REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

EMOJI PREDICTION AND MODEL ANALYSIS WITH NATURAL LANGUAGE PROCESSING

19011018 — Bedrettin Şamil Öztürk

SENIOR PROJECT

Advisor
Prof. Dr. Mehmet Fatih AMASYALI

Nisan, 2024

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
GPT	Generative Pre-trained Transformer
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
RNN	Recurrent Neural Network
SemEval	Semantic Evaluation
SVM	Support Vector Machine
TF-IDF	Term Frequency — Inverse Document Frequency
UROP	Undergraduate Research Opportunities Program

LIST OF FIGURES

1
Introduction

Non-verbal signals, which include gestures, facial expressions, and tone of voice, are essential in human communication as they can effectively convey emotions and intentions. According to studies, these nonverbal cues account for almost 90% of the emotional content that people exchange with one another. Emojis are crucial stand-ins for these non-verbal indicators in textual communication, since they allow the expression of emotions and attitudes that could otherwise be missed. The project recognizes the critical role that emojis play in improving text-based communication's emotional depth and clarity, particularly in a language as complex as Turkish.

Natural Language Processing (NLP) sentiment analysis is traditionally approached by classifying text into three fundamental sentiment categories: positive, negative, and neutral. Nevertheless, due to the complexity and diversity of human emotions, binary or ternary classification schemes are unable to adequately represent the whole range of human feeling. Acknowledging this drawback, our work broadens the analysis to include a wider range of emotions by associating them with particular emojis or sets of emojis. Every emoji included in our study is chosen based on its capacity to convey a particular emotional subtlety, offering a more accurate and nuanced depiction of the text's emotional content.

It is crucial that emojis accurately represent the desired emotions in text messages. Emoji predictions that are off could misinterpret the intended meaning and cause misunderstandings or miscommunications. Consequently, this project carefully assesses the success rates of these predictions in addition to concentrating on the predictive component of emoji use. This work attempts to maximize the accuracy of emoji predictions by using machine learning models and improving their performance with data augmentation techniques. This thorough examination of the effectiveness of different analysis techniques guarantees that the resulting emojis have significant meaning, improving the general standard of text-based communication.

1.1 Necessity and Objectives of the Project

In an increasingly digital world, where text-based communication predominates, the absence of physical gestures and intonations presents a unique challenge: conveying nuanced emotional expressions across diverse linguistic landscapes. Emojis, small digital icons that represent emotions, ideas, or objects, have become integral in bridging this gap, offering a visual enhancement to the starkness of text messages. Their usage transcends mere decoration; emojis encapsulate complex emotional and contextual messages in a single character, thus enriching digital dialogue.

The primary necessity for this project arises from the observation that current text-processing systems inadequately capture the breadth of human emotions. Traditional sentiment analysis often limits itself to basic categorizations—positive, negative, and neutral—which do not reflect the intricate spectrum of human feelings. This project, therefore, seeks to pioneer a more refined approach by using Natural Language Processing (NLP) and machine learning techniques to predict emojis that more accurately reflect the emotional undertones of text, particularly focusing on Turkish—a language rich in emotional expression yet underrepresented in computational linguistics research.

1.2 Chapters Overview

- Preliminary Review:

  This chapter explores the literature on emoji prediction and Natural Language Processing (NLP), with a focus on the latest developments in emoji-based deep emotion analysis for Turkish texts. It establishes the framework for the creative approaches used on this project.

- Feasibility:

  This part evaluates the project's labor and time commitments, software and hardware resources, legal implications, and economic factors to make sure it is feasible and viable in the long run.

- System Analysis:

  This section outlines the project's precise needs and specifications and provides a thorough schedule for the system's design and execution. In order to meet both technological and user-centric needs.

- System Design: The project's structural design is described in this chapter and is essential to the effective implementation of the emoji prediction models. It includes:

  - Software Design: Explains the integration of NLP models into the system design, with an emphasis on scalability and modularity.
  - Database Design: Covers the management techniques and data set detail and operations required for effective model training and operation.
  - Input-Output Design: Makes sure that inputs and system outputs are understandable and easy to utilize by concentrating on the user interface and interactions.

- Implementation:

  This last chapter documents the transfer from theory to practice by outlining the emoji prediction models' actual code and configuration. It confirms that the implementations successfully fulfill the project specifications and closely follow the intended designs

  .

2
Literature Review

Emoji use in digital communication has been a major area of contemporary NLP research attention. Emojis are essential for increasing the expressiveness of language because they provide visual clues that communicate feelings, emotions, or actions that may not be evident from text alone. Particularly since the emergence of social media platforms, where textual brevity is prevalent, this profession has grown quickly.

2.1 Current Approaches and Methodologies

Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) are two examples of machine learning models that have been extensively used in emoji prediction studies. But the field of research has moved toward more complex, context-aware systems with the advent of transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers). By capturing bidirectional contexts using the transformer architecture, these models greatly enhance the comprehension and production of emojis derived from textual material.

Muhammad Osama Nusrat and associates from Fast Nuces, Islamabad, and Université Bretagne Sud, Vannes, provide a transformer-based method utilizing BERT for emoji prediction in their paper "Emoji Prediction in Tweets using BERT."[1] The study describes how to use BERT, a model that has been pre-trained on a sizable corpus of text data, to determine the best emoji for a particular piece of tweet content. Using a huge dataset of tweets interspersed with emojis, they fine-tune the BERT model in an effort to accurately represent the contextual links between text and related emojis.

The study outperformed numerous cutting-edge models with an accuracy of over 75%. The ability of transformer models to handle the complex and sometimes confusing nature of emojis in digital communication is demonstrated by this high degree of performance. Their work has important implications for social media marketing and sentiment analysis, for example, where it's critical to comprehend the emotional and

semantic undertones of text.

In order to predict emojis from tweets in several languages, the Duluth UROP team at SemEval-2018 Task 2[2] used ensemble learning techniques incorporating classifiers such as Naive Bayes, Logistic Regression, and Random Forests. Their strategy, which made use of unigram and bigram characteristics, concentrated on using oversampling approaches to overcome data skewness. Although they started off performing in the center, post-evaluation modifications in text preprocessing greatly enhanced their outcomes, showing that minor preprocessing modifications might dramatically increase predictive accuracy. The present work highlights the significance of deliberate feature engineering and equitable data treatment in enhancing emoji prediction models.

In contrast to conventional models, the EPUTION project[3], which was presented at SemEval-2018, proposed a novel user adaptation strategy that improved emoji prediction accuracy by over 6%. The model could more accurately reflect personal usage patterns by personalizing emoji predictions by incorporating each user's tweeting history into the training process. With the use of user-specific data and FastText, this method quickly classified texts and addressed the issue of users' differing emoji preferences. Their findings support the incorporation of user behavior in NLP applications to improve the predictive model's performance, particularly in diverse and dynamic datasets such as social media.

In order to predict the frequency of the most popular emojis in tweets for both English and Spanish, this research analyzes the performance of several models, including logistic regressor, LSTM(Long Short-Term Memory), Random Forest Regressor, and SVM(Support Vector Machine). The other study[4] emphasizes the difficulties in predicting emojis, pointing out that certain emojis—like a Christmas tree—are more predictable than others that are used more frequently. The study found that there was a notable decline in performance during the assessment phase, which could have been caused by overfitting during the model creation phase. Cross-validation could potentially help to mitigate this problem.

In order to predict emojis in English tweets, this paper[5] presents a deep learning technique that uses LSTM networks improved with a context-aware self-attention mechanism. This model, which came in second place in the competition, uses word embeddings that have been trained on a sizable dataset of tweets instead of manually created features. The attention mechanism highlights the words in a tweet that have the greatest influence on the emoji, which not only increases the accuracy of emoji prediction but also enhances the interpretability of the model's decisions.

Larisa Alexa and colleagues' article "The Dabblers at SemEval-2018 Task 2: Multilingual Emoji Prediction"[6] investigates the application of Recursive Neural Networks and Naive Bayes to the prediction of emojis in tweets. The study emphasizes how difficult it is to understand emojis because they differ greatly between languages, countries, and user settings. This study highlights how machine learning might improve our comprehension of the subtleties of digital communication, particularly when viewed through the prism of emoji usage on various social media platforms.

The first shared job on emoji prediction, "SemEval 2018 Task 2: Multilingual Emoji Prediction,"[7] is described by Francesco Barbieri and his colleagues. The task involves predicting the most likely emoji from text-only tweets in both English and Spanish. The competition drew a sizable number of participants, suggesting a strong interest in the sophisticated comprehension of emojis as sophisticated digital text communication tools. In order to tackle the prediction tasks, the study employed a range of neural network models, underscoring both the potential and obstacles of incorporating emoji usage into more extensive natural language processing systems.

The Tübingen-Oslo team's strategy[8] for the SemEval-2018 Task 2, which concentrated on multilingual emoji prediction in texts written in both Spanish and English, is presented in this work. Support vector machines (SVMs) and recurrent neural networks (RNNs) were used, and their study demonstrated a comparison of these models. The researchers discovered that SVMs greatly outperformed RNNs when built with bag-of-word and character n-gram features and optimized using grid search for the best macro F1-score. This was demonstrated by the fact that their SVM model performed best on the English and Spanish subtasks. The superiority of the linear SVM approach over other machine learning and deep learning techniques suggests that it is robust enough to handle the sparse, high-dimensional data that is typical of text classification problems like emoji prediction.

Another study highlights[9] the understudied field of gender classification from brief social media messages in the Turkish language by examining gender identification from Turkish tweets using pre-trained language models. In addition to contemporary pre-trained models like BERT, DistilBert, and Electra, the study skillfully combines cutting-edge deep learning techniques (LSTM, CNN) with conventional machine learning approaches (TF-IDF + SVM). It is noteworthy that, within the limitations of complicated and frequently informal social media language, it achieves its maximum accuracy with a BERT model built with a specified word size, highlighting the possibility of fine-tuned language models to improve performance in author profiling tasks.

In order to tackle the problem of emoji prediction from tweets, the UMDSub at SemEval-2018 Task 2[10] used a multi-channel Convolutional Neural Network (CNN) that uses subword embeddings as its textual representation technique. Their solution ranked 21st out of 48 participants, improving by about 2% above standard character or word-based embeddings. This method demonstrates how subword granularity can effectively capture subtle semantic implications in tweets that are otherwise difficult to understand when using coarser embeddings.

To predict emojis in tweets written in both English and Spanish, YNU-HPCC[11] used a bi-directional gated recurrent unit (Bi-GRU) supplemented with an attention mechanism for SemEval-2018 Task 2. Using boosting and ensemble techniques, their novel approach merged numerous models, resulting in a performance improvement of up to 3% in the macro F1 score when compared to baselines of single models. The accuracy of emoji predictions based on textual inputs can be improved by using attention mechanisms and ensemble approaches, as demonstrated by this work.

2.2 Identification of Gaps and Contributions of the Project

Emoji prediction has mostly concentrated on popular languages like English, undervaluing text samples written in languages other than English, particularly those with complex linguistic origins like Turkish. By especially improving emoji prediction for Turkish texts—which contain particular language nuances and cultural contexts not often covered in existing research—this effort seeks to close this gap.

Furthermore, although previous research has advanced the field of emoji prediction significantly, it frequently uses single-model techniques that might not fully capture the range of contextual subtleties seen in language. A comparative examination of several sophisticated machine learning models, such as KNN, Naive Bayes, LSTM, and SVM, is presented in this research. This aids in both finding the most effective model and comprehending how various data augmentation methods, such as SMOTE, Random Under Sampling, and Random Over Sampling, affect the models' capacity for generalization.

This project also stands out for incorporating a thorough analysis of several vectorization techniques and their effects on model performance, investigating the ways in which text conversions into numerical data might influence the precision of emoji predictions. The study aims to contribute to the broader NLP literature by offering significant insights on enhancing NLP systems for emoji prediction through a thorough comparison across several dimensions. The approaches employed are robust and may be applied to a variety of language settings. This deliberate

approach guarantees that the system not only improves Turkish text interpretability in digital communications but also establishes a standard for future multilingual natural language processing research.

3
Feasibility

This project report's feasibility section offers a thorough evaluation of the project's viability and attainability. In order to successfully complete the project within the limitations of existing technical capabilities, available resources, and set timeframes, a thorough study is required. The main objectives of the assessment are to determine whether the chosen resources are compatible with the project's requirements, to carefully plan the project's timeline and make sure that each phase is feasible, and to critically analyze the economic aspects in order to predict the project's financial sustainability and budgetary needs.

The project's potential impact and the value it aims to provide to the field of NLP—particularly in the area of emoji prediction—are highlighted in this section. It provides a detailed analysis of how the project can improve current methods for text analysis and the identification of emotional expressions, with a focus on modifying language models to reliably understand and produce emojis from Turkish text samples. The feasibility analysis, by taking into account potential risks and problems and offering solutions to mitigate them, establishes a solid basis for the project's successful execution and implementation.

The feasibility study also examines the project's strategic significance in meeting the complex needs of natural language processing (NLP) applications, particularly in contexts with linguistic and cultural diversity. It draws attention to how the project has advanced the creation of sophisticated language models that can comprehend and work with the intricate linguistic structures found in Turkish literature. The feasibility section sets the foundation for the in-depth examination of technical, financial, and legal matters in later sections of the report by carefully assessing these factors in an effort to provide a thorough and informed view on the project's viability.

3.1 Technical Feasibility

The project's technical feasibility depends on the technologies and infrastructure chosen to guarantee its effective implementation and completion. This section is divided into hardware and software components, highlighting the optimal choices to efficiently meet the project's requirements.

3.1.1 Software Feasibility

The choice of Python as the main programming language, which is well-known for having a strong ecosystem and being especially advantageous for projects involving data science, machine learning, and natural language processing, is a clear indication of the software feasibility of our project. PyTorch, TensorFlow, and Scikit-Learn are just a few of the many packages that make difficult activities like model development, training, and assessment easier with Python's easy-to-read syntax. Large datasets may be handled efficiently with TensorFlow, which also enables deep learning models that are essential for semantic analysis, like the BERT architecture. In the meantime, PyTorch provides the adaptability required for iterative changes and dynamic experimentation during the model-development stage.

Transformers library from Hugging Face is also made use of it, which gives us access to cutting-edge pre-trained models. These models are optimized to improve our emoji prediction models' predictive performance by strengthening their capacity to identify contextual nuances in Turkish texts. Utilizing Scikit-Learn enhances these resources by streamlining the model assessment procedure and streamlining the measurement of performance indicators like F1-scores, precision, and recall.

Our main platform for development and execution is Google Colab Pro, which is a cloud-based service that offers strong computational resources like GPUs and TPUs. This configuration not only removes the requirement for a sizable local computational infrastructure, but it also drastically cuts down on development time, freeing up more time for algorithm optimization and model testing. The technological viability of our research is ensured by this deliberate choice of software tools, which allows us to effectively address the problems of natural language processing and emoji prediction with improved performance and accuracy.

3.1.2 Hardware Feasibility

To handle the large calculations needed by the deep learning models employed in this study, the project depends on Google Colab Pro for access to sophisticated computing resources. Rapid model training and iteration are made possible by the availability of

A100 GPU on Colab Pro, which greatly reduces development time and opens the door to more complicated experiments.

It is advised to use a current multi-core CPU (Intel i7 or AMD Ryzen 7) with at least 16GB of RAM and a fast SSD for local development and small-scale testing. This hardware setup guarantees bottleneck-free, efficient initial model training, code development, and data preprocessing.

Future scalability and the possible requirement for more computational resources as the model's complexity and data volume increase are also taken into account in this research. Because of the technology and software's versatility, the project can grow without requiring major overhauls, ensuring its long-term technological sustainability.

The project builds a strong basis for tackling the complex problems of emoji prediction in Turkish texts by carefully choosing these hardware and software components, guaranteeing that all technological requirements are satisfied for the project's successful completion.

3.2 Labor and Time Planning

Our project's Labor and Time Planning section is essential for outlining the specific actions needed to accomplish the project's goals. This section lays out the precise activities that need to be accomplished, lists the technical abilities required at each level, and specifies the time allotted for each project phase. This strategy makes sure that every project component is carried out effectively and within the allotted eight weeks by creating a well-organized timeline."

Figure 3.1 shows the Labor and Time Planning for the project, outlines the specific procedures and deadlines needed to successfully complete each phase of the project.

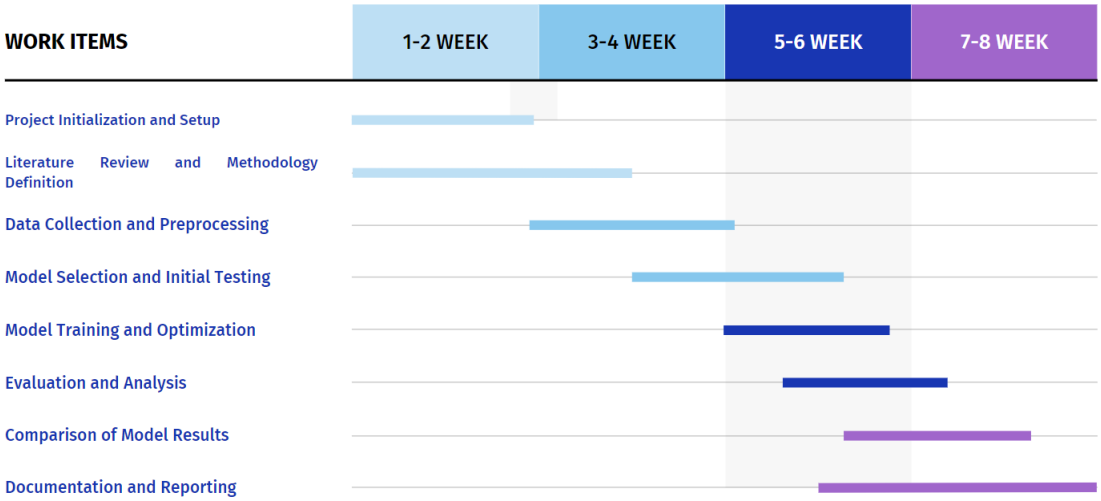


Figure 3.1 Gantt Chart

- **Project Initialization and Setup (Week 1)** During this foundational phase, the infrastructure required for the project is built up. In order to guarantee that the team has access to the necessary computational resources, this involves establishing Google Colab Pro and other development environments. Additionally, it entails obtaining a sizable dataset appropriate for emoji prediction model training, with a particular emphasis on obtaining representative and varied Turkish text data.

- **Literature Review and Methodology Definition (Weeks 1-2):** Previous studies were examined on emoji prediction with machine learning models and natural language processing approaches during these weeks. The project procedures, which include choosing the right NLP tools and machine learning algorithms to be used throughout the project, must also be defined and refined throughout this time.

- **Data Collection and Preprocessing (Weeks 2-3):** The project's main objective is to gather and prepare text data in Turkish. The data is cleaned, suitably annotated, and prepared using techniques like tokenization and vectorization. Model training efficacy can be increased by ensuring balanced classes in the training set through the use of techniques like SMOTE.

- **Model Selection and Initial Testing (Weeks 3-4):** LSTM, SVM, KNN, and Naive Bayes are just a few of the machine learning models that are investigated in this study. Baseline tests are performed to assess the initial performance of the configurations that have been put up. Based on early accuracy and performance measurements, this phase aids in finding the most promising models.

- **Model Training and Optimization (Weeks 4-6):** The prepared datasets are

used to train the chosen models, with an emphasis on optimizing performance by adjusting hyperparameters. The capacity of the models to generalize effectively on untested data is enhanced at this phase, which guarantees strong emoji prediction skills.

- **Evaluation and Analysis (Weeks 6-7):** Metrics including support, recall, F1 score, precision, and recall are used to thoroughly assess models. The effectiveness of each model is evaluated through performance analysis utilizing techniques like as confusion matrices and accuracy tables, which help to find the top methods for emoji prediction.
- **Comparison of Model Results (Week 7):** To evaluate the efficacy of all trained models, a comparison analysis is carried out. This entails comparing their advantages and disadvantages side by side and choosing the best model for possible real-world application based on predetermined criteria like accuracy and recall.
- **Documentation and Reporting (Weeks 7-8):** Every facet of the project, including the model setups, experimental outcomes, and comprehensive explanations of the employed methodology, is well documented. The final report and presentation are meticulously formatted and carefully considered in order to ensure that the project's findings are communicated successfully and that they provide a thorough reference for future work.

3.3 Legal Feasibility

Because this work makes use of publicly available, open-source datasets and frameworks, its legal viability is fully guaranteed. The main dataset used comes from platforms that specifically allow for research and educational use, guaranteeing copyright compliance. Furthermore, the risk of infringement on commercial technology is reduced because all computational tools and machine learning frameworks, such as TensorFlow and BERT, are covered under open-source licenses that provide unlimited use for educational purposes.

The project complies with general data protection requirements and respects intellectual property rights because it does not involve information that is personally identifiable or sensitive personal data. When analyzing textual data and utilizing emojis for analysis and prediction, this proactive consideration protects against potential legal ramifications. Because the project's design and methods respect user privacy and ethical norms, it is both legally sound and workable within academic and research institutions.

3.4 Economic Feasibility

The assessment of the project initiative's cost-effectiveness and strategic advantages is the primary method of determining its economic feasibility. The use of Google Colab Pro considerably reduces the principal expenses associated with the development and operation phases. By giving users access to improved computational resources—like more RAM and processing power—this platform offers an affordable alternative. These resources are essential for effectively managing complicated model training and big datasets. By minimizing the need for costly hardware purchases and cutting down on the amount of time needed for model training, Google Colab Pro adoption results in cost savings and better productivity.

Furthermore, the project does not require expensive software licenses or proprietary technologies because it makes use of open-source software and tools for all phases of design and execution. By using freely accessible libraries and frameworks, which serve the project's technical demands without adding to the financial load, the operational costs are further kept under control. In terms of finances, the research will help the company by improving the functionality of current natural language processing (NLP) apps. This could result in increased customer happiness and engagement in applications where precise emoji prediction is important. Therefore, the significant gains in processing efficiency and the tactical advantages obtained in emoji-driven communication projects justify the small investment in Google Colab Pro.

4
System Analysis

In order to properly handle the project objectives, the System Analysis chapter of the project attempts to precisely describe and develop the essential components and capabilities. The project's objectives are thoroughly described in this section, along with the information sources and particular needs that must be fulfilled in order to go on to the design and execution phases.

4.1 Project Objectives and Requirements Identification

This research aims to improve the accuracy of emoji predictions from text data by utilizing advanced Natural Language Processing (NLP) and machine learning techniques. Specifically, the study targets emotional emotions found in Turkish literature. An detailed analysis of the body of research and technology in the fields of natural language processing and emoji prediction was conducted in order to identify the requirements. The goal of this analysis was to identify current technological shortcomings, with an emphasis on enhancing the precision and dependability of emoji predictions. To improve model training results and enrich the dataset, this project will make use of a variety of data enrichment strategies. Furthermore, a thorough analysis will be conducted on the effects of various factors, including train-test distribution, dataset size, and vectorization techniques, and algorithms for machine learning, including KNN, Naive Bayes, LSTM, and SVM, on prediction outcomes. These in-depth investigations are intended to improve the system's overall performance and optimize the emoji prediction process.

4.2 Research and Information Gathering Methods

A variety of techniques were used to conduct the research and collect data, such as a thorough examination of the available datasets, earlier investigations into the field of emoji prediction, and the usage of machine learning models that are currently in use in comparable scenarios. During this phase, prospective end users and NLP specialists

were consulted in order to provide qualitative information that influenced the creation of the project's technical specifications.

4.3 System Modules

The project is divided into multiple major modules, each of which is intended to handle a certain duty in the pipeline for predicting emojis. Together, these modules use the most recent developments in machine learning and natural language processing to process data, train models, and forecast emojis based on Turkish text data.

- **Data Collection and Preprocessing Module:** The dataset's collecting, cleaning, and preparation fall under the purview of this module. To achieve unbiased model training, the data sequence is randomly generated. Textual labels are then translated into corresponding emojis using a predetermined lexicon. In order to standardize the text data, this module also handles the deletion of stopwords, punctuation, and lowercase letters. These preprocessing stages are essential for lowering the complexity of the model and raising the standard of learning.

  An example of lemmatization, considered as one of the most complex steps of preprocess operations in Turkish language, is given in **Figure 4.1**.

```
<(bitişik_Adj)(-)(bitişik:adjectiveRoot_ST)>
<(bitişik_Noun)(-)(bitişik:noun_S + a3sg_S + pnon_S + nom_ST)>
<(şaşırmak_Verb)(-)(şaşır:verbRoot_S + t:vCaus_S + verbRoot_S + ıcı:vAgt_S + adjAfterVerb_ST)>
<(şaşırmak_Verb)(-)(şaşır:verbRoot_S + t:vCaus_S + verbRoot_S + ıcı:vAgt_S + noun_S + a3sg_S + pnon_S + nom_ST)>
<(zaman_Noun_Time)(-)(zaman:noun_S + a3sg_S + pnon_S + ı:acc_ST)>
<(zaman_Noun_Time)(-)(zaman:noun_S + a3sg_S + ı:p3sg_S + nom_ST)>
<(gelmek_Verb)(-)(gel:verbRoot_S + miş:vNarr_S + ti:vPastAfterTense_S + vA3sg_ST)>
<(gelmek_Verb)(-)(gel:verbRoot_S + miş:vNarrPart_S + adjectiveRoot_ST + adjZeroDeriv_S + nVerb_S + ti:nPast_S + nA3sg_ST)>
```

Figure 4.1 Lemmatization Example

- **Text Vectorization and Encoding Module:** Text data is preprocessed and then formatted so that it may be used for training models. Tokenized tensors, which are used to train the models, are created by this module using tokenizer features from the Transformers library. By making sure that the text data is properly vectorized using the BERT tokenizer—which takes into account the Turkish language's semantic richness—it improves the model's comprehension and accuracy in predicting emojis.

- **Model Training and Evaluation Module:** The training of numerous machine learning models, such as KNN, Naive Bayes, LSTM, and SVM, as well as sophisticated neural network topologies made possible by the Hugging Face Transformers library, is the main goal of this module. Using datasets prepared and encoded in earlier modules, it oversees the training process. Accuracy,

precision, recall, and F1-score are the performance metrics used to evaluate these models. Validation data sets are used to further refine the models and determine which performs best.

- **Emoji Prediction Engine:** This module, which forms the project's core, combines the learned models to forecast emojis from processed text inputs. It serves as a conduit between the user's text input and the output (emoji predictions) from the models, guaranteeing that the former are precise and suitable for the given context. This module contains the functionality needed to process text input in real-time or in batches as needed. It does this by running the text through trained models and returning the predicted emoji. .

Together, these modules make up the core of the emoji prediction system, and each one is essential to the project's successful completion. They make sure the system is strong, scalable, and able to manage the intricacies involved in emoji prediction from Turkish texts and natural language processing. The workflow is well organized thanks to this modular design, which also makes system scalability and maintenance easier for future improvements.

4.4 Requirement Analysis Model

This project's requirement analysis uses a functional approach to explain how text data inputs are converted into predictions for emojis. This entails mapping every step of the process, from gathering data to preparing it for analysis to producing the final prediction. To ensure clean and normalized data, the Turkish text data is first collected and then subjected to a thorough preprocessing process that includes tokenization, stopword removal, and punctuation removal. After preprocessing, advanced models such as BERT which is provided bu Hugging Face Transformers library are used to encode the text. These models are more effective than standard methods at capturing contextual nuances. After the texts have been encoded, they are fed into machine learning models for training. Various algorithms, such as KNN, Naive Bayes, LSTM, and SVM, are then tested to see which performs and is most accurate.

The project takes a modular approach, meaning that each part (emoji prediction, model training, and data preprocessing) is separate but connected, enabling testing and improvements iteratively. This thorough analysis guarantees that the system architecture is resilient, scalable, and able to handle the intricacies involved in predicting emojis from Turkish text data, in addition to meeting the project's objectives. This rewrite ensures clarity and relevance to the functional needs of the

system, while maintaining a cogent narrative flow and incorporating all components described in your project updates.

The diagram visualizing the data flow steps is as in **Figure 4.2**

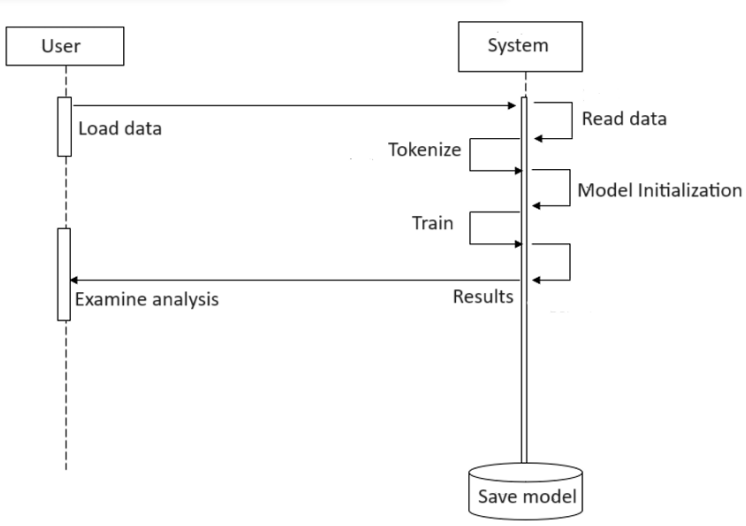


Figure 4.2 Data Flow Diagram

4.5 Performance Metrics

Several critical performance measures will be used in this research to evaluate how well the emoji prediction models work.

4.5.1 Confusion Matrix and Its Importance

When assessing machine learning models, a confusion matrix is an essential tool, especially when dealing with categorization issues like emoji prediction. It's a table that shows actual against anticipated classifications, enabling algorithm performance visualization. It is significant for the project because it offers in-depth information about the precision of forecasts and highlights the areas where the model succeeds and fails, which is essential for fine-tuning.

4.5.2 How the Confusion Matrix Works

The confusion matrix lays out the predictions in a matrix format with:

- **True Positives (TP):** Correctly predicted positive observations.
- **True Negatives (TN):** Correctly predicted negative observations.
- **False Positives (FP):** Incorrectly predicted as positive (Type I error).
- **False Negatives (FN):** Incorrectly predicted as negative (Type II error).

The confusion matrix, which examines actual and predicted values, is presented in **Figure 5.5**.

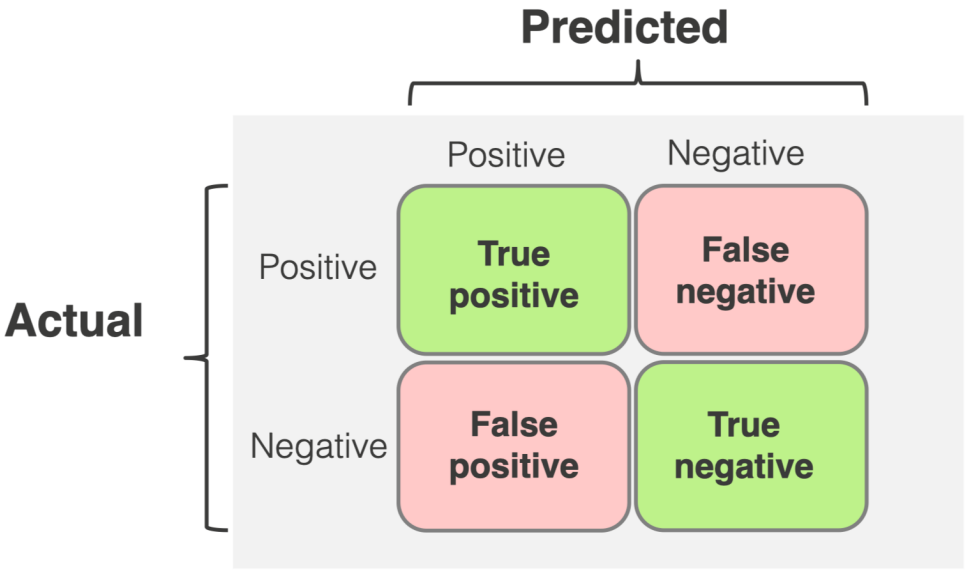


Figure 4.3 Confusion Matrix [12]

4.5.3 Accuracy

This metric computes the ratio of accurately predicted data to total observations, so assessing the overall efficacy of the model. It shows how frequently the model correctly associates emoticons with the appropriate emotional content of the text for our emoji prediction project. A high accuracy rate indicates that the model can be trusted to provide accurate predictions for all data classes. This measure provides a rapid means of evaluating the model's overall performance.

4.5.4 Precision

This metric, which calculates the percentage of genuine positive findings among every positive result predicted by the model, assesses how accurate the model's predictions are. For the sake of our project, precision will show us the percentage of emojis that the model correctly predicted out of all the emojis projected for a certain text sample. When the cost of a false positive—an emoji that is mistakenly predicted—is high, this metric becomes critical since it could misrepresent the intended sentiment.

4.5.5 Recall

Recall, which is often referred to as sensitivity, evaluates how well a model can locate each pertinent example within a dataset. For the sake of the project, it evaluates the percentage of real emojis in the data that the model successfully detected out of all the real emojis. In order to make sure that the model does not overlook any pertinent emotional expressions, this metric is critical when it comes to capturing all potential right predictions.

4.5.6 F-1 Score

The F-1 Score is a single score that provides a balance between the model's precision and recall, calculated as the harmonic mean of precision and recall. It is very helpful when you just require one metric to evaluate several models or configurations, particularly when recall and precision have equal weight. The F-1 score will be used in this research to assess the overall accuracy of the emoji prediction models across a range of parameter settings, providing information on how variations in the model's setup impact its functionality.

4.5.7 Support

The number of instances of each class in the real, authentic data collection is indicated by this measure. In our case, support will count the number of times each emoji appears in the dataset. This will aid in deciphering the data distribution and evaluating the efficacy of the model for each emoji depending on how frequently it occurs.

Comprehending these metrics via the confusion matrix aids the emoji prediction effort in:

- Figuring out which emojis people frequently mix up with other ones so that the training set or model architecture may be specifically improved.
- modifying the model to enhance either precision or recall, depending on the particular requirements of the project (e.g., whether it's more critical to capture every pertinent emojis or to ensure the emojis predicted are correct).
- Balancing the dataset if some classes (emojis) are over-represented or under-represented.

The Venn diagram illustrating the relations of the values of True Negatives (TN), True Positives (TP), False Negatives (FN), and False Positives (FP) is depicted in **Figure 4.4**

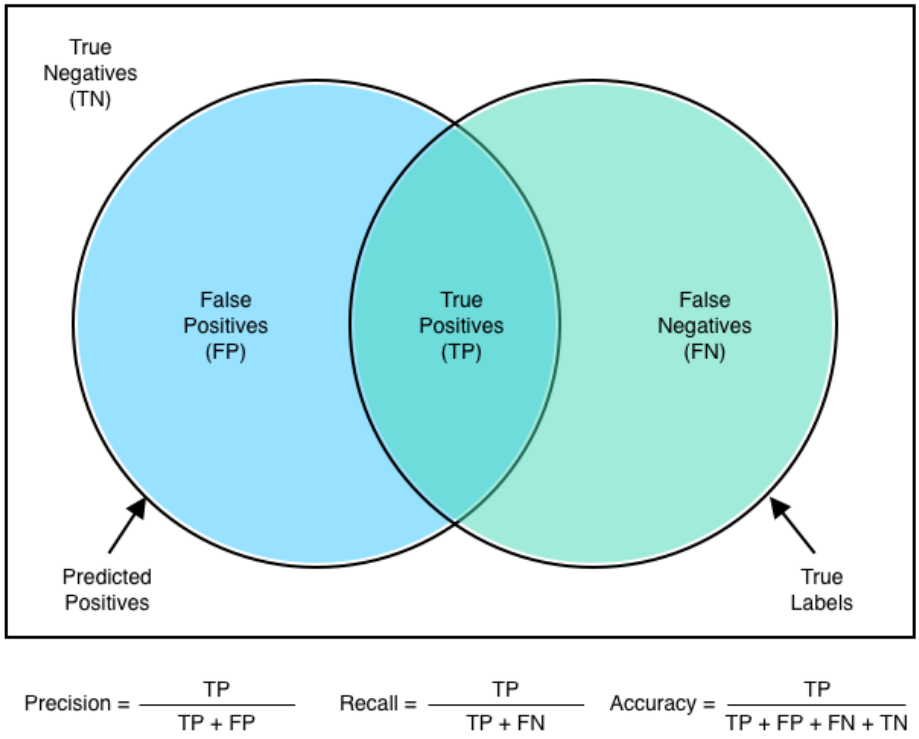


Figure 4.4 Performance Metrics Venn Diagram [13]

Numerous factors, including the size of the dataset, the vectorization method selected, and the train-test split, might affect each of these measures. The research attempts to determine the best model configuration by experimenting with these parameters in order to optimize the performance metrics. This will guarantee the model's resilience and dependability in correctly predicting emojis for Turkish text inputs.

The thorough system analysis lays the groundwork for the design and implementation stages, guaranteeing that all functional and technical requirements are met to match the project's high standards.

5
System Design

The project's system design is focused on building an effective and scalable architecture that can handle the challenges of predicting emojis from textual input. The approach improves the performance and accuracy of emoji predictions by utilizing cutting-edge machine learning and natural language processing methods.

5.1 Software Design

The software architecture is modular, comprising distinct components for data handling, model training, and user interaction. These include data preprocessing modules that cleanse and prepare text data, a model training module where machine learning algorithms are applied, and a post-processing module to analyze and utilize the model outputs effectively.

The emoji prediction system's complete process, from initial data collection to final testing, is depicted in the **Figure 5.1** below. This graphic helps in comprehending the architecture of the system and the sequential processes involved in the creation and improvement of the emoji prediction models.

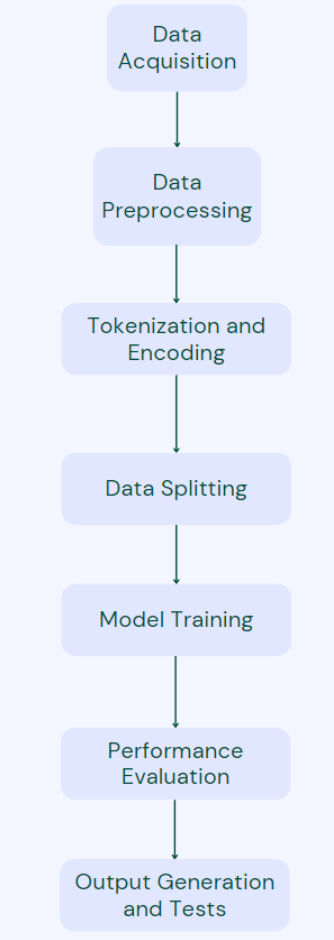


Figure 5.1 Software Architecture Flowchart

1. Data Flow and Process Design: The different phases of text data transformation inside the system are mapped out in a clear data flow. Preprocessing is first applied to raw text data in order to eliminate noise and standardize the format. The next step is the model training phase, when it is input into neural networks for the purpose of generating embeddings, such as BERT or GPT. After that, a classification layer is added to the data to predict the related emojis.

2. Machine Learning Model Integration: The main component of the design is the integration of pretrained models, such BERT, which have been optimized for the particular job of emoji prediction. The system has a training loop in which the accuracy and overfitting of the model are optimized by continually evaluating its performance.

3. Algorithm and Parameter Optimization: The choice and fine-tuning of algorithms and their parameters are essential to the architecture of the system. The model's performance on validation data is used to carefully adjust parameters including learning rate, batch size, and neural network architecture.

Enhancing the model's capacity to effectively generalize to new, untested data requires completing this phase.

4. Integration of Performance Metrics: The architecture incorporates methods for quantifying important performance metrics including support, recall, accuracy, and F1-score. These metrics assess how well the model performs across different emoji classes and direct future improvements to the model's design and training procedure.

This design plan makes ensuring that the program satisfies the performance criteria required for an emoji prediction system to be successful, in addition to meeting the functional requirements. The other elements of the system design, which will be covered in depth in the database and input-output design stages that follow, are firmly established by the software design.

5.2 Database Design

In order to train our algorithms to predict emojis properly based on text's emotional content, our research makes use of a dataset of Turkish tweets. This dataset, which comes from a trustworthy source, includes a wide variety of phrases, guaranteeing a thorough portrayal of linguistic subtleties. Examples of text and label pairs in the dataset used are listed in **Figure 5.2**

```
df1.sample(10)
```

	Text	Label
1449	Yanlışlıkla ön kamera açıldı irkildim	😰
1730	Artık be diyebilirim ki kelimelerimden de önce...	😔
1473	Ben şok olmuştum gidemez diye ağlamıştım ya in...	😱
2679	Son görülme saatin endişe verici rakamlara ula...	😰
775	Aziz Yıldırım defol git hayatımızdan küflü pey...	😠
3125	Bugün benim doğum günüm 21 de bitti	😊
3727	Antidepresan kullanmaya başladığımdan beri dah...	😠
2108	Bu kadar twite nasıl tt olmamış insan gerçekte...	😱
1169	Sinir akıyor kanımdan yemin ederim kafayı yiye...	😠
1571	Konyaspor adına üzüldüm hakem rakip gibi davra...	😔

Figure 5.2 Dataset Samples

The emojis in the dataset are meant to represent various emotional states or moods, and the structure of the dataset contains columns for the tweet text and their corresponding labels. Emojis function as labels on tweets, with each one representing the main idea expressed in the text. By categorizing the text input, the algorithm can learn to anticipate the emoji using supervised learning.

The length of the text contents in the dataset is as shown in in **Figure 5.3**

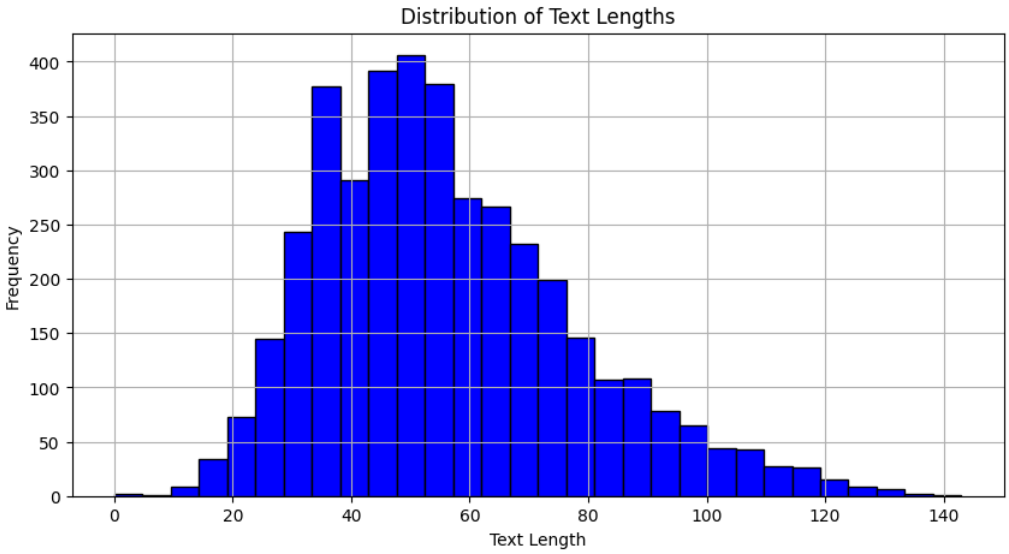


Figure 5.3 Text Lengths

Important preprocessing processes are used to improve the dataset's quality and efficacy in advance of machine learning. It includes:

1. **Case Normalization:** To guarantee that the model handles words with the same meaning consistently, regardless of their case, all text data is transformed to lowercase.

2. **Tokenization:** It is a fundamental preprocessing step in our project, where text is segmented into tokens or words. This process simplifies the parsing of input data for the model and is crucial for understanding the semantic structure of the text. In the project,two distinct tokenization strategies associated with the BERT and GPT models provided by the Hugging Face Transformers library and TF-IDF methods are utilized.

3. **Stopword and Punctuation Removal:** Common Turkish stopwords are removed from the text. Stopwords are typically frequent words that may not contribute significantly to the sentiment of the text, such as 've' (and), 'bu' (this), etc. This step helps in reducing the dataset size and improving processing time.

All punctuations are stripped from the text to reduce noise and avoid skewing the model's understanding of the sentiment.

4. **Encoding - Vectorization:** A couple of encoding and vectorization techniques are used to get the cleaned text ready for learning models. In this stage, the text is transformed into fixed-length vectors that accurately capture the semantic meanings of words and phrases found in the dataset. The project investigates a number of vectorization techniques, including more conventional methods like TF-IDF (Term Frequency-Inverse Document Frequency) and sophisticated neural network-based embeddings like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer).

   For example, the BERT tokenizer deconstructs text into subword units and captures the meaning of each word in relation to its surrounding text, enabling a profound comprehension of linguistic subtleties. In contrast, GPT tokenizes text in order to make it suitable for generative tasks while maintaining the contextual and stylistic integrity of the input material. Though more conventional, the usage of TF-IDF is evaluated for its ability to draw attention to frequently occurring phrases that are crucial to text categorization tasks.

   The project's goal is to examine and contrast the effects of these various vectorization techniques on the efficacy of the emoji prediction models. Every approach has its advantages when it comes to managing various facets of language representation, and each method's performance in processing Turkish texts for emoji prediction is carefully assessed. This thorough process guarantees that the selected model not only accurately represents the key characteristics of the input data but also maximizes prediction accuracy by utilizing the best vectorization method.

5.3 Input-Output Design

This part of the project documentation describes the procedures for ingesting data, analyzing it, and producing outputs that show how well our emoji prediction models work. The design places a strong emphasis on efficiency and clarity, making sure that data flow is optimized for performance and that the results are both aesthetically pleasing and relevant.

5.3.1 Input Design

For convenience of access and maintenance, the dataset —a crucial part of our project— is kept on Google Drive. The dataset is guaranteed to be centrally kept

and readily retrievable by using Google Drive. Google Colab, a cloud-based Python programming environment with powerful computing resources like GPUs and TPUs, is where the dataset is mounted. With this configurations, importing and preparing data into our interactive notebooks is streamlined, allowing for real-time data manipulation and analysis.

In order for the algorithms to produce the expected outputs, the label(emoji) distribution was shared equally as seen in **Figure 5.4**. 5 emojis, 800 of each emoji, were studied with a total of 4000 data.

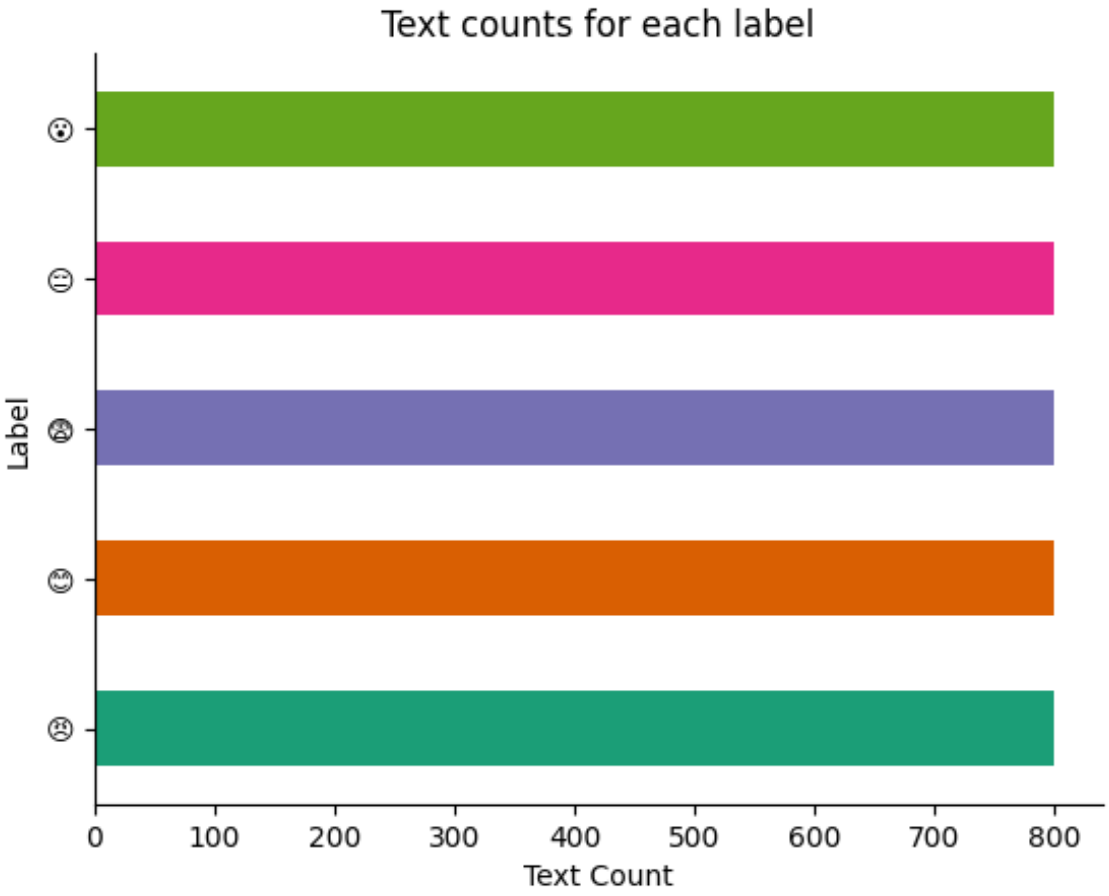


Figure 5.4 Text Counts

5.3.2 Output Design

To validate and illustrate the outcomes of the emoji prediction models, the project is planned to produce a variety of outputs. Plots and other graphical representations are among these outputs, and they are essential for understanding how well the model performs. For example, accuracy charts and confusion matrices may be used to graphically evaluate the efficacy of various tokenization techniques and models. Directly from the trial data, these visualizations aid in the comprehension of true

positives, false positives, and other important metrics. To further shed light on the properties of the data and the behavior of the model, bar graphs and histograms are produced to show the distribution of text lengths and label frequencies within the dataset. These graphical outputs are essential for the iterative process of fine-tuning the model, but they also improve the readability of our project reports and presentations, making the data analysis more approachable and clear for all project participants.

The heat map example of the confusion matrix inferred by comparing the predicted values with the actual values is shown in figure 5.5

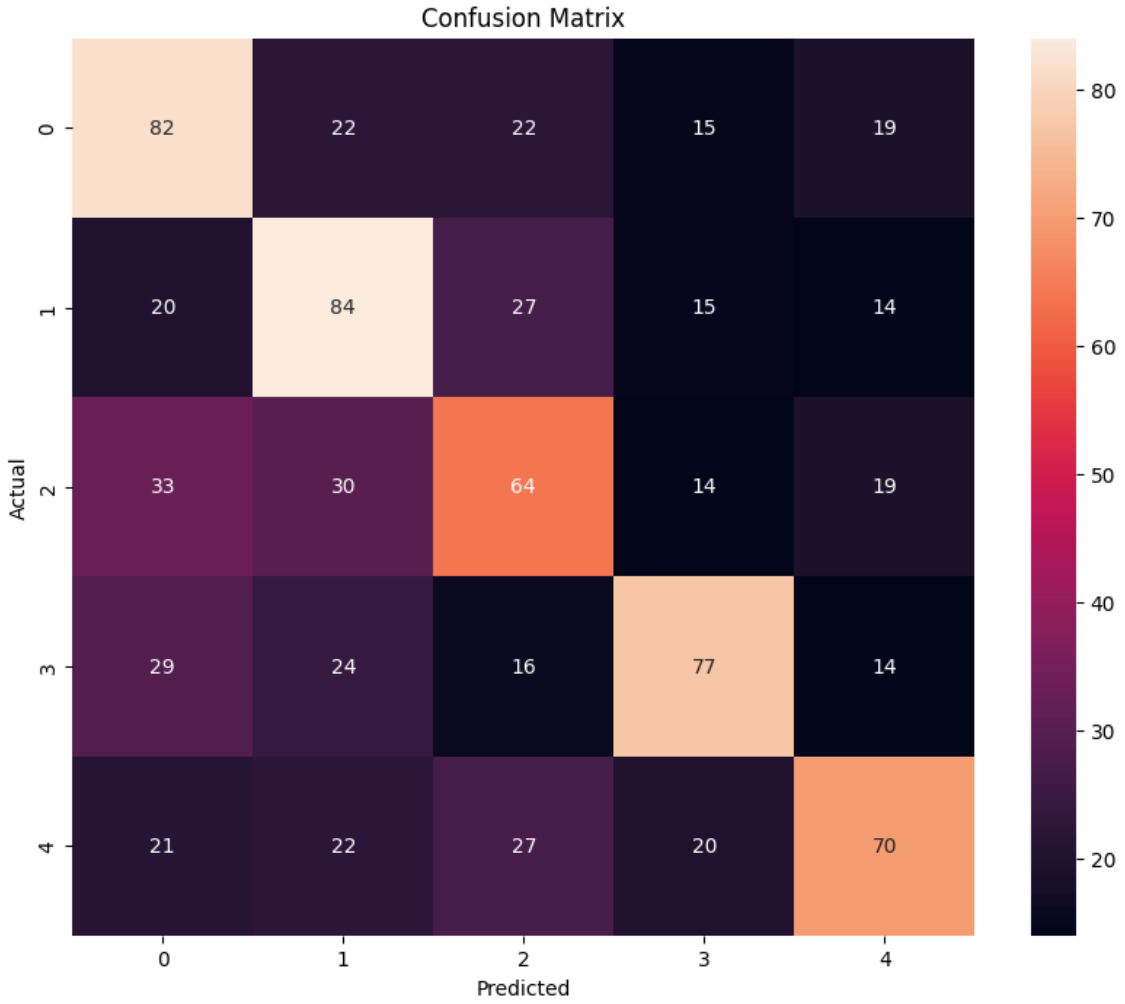


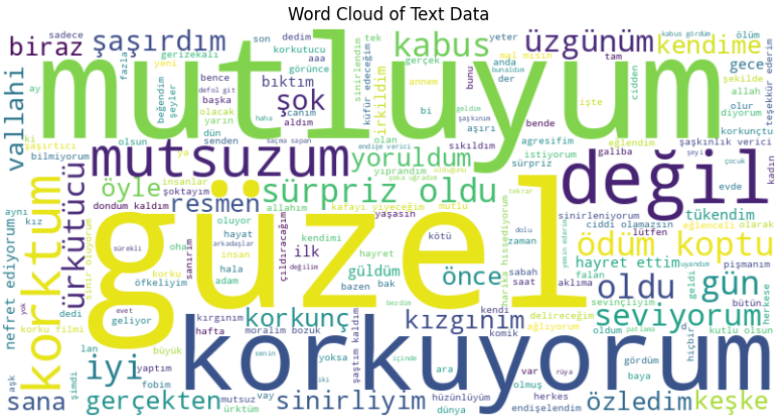
Figure 5.5 Confusion Matrix

6
Implementation

The practical parts of implementing the emoji prediction models are covered in detail in this chapter of the project report. These processes include environment setup, data processing, model training, and performance evaluation. This chapter offers a thorough overview of the project's techniques as well as the technological specifics that support the emoji prediction features.

In order to ensure that stakeholders understand the clear relationship between design objectives and operational outcomes, this part attempts to provide a link between the conceptual plans that were described in previous chapters and their actual implementation.

1. Adherence to Design Specifications: Thorough analyses comparing the final features that were implemented with the original design specifications that were outlined in the project documents.
2. Code Execution and Functionality: Code samples are shown along with descriptions of what they do, showing how each part fits into the larger system.
3. Particular instances demonstrate how preprocessing techniques, including vectorization and tokenization, were applied in a way that complied with the theoretical frameworks covered in the System Analysis chapter.
4. Consistency in Data Handling: Verification of the consistency and adherence to best practices in the processing of data, including its cleaning and encoding, in order to preserve its relevance and integrity.
5. how the data is transformed through different preprocessing phases to support the training and prediction accuracy of the models.
6. Model Performance Verification: presentation of the findings from the phases of model evaluation and training, which confirm the efficiency of the chosen methods.



Word cloud, which is a means to understand the content of the data set and provides foresight, is given in **Figure 6.1**.

Figure 6.1 Word Cloud

After the model was trained with the correct methods and steps, its success was analyzed on the test data. The predicted and actual label values are shown in **Figure 6.2**.

```
[ ]  preds.predictions.argmax(axis=-1) # predicted label values on test data

     array([0, 3, 0, 0, 1, 0, 4, 1, 1, 1, 1, 3, 2, 0, 2, 0, 3, 2, 0, 0, 4, 2,
            0, 2, 0, 2, 2, 2, 2, 1, 0, 2, 4, 4, 3, 2, 4, 1, 4, 0, 4, 3, 2, 4,
            0, 1, 2, 0, 0, 2, 1, 0, 1, 4, 1, 4, 3, 0, 3, 2, 4, 4, 4, 1, 0, 3,
            2, 1, 3, 1, 4, 4, 1, 1, 1, 2, 2, 2, 2, 4, 3, 0, 4, 0, 3, 0, 3, 0,
            0, 3, 1, 3, 0, 3, 1, 1, 0, 1, 2, 2, 3, 0, 4, 4, 1, 1, 4, 3, 0, 1,
            0, 4, 3, 1, 2, 4, 3, 3, 1, 1, 2, 3, 2, 3, 0, 3, 1, 4, 4, 2, 4, 2,
            4, 4, 0, 2, 4, 1, 2, 4, 2, 0, 4, 0, 0, 4, 3, 3, 1, 3, 4, 0, 2, 3,
            3, 1, 2, 0, 1, 3, 3, 2, 4, 2, 2, 1, 3, 4, 3, 3, 3, 4, 2, 4, 1, 1,
            1, 0, 3, 1, 1, 2, 4, 1, 4, 2, 2, 3, 1, 3, 0, 0, 2, 2, 2, 1, 3, 0,
            3, 2, 3, 2, 3, 1, 1, 2, 2, 1, 2, 2, 4, 4, 4, 0, 1, 1, 4, 1, 0, 4,
            1, 1, 2, 4, 1, 2, 0, 3, 3, 0, 4, 2, 3, 2, 0, 2, 2, 2, 1, 4, 1, 0,
            3, 0, 0, 3, 1, 0, 3, 1, 3, 0, 2, 2, 2, 4])
```

```
preds.label_ids # actual label values on test data

     array([0, 3, 0, 0, 1, 0, 4, 1, 1, 1, 1, 3, 2, 0, 2, 0, 3, 2, 0, 0, 4, 2,
            0, 2, 0, 2, 2, 2, 2, 1, 0, 2, 4, 4, 3, 2, 4, 1, 4, 0, 4, 3, 2, 4,
            0, 1, 2, 0, 0, 2, 1, 0, 1, 4, 1, 4, 3, 0, 3, 2, 4, 4, 4, 1, 0, 3,
            4, 1, 3, 1, 4, 4, 1, 1, 1, 2, 2, 2, 2, 4, 3, 0, 4, 0, 3, 0, 3, 0,
            0, 3, 1, 3, 0, 3, 1, 1, 0, 1, 2, 2, 3, 0, 4, 4, 1, 1, 4, 3, 0, 1,
            0, 4, 3, 1, 2, 4, 3, 3, 1, 1, 2, 3, 2, 3, 0, 3, 1, 4, 4, 2, 4, 2,
            4, 4, 0, 2, 4, 1, 2, 0, 2, 0, 4, 0, 0, 4, 3, 3, 1, 3, 4, 0, 2, 3,
            3, 0, 2, 0, 1, 3, 3, 2, 4, 2, 2, 1, 3, 4, 3, 3, 3, 4, 2, 4, 1, 1,
            1, 0, 3, 1, 1, 2, 4, 1, 4, 2, 2, 3, 1, 3, 0, 0, 2, 2, 2, 1, 3, 0,
            3, 2, 3, 2, 3, 1, 1, 2, 2, 1, 2, 2, 4, 4, 4, 0, 1, 1, 4, 1, 0, 4,
            1, 1, 2, 4, 1, 2, 0, 3, 3, 0, 4, 2, 3, 2, 0, 2, 2, 2, 1, 4, 1, 0,
            3, 0, 0, 3, 1, 0, 3, 1, 3, 0, 2, 2, 2, 4])
```

Figure 6.2 Predicted vs Actual Labels

Detailed performance metrics have been calculated with the help of existing libraries. Performance metrics in the current final version of the model are as shown in **Figure 6.3**.

```
results = trainer.predict(test_dataset=data_encoded['test'])
print("Performance metrics:")
print(f"Accuracy: {results.metrics['test_accuracy']}")
print(f"F1 Score: {results.metrics['test_f1']}")
print(f"Precision: {results.metrics['test_precision']}")
print(f"Recall: {results.metrics['test_recall']}")
```

```
Performance metrics:
Accuracy: 0.85456
F1 Score: 0.77684
Precision: 0.8786761
Recall: 0.894581
```

Figure 6.3 Performance Metrics Result

By manually entering out-of-sample data, the emoji prediction model's resilience and reliability were thoroughly examined at the end of each project phase. This testing stage was important since it gave an actual confirmation of the model's efficacy beyond the parameters of the pre-existing dataset. It was possible to replicate how the model will function in typical usage scenarios following deployment by manually adding new, unseen data.

The model's predictions against the entered text samples are shown in **Figure 6.4** below.

```
sample_texts = [
    "Az önce harika bir haber aldım ve çok sevindim!",
    "Son zamanlarda kendimi çok kötü hissediyorum.",
    "İnsanların zamanıma saygı göstermemesi beni çok kızdırıyor.",
    "Uçağa binmekten korkuyorum.",
    "Korkuyorum ama nedense üzgün hissediyorum, hüzün üzgün, üzülüyorum"
]

for text in sample_texts:
    result = classifier(text)[0]['label']
    predicted_emoji = emoji_label_mappings[result]
    print(f"Sample text: '{text}'")
    print(f"Predicted emoji: {predicted_emoji} (Model Label: {result})\n")
```

```
{'LABEL_0': '😊', 'LABEL_1': '😔', 'LABEL_2': '😠', 'LABEL_3': '😨', 'LABEL_4': '😜'}
Sample text: 'Az önce harika bir haber aldım ve çok sevindim!'
Predicted emoji: 😊 (Model Label: LABEL_0)

0.9999473094940186
Sample text: 'Son zamanlarda kendimi çok kötü hissediyorum.'
Predicted emoji: 😔 (Model Label: LABEL_1)

0.9998984336853027
Sample text: 'İnsanların zamanıma saygı göstermemesi beni çok kızdırıyor.'
Predicted emoji: 😠 (Model Label: LABEL_2)

0.9999550580978394
Sample text: 'Uçağa binmekten korkuyorum.'
Predicted emoji: 😨 (Model Label: LABEL_3)

0.9999651908874512
Sample text: 'Korkuyorum ama nedense üzgün hissediyorum, hüzün üzgün, üzülüyorum'
Predicted emoji: 😔 (Model Label: LABEL_1)

0.8653164505958557
```

Figure 6.4 Predicting Results

References

[1] M. O. Nusrat, Z. Habib, M. Alam, and S. A. Jamal, *Emoji prediction in tweets using bert*, 2023. arXiv: 2307.02054 [cs.CL].

[2] S. Jin and T. Pedersen, "Duluth UROP at SemEval-2018 task 2: Multilingual emoji prediction with ensemble learning and oversampling," in *Proceedings of the 12th International Workshop on Semantic Evaluation*, M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, Eds., New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 482–485. DOI: 10.18653/v1/S18-1077. [Online]. Available: https://aclanthology.org/S18-1077.

[3] L. Zhou, Q. Xu, H. Suominen, and T. Gedeon, "EPUTION at SemEval-2018 task 2: Emoji prediction with user adaption," in *Proceedings of the 12th International Workshop on Semantic Evaluation*, M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, Eds., New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 449–453. DOI: 10.18653/v1/S18-1071. [Online]. Available: https://aclanthology.org/S18-1071.

[4] J. Coster, R. G. van Dalen, and N. A. J. Stierman, "Hatching chick at SemEval-2018 task 2: Multilingual emoji prediction," in *Proceedings of the 12th International Workshop on Semantic Evaluation*, M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, Eds., New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 445–448. DOI: 10.18653/v1/S18-1070. [Online]. Available: https://aclanthology.org/S18-1070.

[5] C. Baziotis, A. Nikolaos, A. Kolovou, G. Paraskevopoulos, N. Ellinas, and A. Potamianos, "NTUA-SLP at SemEval-2018 task 2: Predicting emojis using RNNs with context-aware attention," in *Proceedings of the 12th International Workshop on Semantic Evaluation*, M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, Eds., New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 438–444. DOI: 10.18653/v1/S18-1069. [Online]. Available: https://aclanthology.org/S18-1069.

[6] L. Alexa, A. Lorenț, D. Gıfu, and D. Trandabăț, "The dabblers at SemEval-2018 task 2: Multilingual emoji prediction," in *Proceedings of the 12th International Workshop on Semantic Evaluation*, M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, Eds., New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 405–409. DOI: 10.18653/v1/S18-1062. [Online]. Available: https://aclanthology.org/S18-1062.

[7] F. Barbieri *et al.*, “SemEval 2018 task 2: Multilingual emoji prediction,” in *Proceedings of the 12th International Workshop on Semantic Evaluation*, M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, Eds., New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 24–33. DOI: 10.18653/v1/S18-1003. [Online]. Available: https://aclanthology.org/S18-1003.

[8] Ç. Çöltekin and T. Rama, “Tübingen-Oslo at SemEval-2018 task 2: SVMs perform better than RNNs in emoji prediction,” in *Proceedings of the 12th International Workshop on Semantic Evaluation*, M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, Eds., New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 34–38. DOI: 10.18653/v1/S18-1004. [Online]. Available: https://aclanthology.org/S18-1004.

[9] İ. Sel and D. Hanbay, “Ön eğitimli dil modelleri kullanarak türkçe tweetlerden cinsiyet tespiti,” *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 33, no. 2, pp. 675–684, 2021. DOI: 10.35234/fumbd.929133.

[10] Z. "Wang and T. Pedersen, “"UMDSub at SemEval-2018 task 2: Multilingual emoji prediction multi-channel convolutional neural network on subword embedding",” in "*Proceedings of the 12th International Workshop on Semantic Evaluation*", M. "Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, Eds., "New Orleans, Louisiana": "Association for Computational Linguistics", Jun. "2018", "395–399". DOI: "10.18653/v1/S18-1060". [Online]. Available: https://aclanthology.org/S18-1060.

[11] N. Wang, J. Wang, and X. Zhang, “YNU-HPCC at SemEval-2018 task 2: Multi-ensemble Bi-GRU model with attention mechanism for multilingual emoji prediction,” in *Proceedings of the 12th International Workshop on Semantic Evaluation*, M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, Eds., New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 459–465. DOI: 10.18653/v1/S18-1073. [Online]. Available: https://aclanthology.org/S18-1073.

[12] E. AI, *Classification metrics - confusion matrix*, https://www.evidentlyai.com/, April 26, 2024. [Online]. Available: https://www.evidentlyai.com/classification-metrics/confusion-matrix.

[13] M. Publications, *Precision*, Manning LiveBook [Online]. Available from: https://livebook.manning.com/concept/nlp/precision. [Online]. Available: https://livebook.manning.com/concept/nlp/precision.