A Comparative Study of Language Models for Emoji Recommendations

Maja Jurić[1], Maria Fain[1] and Klara Iličić[1]

[1] Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

Abstract

Purpose – Emojis have become a standard part of communication on social networks, adding depth and nuance to textual interactions. In order to speed up the user's generation of messages, it is necessary to develop systems that will suggest emojis. The developed system should understand the context and identify different emotions in the conversation. Our goal is to develop a system that will be able to recognize the emotion in a sentence and accordingly suggest an emoji to the user.

Design/Methodology/Approach – For training machine and deep learning models, we used a dataset of 9806 sentences classified into 8 emotions (sadness, happiness, anger, fear, shame, disgust, guilt and surprise). The dataset was preprocessed using standard natural language processing methods. The models we used for emotion detection range from the classic machine learning algorithms such as SVM, Logistic Regression and Naïve Bayes to more advanced, deep learning models, precisely bidirectional long short-term memory networks (Bi-LSTM), and the currently most popular language models such as BERT, RoBERTa, LLaMa and Phi.

Findings – The performance of different machine learning and deep learning models was compared in order to gain an insight into which model is best to use in the background of the emoji suggestion system. Well-known language models proved to perform better and more accurately.

Originality/Value – This study not only contributes to the ongoing exploration of emotion classification in sentences but also underscores the practical utility of the developed models by implementing a functional software system. The originality lies in the comprehensive comparison of diverse models and the subsequent application of the selected models to create a software solution capable of providing intelligent emoji suggestions based on contextual understanding and emotional nuances in conversations.

Keywords – Machine Learning; Deep Learning; Emotion Classification; Emoji Recommendation; Natural Language Processing; Language Models.

Paper Type – Research paper.

1 Introduction

Emojis are small digital images or icons used to express or highlight an emotion or idea. They are widely popular and utilized by many digital users worldwide on different social platforms such as Facebook, Twitter, Instagram and many others. Emojis have become a standard part of text messages, statuses and tweets, making them more vivid and enhancing words or picturing ideas that can hardly be expressed with words. Moreover, some users prefer using them over plain text because they can express more words with just a single icon. The number of available emojis keeps expanding over the years, making it somewhat difficult for users to choose the emoji which fits their message best. This is where the need for an emoji recommender which we implemented comes from.

In this paper, we aim to give automated suggestions of emojis which are most relevant given the input plain text of a user. Our recommendations are based on emotion detection as well as content analysis. For the emojis that represent emotions, we have implemented several language models for which we gave descriptions and compared their results on emotion detection. After a certain emotion was detected in the text by our models, the emojis representing that emotion are suggested. As for the content analysis, we simply search the input plain texts for words or pairs of words that fit the names of certain emojis and include them in the recommendation as well. These emojis are more descriptive of ideas or situations, rather than emotions.

The models we used for emotion detection range from the classic machine learning algorithms such as SVM, Logistic Regression and Naïve Bayes to more advanced, deep learning models, precisely bidirectional long short-term memory networks (Bi-LSTM), and the currently most popular language models such as BERT, RoBERTa, LLaMa and Phi. We will describe how each of the models tackles the challenge of emotion detection, compare their results and provide ideas on how to improve their results or why some models give better accuracy than others.

2 Related work

In recent history, there has been an increase in research of social media and the language used in it. Emojis are a new addition to the language used in online conversations and posts, making them specifically interesting to natural language processing (NLP) researchers.

2.1 Emoji Recommender Systems

Multiple studies have been conducted on emoji recommendations and predictions, usually also exploring the history of emojis and their use in different age groups, genders, etc. In their work, (Zheng, Zhao, Zhu, & Qian, 2022) pay special attention to user history and base their model on dynamic user preferences showing that different users may have different preferences or habits when using emojis. On the other hand, (Xie, Liu, Yan, & Sun, 2016) focus only on the contextual information of the dialogue between users. They use a hierarchical long short-term memory model to construct dialogue representations and then use a classifier for emoji classification. This approach is used when user information is not available or difficultly accessed. A solution which combines the two aforementioned recommender approaches is given by (Zhao, Liu, Chao, & Qian, 2021). They have proposed fusing the personal user information with context. They combine the two via a score-ranking matrix factorization framework. Their work considers user history as well as other features such as gender which gives even more dimensionality to their emoji recommender system.

Other research battles with problems such as making emoji recommendations for multilingual texts, such as English and Spanish in (Barbieri, et al., 2018). Similarly, with the explosive growth of a Twitter-like social platform Weibo in China, (Zhao, Dong, Wu, & Xu, 2012) decided to explore the language used in the platform. Realizing the Chinese corpus was small compared to the more popularly researched English one, they decided to extend it by mapping emoticons into categories of sentiment and then analyzing the sentiment in tweets and predicting the corresponding emojis.

2.2 Emotion Recognition

Emotion recognition in text is a highly researched area of NLP which we have implemented for our solution for making emoji suggestions. This is why emotion recognition in dialogue is of interest in our research. (Zhang, et al., 2023) have proposed a context and emotion knowledge tuned large language model (LLM) which they have obtained by fine-tuning LLMs with benchmarking multi-modal emotional dialogues. They state that LLMs are not specifically designed for emotion understanding tasks. Similar research has been conducted by (Chudasama, et al., 2022), also exploring the audio, video and transcript parts of online conversations. They have introduced a multi-modal fusion network. This kind of text analysis was a part of the motivation behind our emoji analysis.

Other research proposes hybrid models which combine machine learning and deep learning models to identify emotions in text. A model like this is proposed by (Bharti, et al., 2022), consisting of a deep learning part, for which they considered convolutional neural networks and Bi-GRU, and a machine learning approach for which support vector machine was used. They achieved greater accuracy than our separate machine learning and deep learning models. (Ragheb, Azé, Bringay, & Servajean, 2019) is research that also incorporated self-attention mechanisms to focus on the most important parts of the texts they were analyzing for emotion recognition. The use of attention has given us the idea of using state-of-the-art mechanisms such as BERT for our models.

(Seal, Roy, & Basak, 2019) have implemented a keyword-based approach focused on phrasal verbs for detecting emotion in text. This gave us the idea to combine an emotion recognition model with a keyword-based approach for suggesting emojis.

3 Methodology

Emoji recommendation aims to give appropriate emojis attached to the sentences according to the contextual information. We've simplified the emoji recommendation task by treating it as emotion classification. We start by understanding the emotion in the message and then predict the appropriate emoji(s).

3.1 Dataset Description

In this paper we have combined two datasets: the ISEAR[1] dataset and the Emotion Stimulus[2] dataset.

The ISEAR dataset consists of over 7000 entries classified into 7 distinct emotions: sadness, anger, happiness, fear, shame, guilt and disgust. The Emotion Stimulus dataset contains over 2000 entries classified into 7 emotions: sadness, anger, happiness, fear, surprise, shame and disgust. After merging the two datasets, we have one dataset with over 9000 entries classified into 8 emotions (sadness, happiness, anger, fear, shame, disgust, guilt and surprise). The histogram depicting the distribution of classes is illustrated in *Figure 1*.

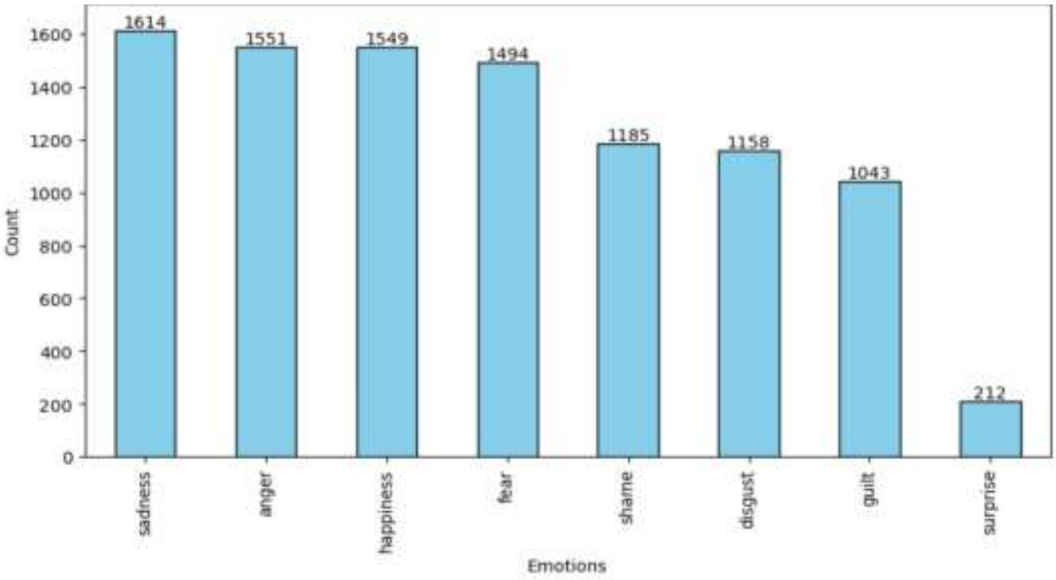
3.2. Dataset Preprocessing

Before we fed the data into the models, we took a crucial step to prepare it. This involved cleaning the text by removing links and mentions, getting rid of stop words and lemmatizing the text. These preprocessing steps were essential for refining the input data, ensuring a cleaner and more effective representation for subsequent model training and analysis. It's worth noting that each model requires its own way of preprocessing and we'll discuss these details in *3.3*

[1] ISEAR Dataset: https://github.com/sinmaniphel/py_isear_dataset

[2] Emotion Stimulus Dataset: https://www.kaggle.com/datasets/parulpandey/emotion-dataset

Figure 1 Emotion Distribution in the Dataset



Experiment settings and implementation details. This ensures that the input data aligns well with the demands of each model.

3.2 Theoretical Background: Emotion Classification Models

This chapter aims to provide a detailed overview of all the models used - from traditional machine learning algorithms like Naive Bayes, Logistic Regression and Support Vector Machines, to advanced neural network architectures such as Bi-LSTM, and state-of-the-art transformer-based models including BERT, RoBERTa and Llama.

3.2.1 Naïve Bayes

Naïve Bayes is a probabilistic machine learning algorithm based on Bayes' theorem. It is commonly used for classification tasks, including text classification, spam filtering, and sentiment analysis. The "naive" aspect of Naïve Bayes arises from the assumption of independence among features, meaning that the presence or absence of one feature does not affect the presence or absence of another (Ray, 2024). Despite its simplicity, Naïve Bayes often performs surprisingly well in various real-world scenarios.

In essence, Naive Bayes calculates the probability of a particular event or classification given the presence of certain features. It's particularly well-suited for problems with a large number of features, making it efficient and computationally inexpensive.

For emotion classification, we used Multinomial Naïve Bayes. Multinomial Naïve Bayes calculates each tag's likelihood for a given sample and outputs the tag with the greatest chance which is appropriate for the emotion classification task (Sriram, 2022).

3.2.2 Logistic Regression

Logistic Regression is a widely used statistical and machine learning algorithm for binary and multiclass classification tasks. Despite its name, logistic regression is used for classification, not regression. It's a linear model that predicts the probability of an instance belonging to a particular class.

The logistic regression model applies the logistic function (also called the sigmoid function) to transform the output into a range between 0 and 1, representing probabilities. The decision boundary is set based on a chosen threshold (often 0.5), classifying instances with predicted probabilities above the threshold as one class and below as the other (Akash, 2022).

In this paper we used Multiclass Logistic Regression, also known as Softmax Regression. Multiclass Logistic Regression is an extension of binary logistic regression to handle scenarios with more than two classes. It's a popular algorithm for multiclass classification tasks where the goal is to assign an instance to one of several possible classes.

Similar to binary logistic regression, multiclass logistic regression applies the softmax function to model the probabilities of an instance belonging to each class. The softmax function converts the raw class scores into probabilities, ensuring that the sum of probabilities across all classes equals 1 (Yang, 2021).

3.2.3 SVM

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for both binary and multiclass classification tasks, as well as regression tasks. SVM uses a kernel trick to map the input features into a higher-dimensional space, allowing it to handle non-linear decision boundaries. SVM aims to find a hyperplane in the feature space that best separates instances of different classes. The optimal hyperplane is the one that maximizes the margin, which is the distance between the hyperplane and the nearest instances of each class, known as support vectors (Saini, 2024).

For multiclass classification, SVM can be extended using one-vs-one or one-vs-all strategies. In the one-vs-one approach, a binary SVM is trained for each pair of classes, and the class with the most "votes" is selected. In the one-vs-all approach, a binary SVM is trained for each class against the rest, and the class with the highest confidence is chosen.

3.2.4 Bidirectional Long Short-Term Memory

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN). Unlike RNNs, LSTMs have an additional forget gate which allows them to capture long-term dependencies in sequential data. The forget gate allows them to selectively keep or forget information from the previous cell states using the sigmoid function. This is why they are often used in NLP, because of their ability to take context into account.

Bidirectional LSTMs process sequential data in both forward and backward directions making them even more useful in NLP because they can capture both the past and future context in a sentence or text. They are essentially a combination of two layers of LSTMs, one which processes the sequence in the forward direction and another which processes the same sequence in the backward direction. These two layers work simultaneously, but have their own hidden states and memory cells, functioning like two separate LSTM networks. Once the forward and backward passes are complete, the hidden states of both layers are combined at each time step.

Even though the ability to process the input sequence in both directions is their best feature, it is also the reason Bi-LSTMs can have high computational costs.

3.2.5 BERT

BERT, short for Bidirectional Encoder Representations from Transformers, is a state-of-the-art natural language processing model developed by Google. It belongs to the Transformer architecture family and has revolutionized various NLP tasks, including text classification, named entity recognition and question answering.

BERT is particularly powerful in text classification tasks due to its bidirectional contextualized embeddings, allowing it to capture intricate relationships and dependencies within sentences. Unlike traditional models that read text sequentially, BERT processes the entire context simultaneously, considering both left and right context words for each token (Devlin et al, 2019).

3.2.6 RoBERTa

RoBERTa, short for Robustly optimized BERT approach, is a variant of the BERT (Bidirectional Encoder Representations from Transformers) model. Developed by Facebook AI Research (FAIR), RoBERTa builds upon BERT's architecture and training methodology, introducing modifications to enhance its performance.

Similar to BERT, RoBERTa is highly effective for classification tasks due to its bidirectional context understanding and contextualized embeddings. In a classification scenario, RoBERTa can be fine-tuned to adapt its pre-trained knowledge to specific tasks. The process involves incorporating task-specific parameters into the model and then training it on the classification data (Efimon, 2023).

RoBERTa's modifications, including removing the Next Sentence Prediction (NSP) objective and training with larger mini-batches, contribute to its robustness and improved performance over the original BERT model.

3.2.7. Llama

Llama is a family of LLMs that was first released in February 2023 by Meta AI. LlamA introduces language models ranging from 7B to 65B parameters, trained exclusively on publicly available datasets. Their researchers said that LLaMA-13B outperforms GPT-3 (175B) on most benchmarks, and LLaMA-65B competes with top models like Chinchilla70B and PaLM-540B. (Touvron, et al., 2024)

Then, in July, Llama 2 was published, which we also decided to use in this paper. Llama 2 is a collection of pretrained and fine-tuned large language models (LLMs) ranging in scale from 7 billion to 70 billion parameters. Fine-tuned models called Llama 2-Chat are optimized for dialogue use cases. Their researchers said that the models surpass open-source chat models in most benchmarks, and according to human evaluations for helpfulness and safety, they could serve as an alternative to closed-source models. All models are released to the research community and are completely free. (Touvron, et al., 2024)

3.2.8. Phi

In recent months, Microsoft Research's Machine Learning Foundations team has unveiled a series of small language models (SLMs) known as "Phi," achieving impressive performance across various benchmarks. The team introduced Phi-1, a 1.3 billion-parameter model, which excelled in Python coding benchmarks. Expanding their focus to common sense reasoning, they developed Phi-1.5, a model of comparable performance to much larger counterparts. Now, the team has released Phi-2, a 2.7 billion-parameter language model showcasing exceptional reasoning and language understanding. Phi-2 matches or surpasses models up to 25x larger on complex benchmarks, thanks to innovations in scaling and data curation. With its compact size, Phi-2 serves as an ideal playground for researchers.

We decided to try Phi-2 because their researchers claim that Phi-2 achieves better performance compared to 25x larger Llama-2-70B model on muti-step reasoning tasks, i.e., coding and math. (Javaheripi & Bubeck , 2024)

3.3 Experiment settings and implementation details

In this chapter, we'll discuss how we set up and implemented each model for our experiments. Our aim is to provide a clear and detailed account of how we conducted our experiments, making it easy for readers to follow our process and understand the basis of our results and conclusions.

3.3.1 Traditional machine learning models

We employed three traditional machine learning models - Multinomial Naïve Bayes, Logistic Regression and Linear Support Vector Machine (SVM) - to classify emotions in textual data.

For preprocessing, the original text was utilized as features (X), while the corresponding emotion labels served as the target variable (Y). The textual features underwent vectorization, transforming them into a format suitable for model training. The dataset was divided into training and test sets using an 80:20 split. Logistic Regression utilized the 'lbfgs' solver and 'auto' setting for multi-class classification. Linear SVM employed a tolerance value of 10^{-5}.

Performance metrics, including accuracy and F1 score, were calculated and confusion matrices were displayed to provide insights into the models' classification capabilities. Traditional machine learning models establish a baseline for comparison with more complex models.

3.3.2 Bi-LSTM

The original dataset was split into a training, validation and test set using a stratified split method to ensure a representative amount of data for each target class. The training set comprised 60% of the data, while the validation and test sets constituted 20% each.

LSTMs require each input sequence to be of the same token length which is why we needed to have a few more steps of preparation before feeding the data to the network. The preparation included encoding the target classes, tokenizing the texts, choosing the appropriate maximum sequence length and then padding the smaller sequences and truncating the larger ones. For word embeddings, we used GloVe[3] pre-trained word vectors, specifically glove.6B.

Then we built our sequential model which consisted of the embedding layer, followed by two bidirectional layers and a dense layer. The optimizer used was Adam with a learning rate of 0.006. The model was set to fit on 30 epochs but finished computing in only 9 because of the early stopping callback that was used.

The model was then evaluated on the validation and test sets using accuracy and loss, and a more detailed evaluation using a classification report and a confusion matrix.

[3] https://nlp.stanford.edu/projects/glove/

3.3.3 BERT and RoBERTa

We employed state-of-the-art transformer-based models BERT and RoBERTa. The dataset was split into training, validation and test sets using a stratified shuffle split strategy to ensure a representative distribution of emotions in each subset. The training set comprised 60% of the data, while the validation and test sets constituted 20% each.

Before passing the input text to the BERT model, it has to be specifically preprocessed for BERT. For that we used the *text.texts_from_array* function from the *ktrain*[4] library with the preprocess mode set to ‘bert’. The process involves tokenizing text into subword units, combining sentences, trimming content to a fixed size and extracting labels for the masked language modeling task. The BERT model was fine-tuned for 10 epochs using a one-cycle learning rate policy.

On the other hand, for RoBERTa, the dataset was converted into a custom *PyTorch* dataset, incorporating the RoBERTa tokenizer. The RoBERTa model was then employed as the backbone for a classification head with one fully connected layer. The model was trained for 10 epochs with a learning rate of $2*10^{-5}$.

Both models were evaluated on the validation set using accuracy and loss metrics. The performance was further assessed on the test set - classification reports, including precision, recall and F1-score, were generated. The confusion matrices provided a visual representation of the models' classification performance across different emotion categories.

3.3.4 Llama

We split the data in the same way as in the previous two cases when we worked with Bi-LSTM, BERT and RoBERT models. This means that the original dataset was split into a training, validation and test set using a stratified split method. The training set comprised 60% of the data, while the validation and test sets constituted 20% each.

We had to change the dataset to be able to use Llama. We transformed the dataset into prompts. Train and test prompts have the same structure, the only thing is that train prompts contain the exact answer with which we will want to fine-tune the model.

The decision was to use Meta’s official Llama-2 model from Hugging Face[5]. Out of 12 different models, the Llama-2-7b-hf model was chosen due to time efficiency. Due to the efficiency of working with LLMs, 4-bit quantization was performed. Efficient fine-tuning of LLMs can be achieved using QLoRA[6]. It was necessary to adjust the quantization parameters that we will later use when loading the model. We create 4-bit quantization with NF4 type configuration which does not change model performance.

Firstly, we load the tokenizer for the Llama-2 language model, then set the padding token as the end-of-sequence (EOS) token, and finally, configure the padding side to be "right," ensuring correct padding direction, as required by Llama 2.

For the sake of comparison, we tested how the base model behaves on our dataset. Accuracy was a low 0.420.

[4] Ktrain library: https://pypi.org/project/ktrain/0.1.6/

[5] https://huggingface.co/meta-llama

[6] https://github.com/artidoro/qlora

Fine-tuning of pre-trained language models (PLMs) involves updating all model parameters, demanding substantial computational resources and extensive data. So, we used Parameter-Efficient Fine-Tuning (PEFT)[7] which operates by selectively updating a small subset of the model's parameters, significantly improving efficiency. The training process was also optimized. The model was trained for 10 epochs with a learning rate of $2*10^{-4}$ and the batch size per device during training was set to 1. During fine-tuning, we monitored training loss and validation loss in each epoch.

The model was evaluated on the validation set using accuracy metric. The performance of the model was further assessed on the test set - classification reports, including precision, recall and F1-score, were generated. The confusion matrices provided a visual representation of the models' classification performance across different emotion categories.

3.3.5 Phi-2

We split the data in the same way as in the previous two cases when we worked with Bi-LSTM, BERT, RoBERT and Llama models. This means that the original dataset was split into a training, validation and test set using a stratified split method. The training set comprised 60% of the data, while the validation and test sets constituted 20% each.

We had to change the dataset to be able to use Phi. We transformed the dataset into prompts. Train and test prompts have the same structure, the only thing is that train prompts contain the exact answer with which we will want to fine-tune the model.

The decision was to use Microsoft’s official Phi-2 model from Hugging Face. We did the quantization as with the Llama model. Firstly, we load the tokenizer for the Phi-2 language model, then set the padding token as the end-of-sequence (EOS) token, and finally, configure the padding side to be "right," ensuring correct padding direction.

For the sake of comparison, we tested how the base model behaves on our dataset. Accuracy was a low 0.381.

Fine-tuning of pre-trained language models (PLMs) involves updating all model parameters, demanding substantial computational resources and extensive data. So, we used Parameter-Efficient Fine-Tuning (PEFT) which operates by selectively updating a small subset of the model's parameters, significantly improving efficiency. The training process was also optimized. The model was trained for 10 epochs with a learning rate of $2*10^{-4}$ and the batch size per device during training was set to 1. During fine-tuning, we monitored training loss and validation loss in each epoch with a learning rate of $2*10^{-4}$.

The model was evaluated on the validation set using accuracy metric. The performance of the model was further assessed on the test set - classification reports, including precision, recall and F1-score, were generated. The confusion matrices provided a visual representation of the models' classification performance across different emotion categories.

4 Results

In this section, we share the results of our research with a specific focus on comparing outcomes from different models. This comparative approach provides a clear and

[7] https://huggingface.co/docs/peft/index

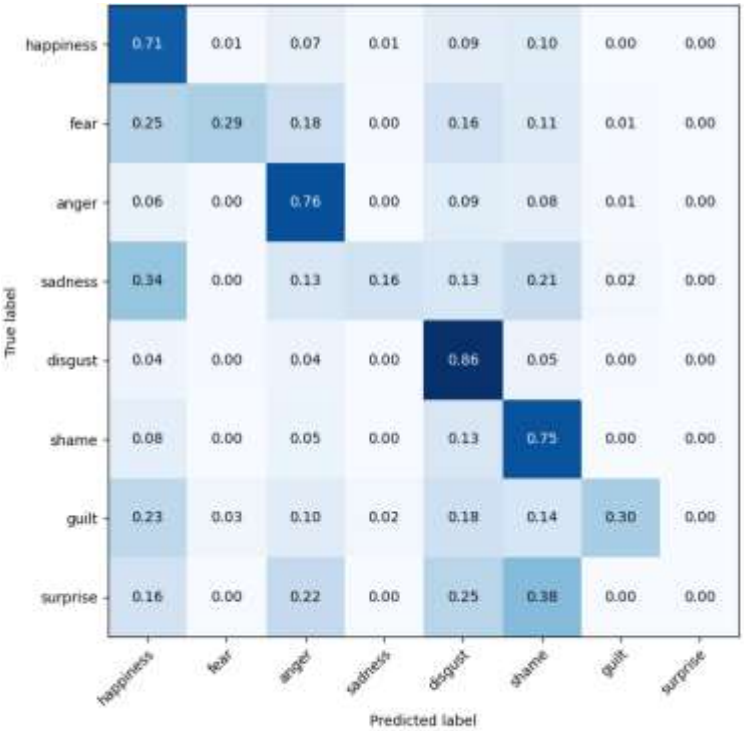
straightforward overview of our findings, offering insights that can be valuable for practical applications and future research within this field.

Accuracy serves as a fundamental metric, reflecting the model's ability to correctly classify instances. We calculated accuracy by dividing the number of correctly predicted instances by the total number of instances in our dataset.

4.1 Naïve Bayes model

The accuracy of the Naïve Bayes model stands at 58%. The confusion matrix suggests that the Naïve Bayes model demonstrates a relatively strong ability to correctly classify sentences conveying disgust and shame. However, the model exhibits some confusion when distinguishing between surprise and shame, as well as between sadness and happiness, which is noteworthy.

Figure 2 Confusion matrix for the Naive Bayes model



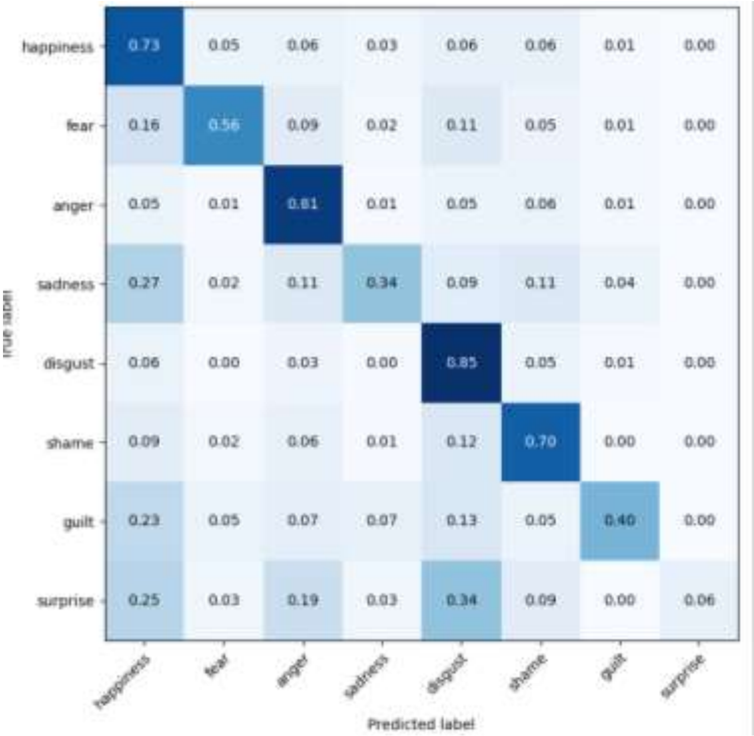
	happiness	fear	anger	sadness	disgust	shame	guilt	surprise
happiness	0.71	0.01	0.07	0.01	0.09	0.10	0.00	0.00
fear	0.25	0.29	0.18	0.00	0.16	0.11	0.01	0.00
anger	0.06	0.00	0.76	0.00	0.09	0.08	0.01	0.00
sadness	0.34	0.00	0.13	0.16	0.13	0.21	0.02	0.00
disgust	0.04	0.00	0.04	0.00	0.86	0.05	0.00	0.00
shame	0.08	0.00	0.05	0.00	0.13	0.75	0.00	0.00
guilt	0.23	0.03	0.10	0.02	0.18	0.14	0.30	0.00
surprise	0.16	0.00	0.22	0.00	0.25	0.38	0.00	0.00

4.2 Logistic Regression model

The accuracy of the Logistic Regression model stands at 64.32%.

Delving into the confusion matrix, we find that the model excels in accurately classifying sentences expressing disgust, shame, and happiness. Yet, it encounters challenges in distinguishing between surprise and disgust.

Figure 3 Confusion matrix for the Logistic Regression model



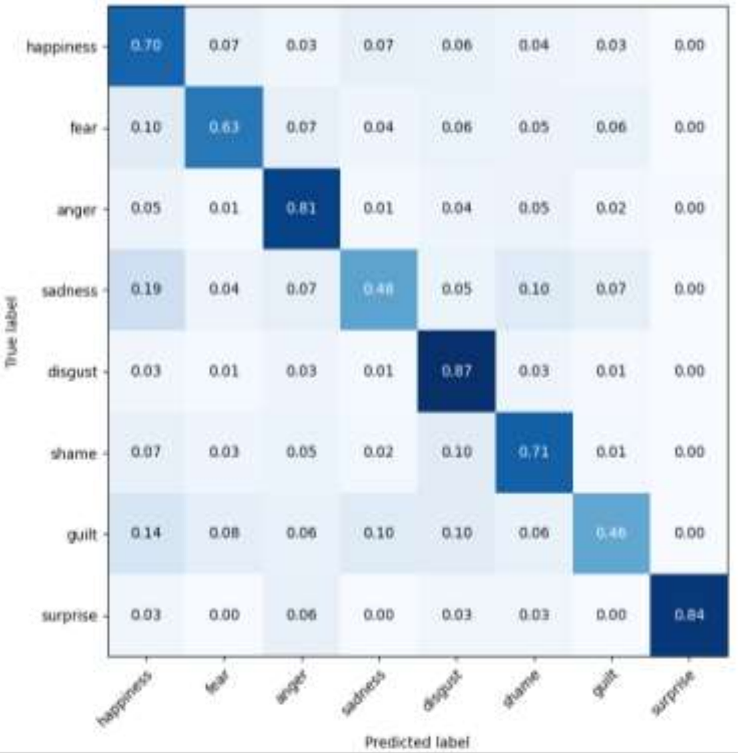
	happiness	fear	anger	sadness	disgust	shame	guilt	surprise
happiness	0.73	0.05	0.06	0.03	0.06	0.06	0.01	0.00
fear	0.16	0.56	0.09	0.02	0.11	0.05	0.01	0.00
anger	0.05	0.01	0.81	0.01	0.05	0.06	0.01	0.00
sadness	0.27	0.02	0.11	0.34	0.09	0.11	0.04	0.00
disgust	0.06	0.00	0.03	0.00	0.85	0.05	0.01	0.00
shame	0.09	0.02	0.06	0.01	0.12	0.70	0.00	0.00
guilt	0.23	0.05	0.07	0.07	0.13	0.05	0.40	0.00
surprise	0.25	0.03	0.19	0.03	0.34	0.09	0.00	0.06

4.3 SVM model

The SVM model achieves an accuracy of 68.6%.

Upon closer examination of the confusion matrix, we find that the model excels in precisely categorizing sentences conveying emotions of disgust, surprise, and anger

Figure 4 Confusion matrix for the SVM model



	happiness	fear	anger	sadness	disgust	shame	guilt	surprise
happiness	0.70	0.07	0.03	0.07	0.06	0.04	0.03	0.00
fear	0.10	0.63	0.07	0.04	0.06	0.05	0.06	0.00
anger	0.05	0.01	0.81	0.01	0.04	0.05	0.02	0.00
sadness	0.19	0.04	0.07	0.48	0.05	0.10	0.07	0.00
disgust	0.03	0.01	0.03	0.01	0.87	0.03	0.01	0.00
shame	0.07	0.03	0.05	0.02	0.10	0.71	0.01	0.00
guilt	0.14	0.08	0.06	0.10	0.10	0.06	0.46	0.00
surprise	0.03	0.00	0.06	0.00	0.03	0.03	0.00	0.84

4.4 Bi-LSTM

The Bi-LSTM model demonstrates a varied performance across different emotion categories, as evident from the precision, recall, and f1-score metrics. Notably, the model excels in accurately identifying instances of surprise with a high precision, recall, and f1-score, each reaching 0.91. It also exhibits strong performance in recognizing emotions such as fear and happiness, with precision, recall, and f1-scores ranging from 0.71 to 0.76 for fear and 0.66 to 0.74 for happiness. However, challenges arise in effectively distinguishing guilt and anger, which can also be seen from the confusion matrix.

The overall accuracy of the Bi-LSTM model stands at 65%, showcasing its capacity to correctly predict emotions across the entire dataset.

4.5 BERT

The results from the BERT model reveal a generally robust performance across most emotion categories. Notably, the model demonstrates high accuracy in predicting instances of surprise, achieving a remarkable f1-score of 0.95. Additionally, it exhibits strong capabilities in identifying instances of happiness, fear, sadness, and anger. From the confusion matrix we see that the model often confuses disgust, shame and guilt for anger.

The overall accuracy of the BERT model stands at 71%, indicating its effectiveness in correctly classifying emotions within the given dataset.

Figure 6 Confusion matrix for the BiLSTM model

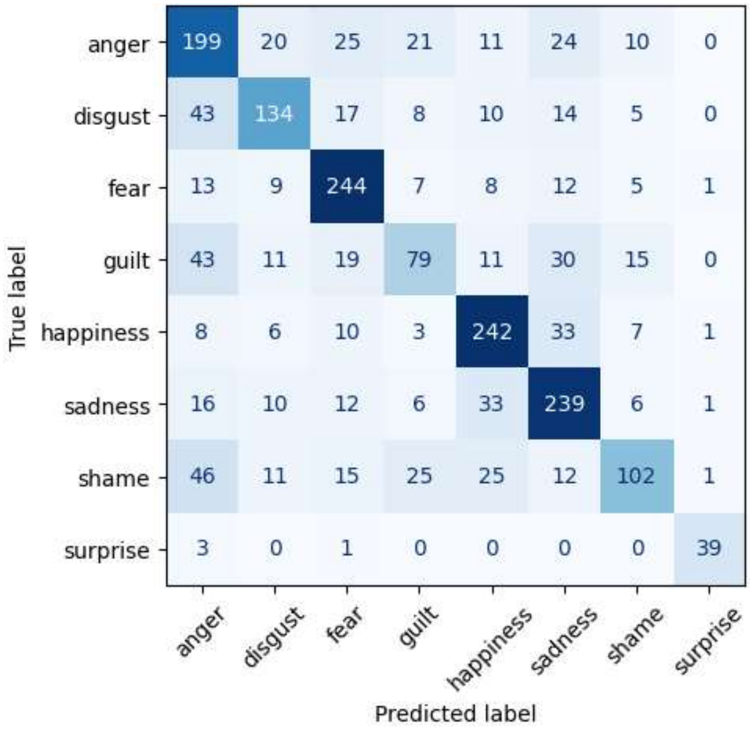
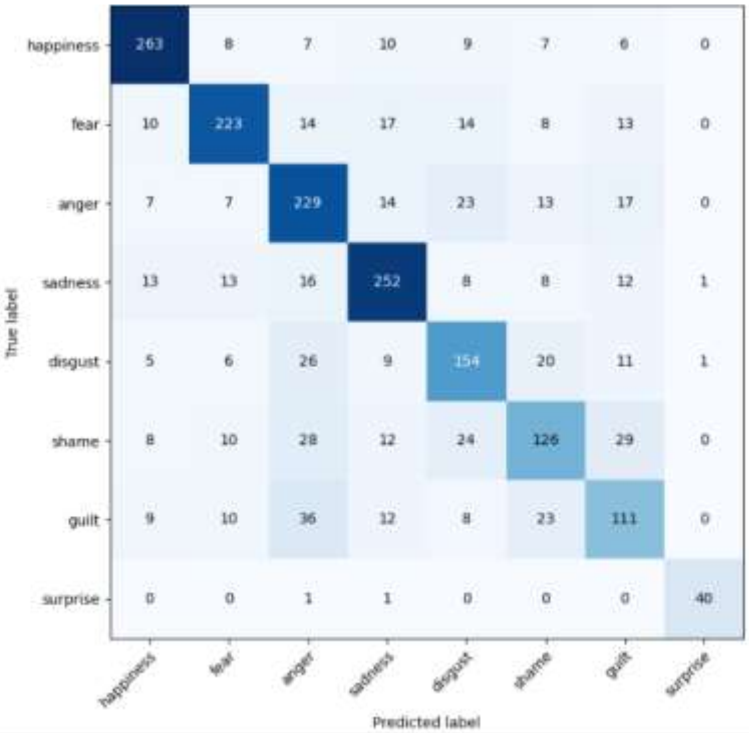


Figure 5 Confusion matrix for the BERT model

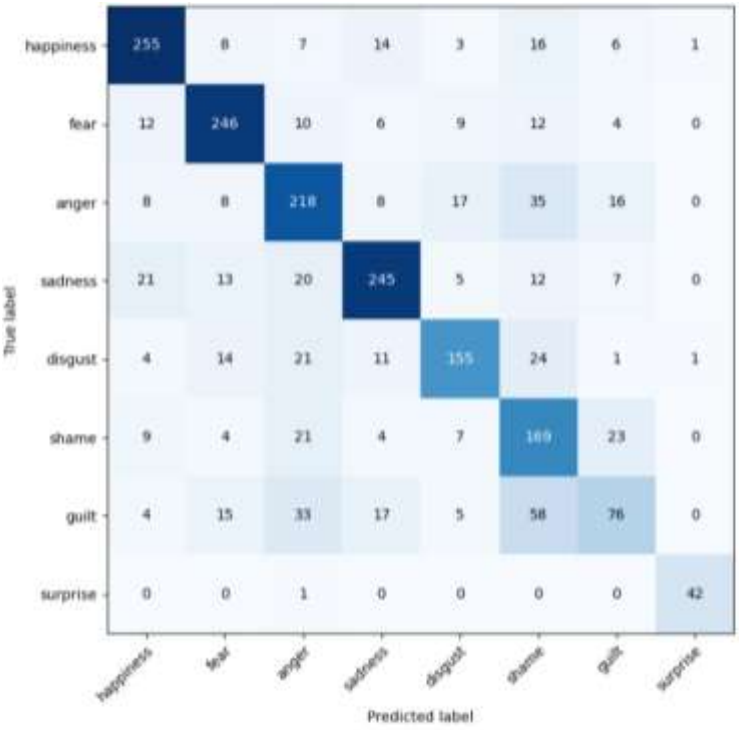


4.6 RoBERTa

The performance of the RoBERTa model in emotion classification is commendable, particularly in predicting instances of surprise, where it achieves a remarkable f1-score of 0.97. The model also demonstrates strong capabilities in accurately identifying emotions such as happiness, fear, and sadness. However, it is notable that the f1-score for instances of guilt is comparatively low. From the confusion matrix, we see that the model often confuses guilt for anger or shame.

The overall accuracy of the RoBERTa model stands at 72%, only slightly better than the BERT model.

Figure 7 Confusion matrix for the RoBERTa model



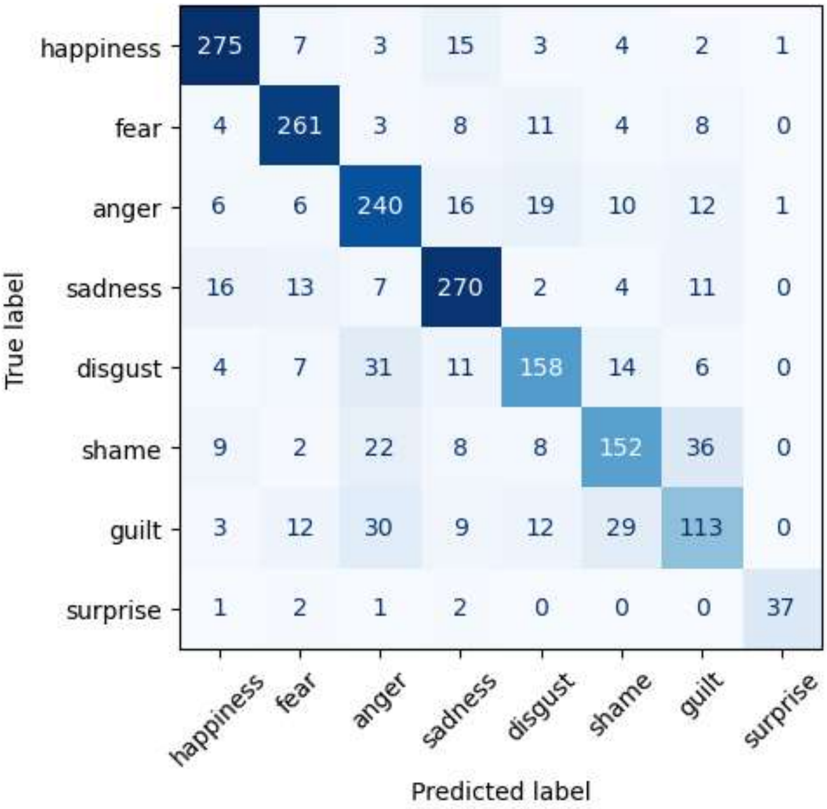
4.7 Llama

The Llama model demonstrates a robust overall accuracy of 77% which signifies its effectiveness in classifying instances within the dataset.

When examining accuracy for individual labels, the model showcases notable performance, particularly excelling in accurately predicting instances for happiness, fear, sadness and surprise with accuracy scores around 88%. These high accuracy values suggest the model's proficiency in handling specific emotion categories.

From the confusion matrix, we see that the model often confuses guilt for anger or shame and shame for guilt. Also, the f1-score for instances of guilt is comparatively low.

Figure 8 Confusion matrix for the Llama model



	happiness	fear	anger	sadness	disgust	shame	guilt	surprise
happiness	275	7	3	15	3	4	2	1
fear	4	261	3	8	11	4	8	0
anger	6	6	240	16	19	10	12	1
sadness	16	13	7	270	2	4	11	0
disgust	4	7	31	11	158	14	6	0
shame	9	2	22	8	8	152	36	0
guilt	3	12	30	9	12	29	113	0
surprise	1	2	1	2	0	0	0	37

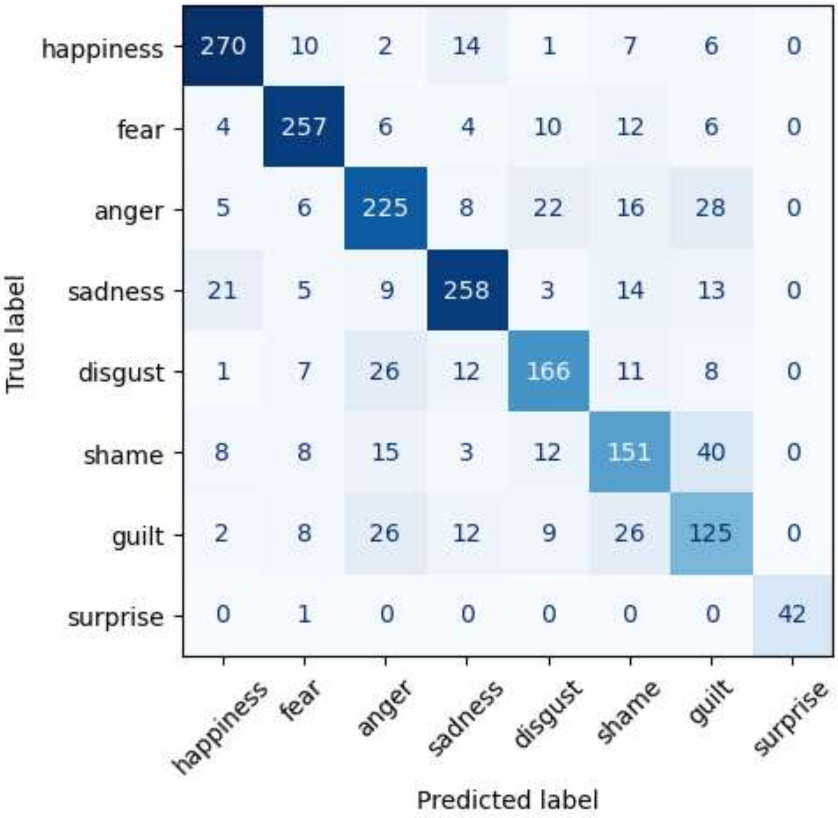
4.8 Phi

The Phi-2 demonstrates a robust overall accuracy of 76% which signifies its effectiveness in classifying instances within the dataset.

When examining accuracy for individual labels, the model showcases notable performance, particularly excelling in accurately predicting instances for happiness, fear and surprise with accuracy scores in the range of 87% to 97%. These high accuracy values suggest the model's proficiency in handling specific emotion categories.

. From the confusion matrix, we see that the model sometimes doesn't distinguish between guilt, shame and anger. Also, the f1-score for instances of guilt is comparatively low.

Figure 9 Confusion matrix for the Phi model



4.9 Overview

The experimental results on emotion classification are demonstrated in Table 1.

The Naïve Bayes model yields a baseline accuracy of 58%, while the Logistic Regression model improves accuracy to 64.32%. The SVM model further enhances accuracy to 68.6%. Moving into neural network models, the Bi-LSTM achieves a 65% accuracy, followed by BERT and RoBERTa models with 71% and 72%, respectively. The Llama model stands out with the highest accuracy at 77%, indicating its robustness in accurately classifying emotions. Phi, with an accuracy of 76%, proved to be almost as effective as Llama.

The results mostly align with our expectations. These results highlight the incremental improvements in accuracy as more sophisticated models are employed with the Llama model exhibiting the highest overall performance. However, we expected that the RoBERTa model would have a significantly higher accuracy score than the BERT model as we suggested in *3.2.6 RoBERTa*.

Table 1 Model accuracy

Model	Naïve Bayes	Logistic Regression	SVM	Bi-LSTM	BERT	RoBERTa	Llama	Phi
Accuracy	58%	64.32%	68.6%	65%	71%	72%	77%	76%

5 Integrating Emotion Classification Models into an Emoji Recommender System

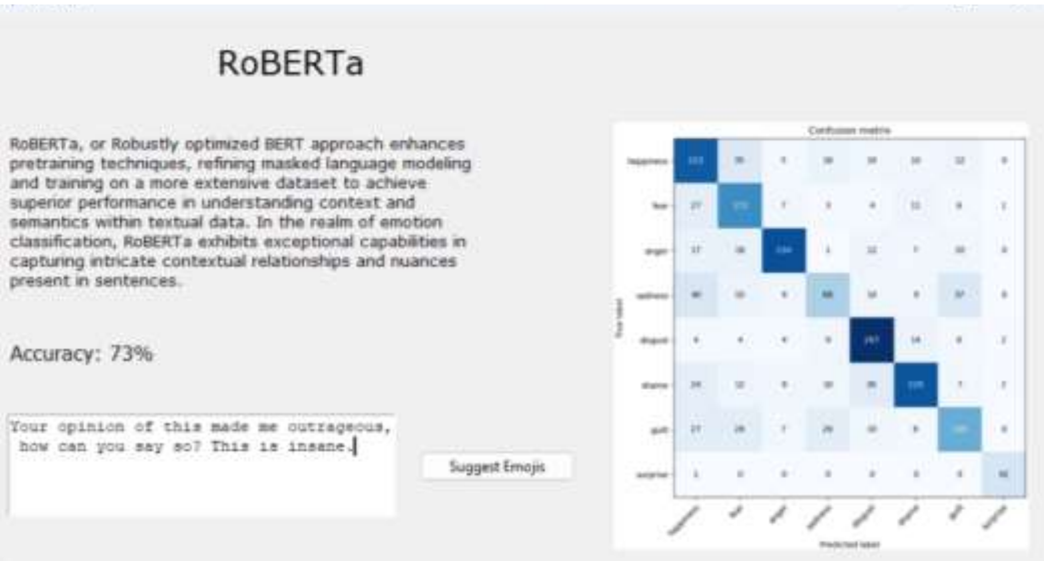
In this chapter we describe how we used the Emotion Classification models within our Emoji Recommender System.

Our recommender system utilizes both the emotional context derived from the text and the text itself to curate personalized emoji recommendations. By employing various Emotion Classification models, our Recommender System strives to accurately identify emotions in the text and propose fitting emojis.

The mechanism operates by associating key emotions with specific keywords. For instance, the emotion 'sadness' corresponds to keywords like 'sad', 'cry', 'disappointed', and 'anxious'. Once the system determines the emotion in a given text, it dynamically searches for emojis based on these associated keywords. Additionally, the system suggests emojis based on keywords present within the text itself, fostering a comprehensive and nuanced recommendation process.

To facilitate emoji searches, we harnessed the capabilities of the advertools library. This library efficiently aids in exploring and obtaining emojis based on predefined keywords. Furthermore, for the display of emojis, we employed the emoji library, contributing to the user-friendly visualization of the recommended emojis.

Figure 10 Using the RoBERTa model for Emoji Recommendation



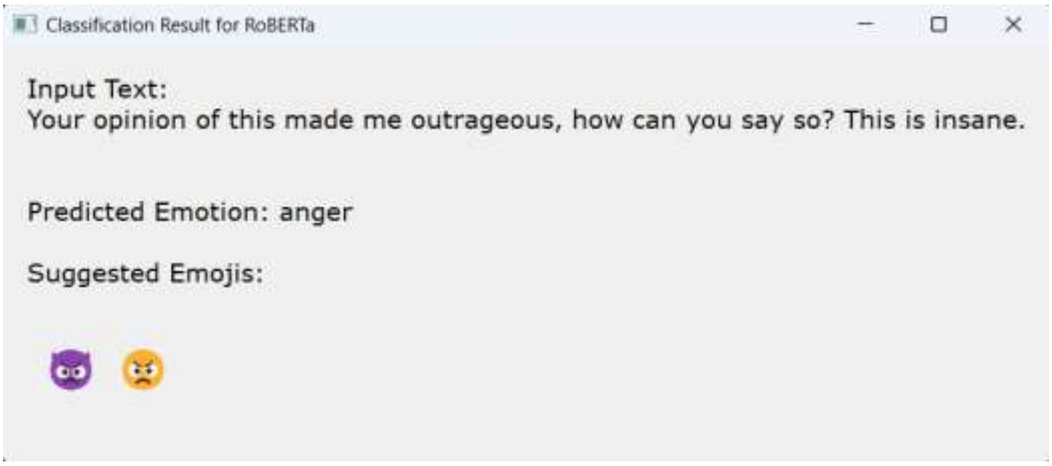
RoBERTa

RoBERTa, or Robustly optimized BERT approach enhances pretraining techniques, refining masked language modeling and training on a more extensive dataset to achieve superior performance in understanding context and semantics within textual data. In the realm of emotion classification, RoBERTa exhibits exceptional capabilities in capturing intricate contextual relationships and nuances present in sentences.

Accuracy: 73%

Your opinion of this made me outrageous, how can you say so? This is insane.

Figure 11 Result display for Emoji Recommendation



Input Text:
Your opinion of this made me outrageous, how can you say so? This is insane.

Predicted Emotion: anger

Suggested Emojis:

👿 😠

5 Discussion

a. Conclusions

The development of a system for recommending emojis requires a high-quality model in the background that will be able to track emotions in a sentence. We started the study with various models, ranging from the classic machine learning algorithms such as SVM, Logistic Regression and Naïve Bayes to more advanced, deep learning models, precisely bidirectional long short-term memory networks (Bi-LSTM), and the currently most popular language models such as BERT, RoBERTa, LLaMa and Phi.

Machine learning methods proved to be the least accurate and today we would no longer take them as a model on which to do software or research. Well-known deep learning models proved to be significantly more accurate, but also larger. Their size requires powerful GPUs. This is why we often have to train and run them on powerful servers. Sometimes they are not available or have a price.

In this study we also saw the importance of small language models, such as Phi, which with 2.7B parameters get an accuracy close to Llama with 7B parameters.

b. Theoretical implications

Researchers have been dealing with the classification of emotions in text for many years. There are different ways of approaching the topic. Also, knowledge from this area is transferable in the areas of human-computer interaction (HCI), customer experience, healthcare, education, security, market research, entertainment...

The theoretical importance of this study is in the presentation of the performance of various machine and deep learning models. In today's era, when many different language models are available, it is necessary to see from practical examples how each one works. Having knowledge about the performance of individual models, we speed up the process of finding the right model for a specific application. It is also necessary to see how we can improve state-of-the-art natural language processing models with fine-tuning. Also, we need to see the possibility of their integration with software.

In this work, we have seen how even with smaller language models we can get results close to the results of large language models.

c. Practical implications

State-of-the-art natural language processing models enable the development of software that will quickly and efficiently recognize the context of various texts. Some of such texts can be the most ordinary messages, but also various public news that have a great impact on today's world. Most of the text published on social networks also contains emojis. Appropriate emojis can help make the text meaningful and create an accurate vision of its content. Apart from the fact that the models presented in the paper allow us to classify emotions on the basis of which emojis are proposed, they also have another significant importance. The models that proved to be the best, Phi and Llama, are open source. This enables us to easily integrate them into the systems, but also develop new innovative applications using them.

d. Limitations and future research

While doing this study, we encountered various limitations. We trained the models on a dataset of 9806 sentences. If we take into account how many sentences are written on social networks in just one day, we can conclude that the dataset does not cover a large part of the sentences whose emotions will need to be recognized. Collecting user messages could be a privacy concern. Alternatively, we could ask the users to write us some sentences that they wrote that day and mark the emotion behind those sentences. In this way, we would collect important data from voluntary users for further training.

Also, an improvement of this system could be the addition of a user feedback option. In this way, we could improve the performance of all models.

Also, resources of computing power and execution time are often a key factor in choosing a model. For further work, we could improve the training parameters of existing models. With the Llama model, we could try using a 13B or 70B parameter model. If the size of the model is important to us, we can improve the fine-tuning small language model Phi check the performance of other small language models and try to work with them.

6 Conclusion

Our study was focused on examining the performance of various models that will help us in building a system for emoji classification. Emojis are used to express or highlight an emotion or idea. They are widely popular and utilized by many digital users worldwide on different social platforms.

We explored various models, ranging from traditional machine learning algorithms such as SVM, Logistic Regression, and Naïve Bayes to cutting-edge deep learning models like Bi-LSTM, BERT, RoBERTa, LLaMa, and Phi. We got acquainted with their specifications and characteristics. Then, we fine-tuned them to classify 8 emotions. It was no surprise when Meta's state-of-the-art Llama and Microsoft's ever-popular Phi proved to be the best models. Besides being the newest, they have the most parameters. We have shown how these models can make it interesting to discern the context of sentences. Models like Llama and Phi enable the development of software for fast and accurate recognition of emotions in messages, public news and posts. However, we can also use them for various other cases where it is important to recognize the emotion in the text. Also, in this study, we got an insight into the simple integration of the model with a small system used by the end user.

There are still opportunities for progress. We need a larger dataset on which to train our models. If we are going to stick to these models, it is necessary to devote more time to choosing the appropriate parameters. Train models with different number of epochs, learning rate... Also, we can try stronger versions of the used models and see how they perform and whether it is worth using them if we take into account Time (GPU hours).

Our study not only shows emotion classification techniques, but also emphasizes the importance of quality language models and working with them. As technology advances, our work serves as a foundation for future tasks in fine-tuning and using these models to improve written communication, as well as its interpretation.

7 Bibliography

Akash. (2022, 4 6). *Logistic Regression and Maximum Likelihood: Explained Simply*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2022/03/logistic-regression-and-maximum-likelihood-explained-simply-part-i/

Barbieri, F., Camacho-Collados, J., F., R., Espinosa-Anke, L., Ballesteros, M., Basile, V., . . . Saggion, H. (2018). SemEval 2018 Task 2: Multilingual Emoji Prediction. *Proceedings of the 12th International Workshop on Semantic Evaluation*, (pp. 24–33). New Orleans, Louisiana.

Bharti, S. K., S., V., Gupta, R. K., Shukla, P. K., Bouye, M., Hingaa, S. K., & Mahmoud, A. (2022). Text-Based Emotion Recognition Using Deep Learning Approach. *Comput Intell Neurosci*.

Chudasama, V., P., K., A., G., Shah, N., Wasnik, P., & Onoe, N. (2022). M2FNet: Multi-Modal Fusion Network for Emotion Recognition in Conversation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, (pp. pp. 4652-4661).

Efimon, V. (2023, September 23). *Large Language Models: RoBERTa — A Robustly Optimized BERT Approach*. Retrieved from Towards Data Science: https://towardsdatascience.com/roberta-1ef07226c8d8

Jacob Devlin, M.-W. C. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*

Javaheripi, M., & Bubeck , S. (2024, January 27). *Phi-2: The surprising power of small language models.* Retrieved from Microsoft Research Blog: https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/

Ragheb, W., Azé, J., Bringay, S., & Servajean, M. (2019). Attention-based modeling for emotion detection and classification in textual conversations. *arXiv preprint*.

Ray, S. (2024, January 24). *Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Saini, A. (2024, January 23). *Guide on Support Vector Machine (SVM) Algorithm*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

Seal, D., Roy, U. K., & Basak, R. (2019). Sentence-level emotion detection from text based on semantic rules. *Information and Communication Technology for Sustainable Development*.

Sriram. (2022, October 2). *Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2024*. Retrieved from upGrade: https://www.upgrad.com/blog/multinomial-naive-bayes-explained/

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., . . . Lample, G. (2024, January 27). *LLaMA: Open and Efficient Foundation Language Models.* Retrieved from arXiv: https://arxiv.org/abs/2302.13971

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., . . . Chen, M. (2024, January 27). *Llama 2: Open Foundation and Fine-Tuned Chat Models.* Retrieved from MetaAI: https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/

Xie, R., Liu, Z., Yan, R., & Sun, M. (2016). Neural Emoji Recommendation in Dialogue Systems. *CoRR.*

Yang, S. (2021, April 18). *Multiclass logistic regression from scratch*. Retrieved from Towards Data Science: https://towardsdatascience.com/multiclass-logistic-regression-from-scratch-9cc0007da372

Zhang, Y., Wang, M., Wu, y., Tiwari, P., Li, Q., Wang, B., & Qin, J. (2023). DialogueLLM: Context and Emotion Knowledge-Tuned Large Language Models for Emotion Recognition in Conversations. *arXiv:2310.11374.*

Zhao, G., Liu, Z., Chao, Y., & Qian, X. (2021). CAPER: Context-Aware Personalized Emoji Recommendation,. *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, pp. 3160-3172.

Zhao, J., Dong, L., Wu, J., & Xu, K. (2012). MoodLens: An emoticon-based sentiment analysis system for chinese tweets. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Zheng, X., Zhao, G., Zhu, L., & Qian, X. (2022). PERD: Personalized Emoji Recommendation with Dynamic User Preference. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 1922-1926).