There are many reasons to receive the result, but a direct possibility would be that empath predicts the same embeddings for many different tweets for some reason, causing the overlapping of the data points. This is what has been observed in point B.

Another possible explanation for the bad performance, also for the previous plot, could be the dataset itself. It is coming from twitter and there are several factors that can contribute to the randomness of the points, like sarcasm or noise.

D. Relationship between topics and emojis

Here is a table showing the calculated weights for topic relationship with emojis.

Emoji	weight_lda	weight_bert
enragedface	2599	3236
loudcrying	2892	2062
partyingface	2275	19280
redheart	2112	14468
smiley	2242	7096
tearsfromjoy	2529	3372

Figure 12 - Weights for topic relation with LDA and BerTopic

The results in this case show that, in general, BERTopic can identify topics related to emoji more precisely than LDA in particular for some emojis, while for others, they achieve the same results, and for some, BERTopic is performing slightly better. We can observe that the weights calculated are almost always higher with BERTopic, probably due to its ability to capture more granular emotional associations. But still, we have some emojis with similar weights, which can indicate that they are associated with more straightforward, less context-dependent themes.

This can be explained by the fact that LDA tends to rely on keyword frequency, which might not capture complex contextual meanings, while BERTopic using embeddings is able to capture better the context of the words.

E. Generation of emoji from text

For this task we couldn't get DeepMoji to work, because it has been implemented in an old version of python and PyTorch and it is not compatible anymore.
Instead, we used SentenceToEmoji. The model was trained to output 5 possible emoticons, out of which only 3 were available in the twitter dataset. Therefore, we could only calculate the percentage of correctly generated emojis for 3 categories. Our results show the best performance with "sad" (0.5213) and a slightly lower performance with "heart" (0.4532) and "happy" (0.3435). An explanation for this behavior is that "heart" and "happy" convey similar emotions, while "sad" is the most unique. This makes it harder for the model to distinguish between "happy" and "heart", but it has less troubles with "sad".

Happy	*Heart*	*Sad*
0.3435	*0.4532*	*0.5213*

Figure 13 - Emoji generation accuracy

F. Antonomy relationship between most frequent words used for antonym emoji

The following table shows the antonymy proportion between the emojis through the analysis of their most frequent words.

	😡	❤️	😂	😭	🥳	🙂
😡	0.066	0.033	0.033	0.033	0.033	0.133
❤️	0.100	0.000	0.033	0.033	0.000	0.100
😂	0.100	0.000	0.000	0.000	0.000	0.033
😭	0.100	0.000	0.000	0.000	0.000	0.033
🥳	0.066	0.000	0.033	0.033	0.000	0.066
🙂	0.133	0.033	0.033	0.033	0.000	0.100

Figure 1414 - Antonymy relationship between emoji's most frequent words

The values we get are very low. As explained before, we saw that the most frequent words used are often very common and not specifically related to the emojis. But removing them would be too time-consuming.
We decided to consider the synonyms, hyponyms and hypernyms of the antonyms given by wordNet, and check if they are part of other emojis most frequent words.
For example, the word "make" has only two antonyms with wordNet ("unmake", "brake"). But we could consider that the words "destroy", "demolish" could be considered as antonyms. This way, we include these related words.

	😡	❤️	😂	😭	🥳	🙂
😡	0.333	0.366	0.333	0.333	0.300	0.466
❤️	0.200	0.133	0.166	0.166	0.100	0.233
😂	0.266	0.266	0.233	0.233	0.233	0.300
😭	0.266	0.266	0.233	0.233	0.233	0.300
🥳	0.166	0.100	0.166	0.166	0.066	0.166
🙂	0.2333	0.166	0.200	0.200	0.133	0.233

Figure 15 - Antonymy relationship between emoji's most frequent words using synonyms, hyponyms and hypernyms.

As expected, the proportions obtained with these modifications are bigger.
We can see that 😡 share a large proportion of antonymy with other emojis which seems logical.

IV. Overall Discussions

This research showed several findings about the relationships between emoticons, emotions and text.

When using different embedding models, we confirmed that emojis can bring additional cues for understanding messages, as suggested in [1].
We found that some emojis like “🌞” and “🌈” are strongly linked to positive emotions, like joy.

The comparison between traditional embeddings like Word2Vec and Glove and transformer-based like DistilBERT and BERTopic offer a basis for selecting models in future studies.

Through our work, we also showed that using synonyms, hyponyms and hypernyms increased the antonymy proportions between the emojis. This could be used for further research.

The dataset presented us with some challenges. It comes from twitter, so there is a lot of noise and sarcasm. The data preprocessing was critical but also problematic. Stopwords and the tokenization provided in nltk are surely not enough. A lot of expressions used in the tweets are very colloquial and the preprocessing scheme was not enough. In the first attempts, ‘Im’, ‘youre’, ‘dont’ and others were among the most used words in every category.
Also, in a lot of tweets there were words followed by an emoji (e.g. bye😉), which may be a problem if not handled correctly. The standard preprocessing scheme sometimes had some troubles with it.
So, we resorted to manual intervention to get rid of colloquial expressions and of emojis wrongly considered as part of a token.

Examples of sarcasm:
“Didn’t get no sleep. Now I’m going to attempt a 2 hour nap before clocking into work. 😪” '@hatrrrhater it’s so refreshing to see a bathroom with a bidet bro 🙏🙏😮\u200d💨'
"BU THEN THERE'S KARINA WHO DIDN'T EVEN TOUCH HER GRAVY 😭😭”
They all belong to the partying face category in the dataset.

Examples of emojis sticked to words:

'@Arsenal Aaron ramsdale i love you 🤍😭'
'Aaron Ramsdale 🤍💯'
'Love heals all 🥰’

In literature, sentiment analysis and sarcasm detection have been often discussed as standalone problems.
Only recently, they are starting to get treated together as one, bigger problem. This paper [3] is an example of the two problems being treated as a bigger one. Sarcasm detection shows promising results in improving the sentiment analysis accuracy. Another possibility to get better results in our study is to also consider sarcasm detection in the analysis and treat it as a multi-task problem.

V. CONCLUSION

This project made us work on a real-life problem and exposed us to all the challenges that come from it.
First of all, it made us more comfortable using SOTA models, like wordNet and DistilBERT. We evaluated their performance, and we understood their strengths and their limitations, as well as their ability to handle noisy data.
Then, we have seen how important the data pre-processing is in real life scenarios and the way it can make the difference between good and poor results. We couldn't only rely on existing libraries, but we had to adapt general tools to our needs. For instance, for tokenization and stop words removal, we used the nltk tokenizer and the nltk corpus, but we also had to make some adjustments through trial and error. Additionally, this project highlighted the importance of the quality of the dataset. Using Twitter data provided us with challenges mainly linked to noise and high level of sarcasm you can encounter on this platform.

There are many potential ways for the future works. One study could be done using another platform to get data. Maybe a more formal one, like Slack where we should encounter less second degree and noise.

We also noticed that certain words, such as “like”, “people”, “even”, ... are quite frequent for all emojis. Because they are all valid words that need to be preserved in reserve context, they usually won’t be filtered out by stop words list. It is a possible direction for future work to filter out similar words between each pair of emoji topic, so that we can evaluate the meaning based on the most distinct and common word for each emoji.

VI. REFERENCES

[1] Derks, Bos and Von Grumbkov, “Emoticons in computer-mediated communication: social motives and social context,” *2008*

[2] Walter and D’Addario*, “The Impact of Emoticons on Message Interpretation in Computer-Mediated Communication”, 2001*

[3] Yik Yak Tang et al., “Sentiment Analysis and Sarcasm Detection using Deep Multi-Task Learning”, 2023

[4] A. Shoeb, G. Melo, "EmoTag1200: Understanding the Association between Emojis and Emotions," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 8957–8967.

[5] Abu Awal Md Shoeb, Shahab Raji, Gerard de Melo, "EmoTag – Towards an Emotion-Based Analysis of Emojis," in Proceedings of RANLP 2019, 2019, pp. 1094-1103.

[6] ericwang2001, Dennis Qin, "Tweets With Emoji — kaggle.com,"

[7] M. Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," in arXiv preprint arXiv:2203.05794, 2022.

[8] E. Fast, B. Chen, M. Bernstein, "Empath: Understanding topic signals in large-scale text," in Proceedings of the 2016 CHI conference on human factors in computing systems, 2016, pp. 4647–4657.