# Sentiment Analysis with Emojis Using NLP

## Introduction

In recent years, the rise of digital communication platforms, particularly social media, has led to an increased use of emojis as a method of expressing emotions, opinions, and context in textual messages. Emojis enhance the understanding of text and have become essential in sentiment analysis, making them crucial elements of modern Natural Language Processing (NLP) applications. Emojis, however, add complexity to NLP models, as their semantics differ from that of plain text, often conveying nuanced emotional meanings.

This project focuses on leveraging emoji embeddings and employing various machine learning models to improve sentiment analysis. The key objective is to explore how emojis can be represented numerically through embeddings and integrate them effectively with text for sentiment analysis using machine learning and deep learning models. The project further extends to compare the effectiveness of various classification approaches, including LSTM, DNN, and transformer-based models like RoBERTa.

## Phase 1: Word2Vec Emoji Embedding

The first stage of the project began by creating embeddings for emojis using the Word2Vec model. Word2Vec is a popular method for generating dense vector representations of words, and in this case, the model was trained specifically to generate embeddings for emojis. The Word2Vec model used the **skip-gram** architecture to train on a dataset consisting of emojis, generating vectors that captured the semantic relationships between these symbols.

### Key Steps and Outcomes:

- **Training Word2Vec on Emoji Dataset**: The model was trained using specific settings, such as vector dimensionality and window size, to produce dense vectors for each emoji.
- **Cosine Similarity for Semantic Relationships**: The resulting embeddings were analyzed using cosine similarity, which enabled the identification of emojis with similar semantic meanings. For instance, emojis representing joy or sadness would have closely related embeddings, facilitating improved sentiment interpretation.
- **Application of Emoji Embeddings**: These emoji embeddings were then stored and used for further sentiment analysis, where their vectors were incorporated into text processing pipelines.

This phase laid the groundwork for incorporating emojis into NLP tasks by converting them into numerical vectors that a model could use, thus bridging the gap between textual and non-textual data in sentiment analysis.

## Phase 2: Word2Vec with Emojis and Text

With the foundation of emoji embeddings established, the next phase of the project involved integrating emojis with traditional text to create a combined dataset for sentiment analysis. The focus was on training the **Word2Vec model** to learn from both words and emojis simultaneously, thus improving the context in which these symbols were understood.

### Approach and Methodology:

- **Dataset Preparation**: A dataset of tweets, which naturally combine text and emojis, was preprocessed by tokenizing both words and emojis. This step allowed the Word2Vec model to treat emojis as part of the text data.
- **Skip-gram Architecture for Training**: The skip-gram model was employed to train on this dataset, capturing the contextual relationships between emojis and words. This approach allowed for the generation of vector embeddings that encoded both word and emoji semantics.
- **List of Similar Words/Emojis**: The trained Word2Vec model enabled the creation of a list of words or emojis that were most similar to each emoji in the dataset, providing insights into how emojis are commonly used in tweets and their corresponding contextual meanings.

This integration of emojis with text using Word2Vec significantly improved the ability of models to understand the emotional content of digital conversations, providing a more nuanced sentiment analysis.

# Phase 3: Classification Using LSTM and DNN

## Initial Text Classification (Without Emojis)

To establish a baseline for sentiment analysis, traditional text classification models were applied to the text-only portion of the dataset. These models were designed to predict the sentiment (positive, negative, neutral, etc.) of text inputs.

- **LSTM Model**: A Long Short-Term Memory (LSTM) model, known for its ability to retain context in sequential data, was employed for the task. The architecture consisted of embedding layers, bidirectional LSTM layers, and dense layers with softmax activation.
- **DNN Model**: A Deep Neural Network (DNN) model was also implemented for binary classification tasks. This model utilized embedding layers followed by dense layers with sigmoid activation for classification.

**Results:**

- The LSTM model, trained on pure text, achieved decent results but exhibited overfitting, with a validation accuracy of approximately **0.76**.
- The DNN model performed similarly but struggled to generalize well on unseen data, with an accuracy of **0.61**.

## Classification with Emoji Names

The next stage involved preprocessing the dataset by replacing each emoji with its corresponding Unicode descriptive name (e.g., 😂 → "face with tears of joy"). This transformation helped convert the emojis into textual entities that could be interpreted by traditional models, making it easier to integrate emoji semantics into classification tasks.

- **LSTM with Emoji Names**: Retraining the LSTM model with this enhanced dataset led to a significant improvement in accuracy, as the descriptive names allowed for better integration of emojis into sentiment analysis.
- **DNN with Emoji Names**: The DNN model showed similar improvements but still lagged behind the LSTM model.

## Results:

- LSTM model performance improved to an accuracy of **0.76** after replacing emojis with their descriptive names.
- DNN model accuracy also improved but remained lower than the LSTM model, at **0.61**.

# Phase 4: RoBERTa Model

To further enhance sentiment analysis, the project employed **RoBERTa**, a transformer-based model known for its state-of-the-art performance on various NLP tasks. RoBERTa (Robustly Optimized BERT Pretraining Approach) builds on BERT but removes the Next Sentence Prediction task, making it more effective for single-text classification tasks like sentiment analysis.

### Key Steps:

- **Dataset Encoding**: The dataset (now containing both text and descriptive emoji names) was tokenized using the RoBERTa tokenizer to convert the text into input format compatible with RoBERTa.
- **Model Architecture**: The RoBERTa model was fine-tuned with several dense layers added for sentiment classification. Dropout regularization was incorporated to combat overfitting, particularly during training on social media data, which can be noisy and diverse.
- **Training and Evaluation**: The model underwent several epochs of training, with validation accuracies and losses tracked to evaluate its performance.

**Results:**

- **Accuracy**: RoBERTa demonstrated a test accuracy of **41.50%**, outperforming previous models in terms of handling more complex sentiment scenarios.
- **Loss**: The model's loss values indicated room for further optimization, although its pre-trained nature gave it an edge in understanding context and sentiment.

# Phase 5: Comparative Analysis and Discussion

The project involved comparing the performance of the various models developed throughout the different phases. The results revealed that **RoBERTa** outperformed the other models, especially when dealing with emojis integrated with textual content. The LSTM model, when emojis were replaced by descriptive names, provided the next best performance, making it a useful approach for tasks requiring a balance between interpretability and accuracy.

### Model Comparisons:

- **RoBERTa**: Accuracy of **41.50%** with robust contextual understanding.
- **LSTM with Emoji Names**: Accuracy of **0.76**, proving effective with descriptive names.
- **DNN**: Mixed performance, with the best accuracy reaching **0.61**, indicating its limitations in handling complex emotional content.

### Overfitting and Model Tuning:

Both LSTM and DNN models displayed overfitting during training, highlighting the need for additional regularization and fine-tuning. RoBERTa, however, showed promise in overcoming these issues, thanks to its advanced transformer architecture.

# Conclusion

This project successfully explored the potential of integrating emoji embeddings with text for sentiment analysis. The combination of Word2Vec emoji embeddings, descriptive emoji names, and advanced models like RoBERTa showcased various approaches for handling emojis in sentiment classification tasks. The results emphasized the importance of contextualizing emojis in NLP applications, with RoBERTa emerging as the most effective model for real-world sentiment analysis tasks.

Future work could focus on optimizing these models further, potentially applying techniques such as transfer learning and experimenting with larger, more diverse datasets to enhance performance across different social media platforms and languages.