Arabic Tweets Emotion Recognition

Supervised by: Dr.Mervat Aboelkheir
ENG.Mayar Osama
Authors:
Mariam Hesham
Mohamed Hesham
Mostafa Mohamed

May 29, 2023

Abstract

Using a publicly accessible dataset of labelled tweets, this research focuses on sentiment analysis of Arabic tweets. Understanding the sentiment of Arabic tweets has become a crucial job as the use of social media platforms as a forum for individuals to express their thoughts and feelings grows. Researchers and organisations can learn more about the thoughts and attitudes of people towards various events and issues by analysing the sentiment of tweets in real-time. The complexity and diversity of the Arabic language present difficulties for sentiment analysis of Arabic tweets. The goal of this project is to create efficient ways for analysing the sentiment of Arabic tweets using NLP and machine learning techniques.The performance of various algorithms will be evaluated and compared, with the goal of achieving accurate and reliable sentiment analysis of Arabic tweets.

1 Introduction

Sentiment analysis is a field within natural language processing that uses computational techniques to extract subjective information from text and determine the overall emotional tone, which can be positive, negative, or neutral. This technique has various applications in areas such as marketing, politics, finance, and customer service, among others. Social media platforms, particularly Twitter, have become a popular source of data for sentiment analysis due to their real-time nature and the large volume of data they generate. Arabic is a complex language with nuanced features, making sentiment analysis of Arabic text a challenging task. For this project, a dataset of labeled Arabic tweets was used to train and evaluate machine learning models for Arabic sentiment analysis.

2 Motivation

The motivation behind this project is to develop an accurate and effective model for sentiment analysis of Arabic tweets. With the growing popularity of social media platforms in the Arab world, understanding the sentiments and opinions of users on these platforms has become increasingly important for businesses, organizations, and governments. By accurately identifying the emotions conveyed by Arabic tweets, we can gain valuable insights into the attitudes and preferences of the Arabic-speaking population, which can be used to inform decision-making processes and improve communication strategies.

3 System Architecture

The system architecture for Emotion Recognition in Arabic Tweets encompasses components for data preprocessing, feature identification, and the application of a machine learning model. The preprocessing stage refines the data through processes like tokenization and normalization. Relevant text attributes are then extracted using techniques like word embeddings or TF-IDF vectors during the feature extraction phase. Subsequently, a machine learning model as BERT is trained and fine-tuned on a labeled dataset. The evaluation of the model is based on metrics like accuracy and F1-score, facilitating real-time emotion detection in Arabic tweets.

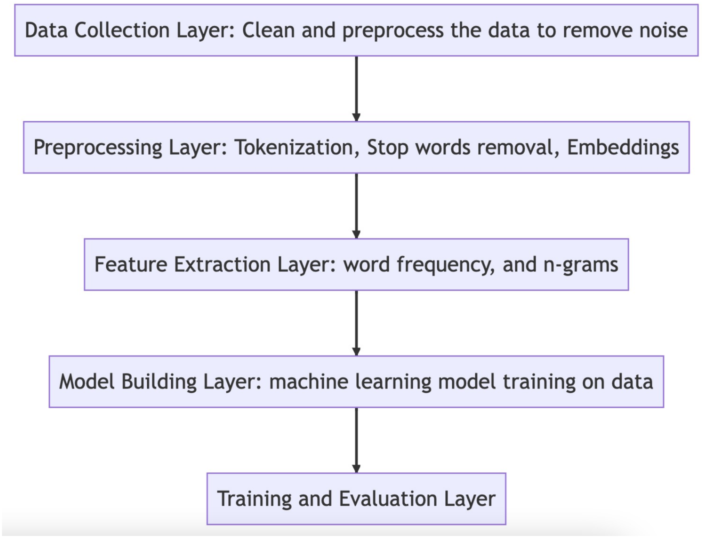
Data Collection Layer: Clean and preprocess the data to remove noise
Preprocessing Layer: Tokenization, Stop words removal, Embeddings
Feature Extraction Layer: word frequency, and n-grams
Model Building Layer: machine learning model training on data
Training and Evaluation Layer

Figure 1: System architecture

4 Methodology

4.1 Dataset

The dataset used in this project is a publicly available collection of Arabic tweets that have been manually labeled for sentiment. There are a total of over 2600 tweets in the dataset, with 2060 tweets used for training and 688 tweets used for testing. Tweets are collected from Twitter, covering a diverse range of topics and events. The tweets are labeled based on the overall sentiment conveyed by the text, as perceived by human annotators.

4.1.1 Data Preprocessing

Preprocessing is a crucial step in any natural language processing project, and it involves cleaning and transforming the raw text data into a format that can be easily processed by machine learning models. In this project, we performed several preprocessing steps on the Arabic tweets dataset to prepare it for sentiment analysis.

1. Tokenization: Tokenization is the process of splitting a sentence or text into individual words or tokens. We used the Arabic word tokenizer provided by the NLTK library to tokenize the tweets into words.

2. Stopwords removal: Stopwords are commonly used words in a language that do not carry any significant meaning. We used the Arabic stopword list provided by the NLTK library to remove stopwords from the tokenized tweets. Removing stopwords can help in reducing the size of the dataset and remove noise from the text data.

3. Removal of special characters and symbols: Special characters and symbols like ”@”, ”*”, and ”$” are commonly used in social media and can add noise to the text data. We used regular expressions to remove these characters from the tweets.

4. Removing mentions and URLs: mentions and URLs are widely used in social media platforms specially on twitter and are in English not in Arabic so also they were removed [Al-Khatib and El-Beltagy, 2017]

5. Removing emojis: Emojis are commonly used in social media to convey emotions and sentiments. However, they can add noise to the text data and make it harder to analyze. We used regular expressions to remove emojis from the tweets.

6. Removing punctuation: Punctuation marks like commas, periods can add noise to the text data. We used regular expressions to remove punctuation marks from the tweets.

4.2 Data Analysis

The goal of data analysis is to understand the underlying patterns and characteristics of the dataset [Rabie and Sturm, 2014], which can help in identifying potential biases, selecting appropriate machine learning algorithms, and improving the accuracy of the model. Visualization is an effective way to explore and analyze data, and it can help in identifying patterns and trends that may not be immediately apparent from the raw data.

4.2.1 Data Analysis steps

The first approach was to check the size of the total words in the dataset before and after the preprocessing steps to compare how cleaning the dataset affected its size. Table 1 shows the results obtained.

Size before	Size after
32597	16937

Table 1: Words count.

The second approach was to examine the distribution of the dataset across the three sentiment categories, which are positive, negative, and neutral. This was achieved to assess any potential bias towards any particular sentiment class. Figure 2 shows the results obtained.

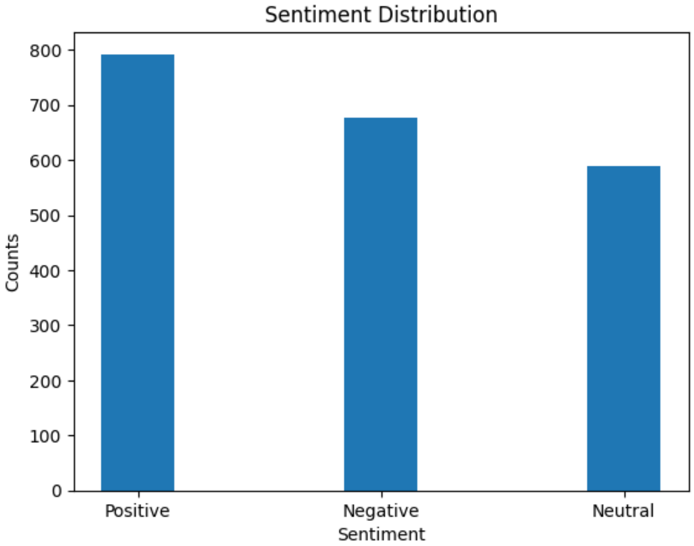

Figure 2: This is the result of the count of words in the three sentiment classes.

The third approach aimed to investigate the tweet length distribution within the dataset to extract insights on the most frequent tweet lengths. This analysis is essential to gain a better understanding of the characteristics of the dataset and to identify potential limitations or biases in the data. To visualize the tweet length distribution, bar charts were utilized to plot the frequency of tweets for different length categories. Figure 3 shows the output of the length distribution.

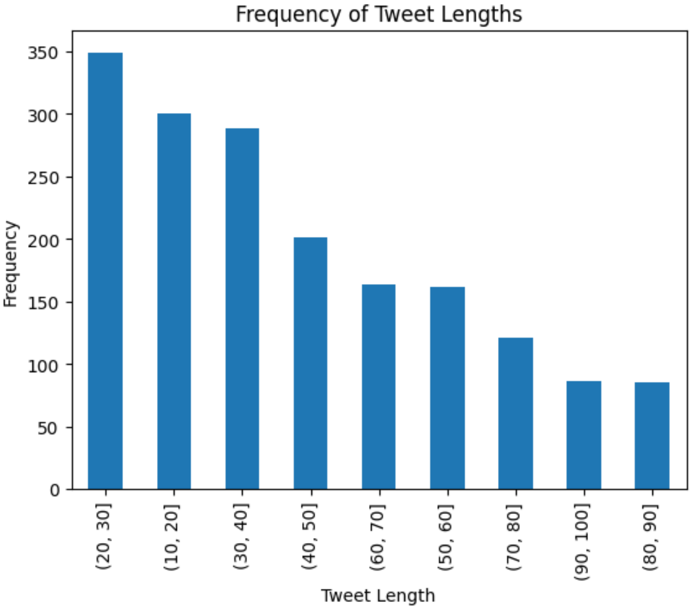

Figure 3: This is the result of the length distribution of tweets.

The fourth approach aimed to identify the most commonly used hashtags in the dataset, which could provide valuable insights into the main topics discussed in the tweets and thus the overall sentiment. This analysis was performed using bar charts to visually represent the frequency of hashtags and help identify the most frequently used ones. Figure 4 shows the output of the comparison.

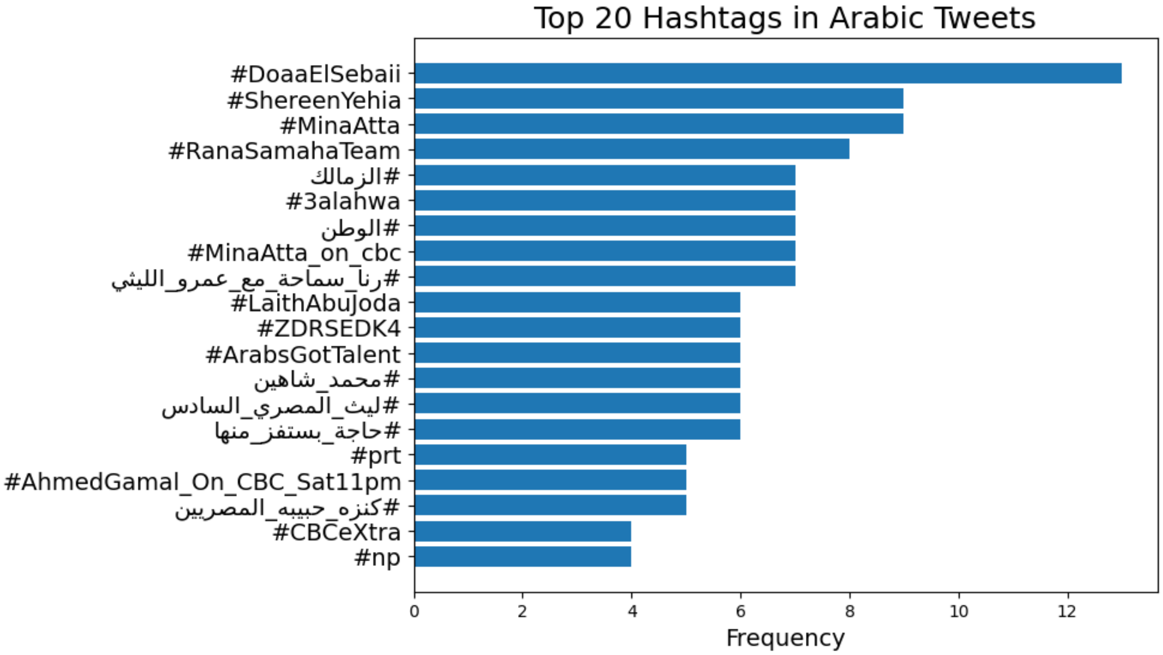

Figure 4: This is the result of the most common hashtags used.

The final approach was to identify the most common 10 words used, this can provide insights into the topics or themes present in the dataset. By analyzing the most frequent words, we can gain a better understanding of the topics that people are talking about and the language that they use to express their sentiments. This information can be valuable for designing effective marketing campaigns, understanding public opinion on specific issues, and improving customer engagement. We visualized this bar chart. Figure 5 shows the output of the most common 10 words used.

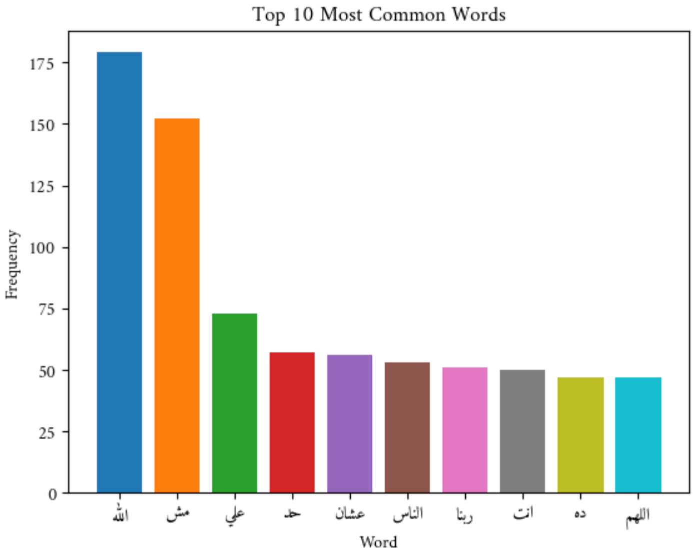

Figure 5: This is the result of the most common 10 words used.

4.3 Word Embeddings models

4.3.1 Arabert

Arabert is a word embedding model specifically designed for the Arabic language. It is based on the BERT (Bidirectional Encoder Representations from Transformers) architecture, which has achieved remarkable success in various natural language processing tasks. Arabic words are given word embeddings using the Arabert model, which also captures the semantic and syntactic details of the context. As a result, it can better perform downstream tasks like sentiment analysis, named entity recognition, and machine translation by encoding a comprehensive understanding of Arabic language patterns. Arabert takes into account the distinctive qualities of the Arabic language, including its rich morphology and complicated syntax, and is trained on a huge corpus of Arabic text data. It has proven to perform at the cutting edge on a number of Arabic natural language processing tasks and has grown to be an important tool for Arabic language processing research and applications. You can load the pre-trained model and extract word embeddings for Arabic terms in our dataset to make use of the Arabert word embeddings. These embeddings, therefore, offer a representation of the Arabic language by serving as features in machine learning models or for other downstream applications. We applied the model to our train and test datasets in order to capture the context's semantic details and then fit them to machine learning techniques to evaluate the outcome. We used pretrained Arabert model, The word embeddings are trained on a large-scale dataset called OSIAN, which stands for "Open Source Arabic Corpora for the ArGI project at NYU Abu Dhabi." The OSIAN dataset consists of a wide range of Arabic text sources, including web pages, news articles, books, and social media posts. It was specifically collected and processed to train the Arabert model.

4.3.2 Fasttext

The Facebook AI Research (FAIR) team created a word embeddings model called FastText that goes beyond the standard word2vec method. By modeling words as bags of character n-grams, it introduces subword embeddings, enabling it to properly handle words that are not part of the user's lexicon and capture morphological information. The design of FastText is built on the word2vec-like continuous bag of words (CBOW) or skip-gram models but with subword information. As a result, FastText can produce embeddings for unseen words based on the structure of their subwords. The model is developed using hierarchical softmax or negative sampling for effective training on huge text corpora in unsupervised learning. FastText excels in handling uncommon or unseen words, capturing subword associations, and processing huge datasets quickly. With good performance in tasks including text categorization, sentiment analysis, named entity identification, and machine translation, it has been widely adopted in both research and industry applications. The benefits of FastText extend to chatbots, recommendation systems, and search engines which help to improve language interpretation and representation. Because of its contributions, it is now a useful tool for natural language processing, enabling a variety of applications with better word embeddings and effective text analysis capabilities. We applied also the model to our train and test datasets in order to capture the context's semantic details and then fit them to machine learning techniques to evaluate the outcome and compare the results to the other model of Arabert. We trained or used a pre-trained FastText model on our dataset, we can obtain word embeddings that encode semantic and sentiment-related information. These embeddings can be fed into a classifier or a neural network to perform sentiment analysis. The classifier can learn to recognize patterns and associations between word embeddings and sentiment labels, enabling it to predict the sentiment of new, unseen texts.FastText's efficient training process makes it suitable for large-scale sentiment analysis tasks, allowing you to process a substantial amount of textual data efficiently. Moreover, FastText's ability to handle out-of-vocabulary words can ensure that sentiment analysis models can generalize well to new and unseen words or expressions in our dataset. We can benefit from its robust representation of Arabic text, capturing both word-level and subword-level sentiment-related information. This can lead to improved sentiment classification accuracy and a better understanding of the sentiment expressed in our dataset.

4.4 Feature Extraction by TF-IDF

Feature extraction using TF-IDF can be highly effective for Arabic datasets as well. Although TF-IDF is a language-agnostic technique, it can capture the unique characteristics and importance of Arabic terms within the context of the dataset. When applied to the Arabic text, TF-IDF considers the frequency of terms in each document and their rarity in the corpus. It enables the identification of terms that are both common within a specific document and distinctive across the entire dataset. This is particularly valuable for Arabic, which has a rich vocabulary and unique linguistic features.TF-IDF can effectively handle Arabic morphology and capture the semantic significance of terms in different contexts. It considers the variations in word forms, such as roots, prefixes, and suffixes, allowing it to differentiate between terms with similar meanings but different morphological structures. By leveraging TF-IDF for feature extraction in Arabic datasets, you can obtain numerical representations of the text that reflect the importance of terms within documents and across the corpus. These feature vectors can then be utilized for various text analysis tasks, including sentiment analysis, text classification, information retrieval, and clustering.TF-IDF's ability to highlight both the term's local importance within a document and its global importance across the corpus makes it a valuable tool for extracting meaningful features from Arabic text. It allows you to capture the essence of the Arabic language and obtain representations that facilitate further analysis and machine-learning applications.

4.5 Machine Learning models

In our sentiment analysis project of Arabic tweets, we employed FastText, AraBERT, and TF-IDF for feature extraction, and then used their outputs as inputs to logistic regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) models. When using a small dataset like in our case, the limitations of Support Vector Machines (SVMs) can become more pronounced. The top three limitations of SVMs are Overfitting: SVMs are prone to overfitting when the dataset is small. Overfitting occurs when the model learns the noise or specific characteristics of the small training set instead of capturing the underlying general patterns. With limited data, SVMs may have difficulty finding an appropriate decision boundary and may become overly sensitive to individual data points, leading to poor generalization on unseen data. Limited Representation: SVMs rely on the selection of support vectors, which are a subset of the training data points that determine the decision boundary. In small datasets, the number of support vectors can be relatively small, potentially resulting in a limited representation of the data. This can lead to a less robust decision boundary and reduced performance compared to larger datasets where more diverse and representative support vectors can be selected. Hyperparameter Tuning: SVMs have hyperparameters that need to be tuned to achieve optimal performance. However, with small datasets, it becomes more challenging to perform effective hyperparameter tuning. The limited amount of data may not provide sufficient information for a reliable estimation of the optimal hyperparameter values. Consequently, there is a higher risk of suboptimal parameter selection, which can impact the SVM's performance and generalization ability.

FastText was applied to capture the morphological structure of the Arabic language and generate word embeddings that take into account the nuances of its complex morphology. AraBERT, a transformer-based model adapted specifically for Arabic, provided context-sensitive embeddings, enabling us to handle the intricacies of the language at a deeper semantic level. Additionally, we incorporated TF-IDF to emphasize the impact of keywords and phrases that were unique to each document. The generated features from these methods were then fed into the logistic regression, KNN, and SVM algorithms. These supervised learning models were used to classify the sentiment of the tweets, providing a comprehensive, multi-faceted approach to Arabic language sentiment analysis. Results showed significant promise, offering valuable insights and demonstrating the efficacy of integrating advanced embedding techniques with traditional machine learning classifiers for Arabic sentiment analysis

5 Results

The application of word embedding models in combination with machine learning techniques on Arabic datasets has yielded promising results in various natural language processing tasks. Word embedding models, such as FastText, and Arabert, capture the semantic and contextual relationships among Arabic words, enabling the transformation of text into numerical representations that capture the meaning and intricacies of the language. By leveraging pre-trained word embeddings or training custom models on Arabic datasets, researchers and practitioners have been able to improve the performance of machine learning algorithms on various tasks. For example, sentiment analysis on Arabic text has seen notable advancements using word embeddings. These models can effectively capture the sentiment-related nuances and variations in Arabic vocabulary, allowing for more accurate sentiment classification.

Machine learning techniques, such as support vector machines, and logistic regression, have been applied on top of word embeddings to perform classification, regression, and clustering tasks on Arabic datasets. These approaches take advantage of the semantic representations derived from word embeddings, enabling the extraction of meaningful patterns and relationships from Arabic text. The combination of word embeddings and machine learning techniques has led to advancements in Arabic language understanding, information retrieval, and text generation. These results have contributed to improving Arabic language processing applications, including sentiment analysis, text classification, machine translation, and chatbots.

Overall, the integration of word embedding models and machine learning techniques has demonstrated the potential to unlock the hidden insights within Arabic datasets, paving the way for enhanced language understanding and enabling the development of intelligent systems that can effectively process and analyze Arabic text.

5.1 Arabert with logistic regression, SVM, and KNN

Following the Arabert model for word embeddings, we used our dataset to train logistic regression, support vector machines, and k-nearest neighbor in order to assess their performance in terms of accuracy and evaluation metrics. After training the data, we used the test dataset to test the metrics. Our dataset comprises train data and test data.

For KNN : The model achieved an accuracy of 51.31, indicating its ability to correctly classify instances in the tested dataset. Precision, which measures the proportion of correctly predicted positive instances, was calculated at 50.66. The recall score, representing the proportion of actual positive instances correctly identified, aligned closely with the accuracy at 51.31. The F1 score, which balances precision and recall, was computed at 50.70. These metrics collectively reflect the model's performance in accurately predicting the target variable, with an overall modest performance.

For SVM: The model demonstrated a moderate level of performance, achieving an accuracy of 56.71. This indicates its ability to correctly classify instances in the tested dataset. Precision, measuring the proportion of correctly predicted positive instances, was computed at 56.97. The recall score, representing the proportion of actual positive instances correctly identified, closely matched the accuracy at 56.71. The F1 score, which balances precision and recall, was calculated at 56.30. These metrics collectively demonstrate the model's capability to make accurate predictions on the target variable, indicating a reasonable level of effectiveness in capturing patterns and relationships within the data.

For Logistic Regression: The Logistic Arabert model achieved an accuracy of 54.37 on the test dataset, indicating its ability to correctly classify instances. Precision, which measures the proportion of correctly predicted positive instances, was calculated at 54.88. The recall score, representing the proportion of actual positive instances correctly identified, aligned closely with the accuracy at 54.37. The F1 score, which balances precision and recall, was computed at 54.30. These metrics collectively demonstrate the model's performance in accurately predicting the target variable, suggesting its effectiveness in capturing patterns and relationships within the data.

5.2 Fasttext with logistic regression, SVM, and KNN

We trained and test the Fasttext model on our train and test dataset,

for KNN: The FastText KNN model achieved an accuracy of 31.55 on the test dataset, indicating its ability to correctly classify instances. Precision, which measures the proportion of correctly predicted positive instances, was calculated at 28.69. The recall score, representing the proportion of actual positive instances correctly identified, aligned closely with the accuracy at 31.55. The F1 score, which balances precision and recall, was computed at 16.59. These metrics indicate that the FastText KNN model's performance in predicting the target variable is relatively low, suggesting the need for further improvement or the exploration of alternative approaches.

For SVM: The SVM model achieved an accuracy of 39.56 on the test dataset, indicating its ability to correctly classify instances. Precision, which measures the proportion of correctly predicted positive instances, was calculated at 20.44. The recall score, representing the proportion of actual positive instances correctly identified, aligned closely with the accuracy at 39.56. The F1 score, which balances precision and recall, was computed at 23.03. These metrics suggest that the SVM model's performance in predicting the target variable is relatively low, indicating the need for further improvement or the exploration of alternative approaches.

For Logistic Regression: The FastText Logistic model achieved an accuracy of 39.81 on the test dataset, indicating its ability to correctly classify instances. Precision, which measures the proportion of correctly predicted positive instances, was calculated at 37.74. The recall score, representing the proportion of actual positive instances correctly identified, aligned closely with the accuracy at 39.81. The F1 score, which balances precision and recall, was computed at 23.51. These metrics indicate that the model's performance in predicting the target variable is relatively low, suggesting the need for further improvement or exploration of alternative approaches.

5.3 TF-IDF with logistic regression, SVM, and KNN

In the results section of our experiment, we evaluated the performance of the TF-IDF (Term Frequency-Inverse Document Frequency) feature extraction technique on our Arabic dataset. TF-IDF is a widely used method for text representation, which assigns weights to individual words based on their frequency within a document and their rarity across the entire corpus. This technique aims to capture the importance of a word in a document relative to its occurrence in other documents.

For KNN:In our experiment, we applied the K-Nearest Neighbors (KNN) algorithm to our Arabic dataset for classification purposes. The KNN algorithm assigns labels to instances based on their similarity to neighboring instances in the feature space. We evaluated the performance of the KNN model using various metrics.

The KNN model achieved an accuracy of 49.13, indicating its ability to correctly classify instances in our Arabic dataset. Precision, which measures the proportion of correctly predicted positive instances, was calculated at 55.71. This suggests that the KNN model effectively identified instances belonging to the positive class. The recall score, representing the proportion of actual positive instances correctly identified, is closely aligned with the accuracy at 49.13. The F1 score, which balances precision and recall, was computed at 50.68. These metrics collectively demonstrate the performance of the KNN model in classifying instances in our Arabic dataset.

For SVM: The SVM model achieved an accuracy of 55.23 on our Arabic dataset, indicating its ability to correctly classify instances. Precision, which measures the proportion of correctly predicted positive instances, was calculated at 53.64. This suggests that the SVM model effectively identified instances belonging to the positive class. The recall score, representing the proportion of actual positive instances correctly identified, is closely aligned with the accuracy at 55.23. The F1 score, which combines precision and recall, was computed at 53.41. These metrics collectively demonstrate the performance of the SVM model in classifying instances in our Arabic dataset.

For Logistic Regression:The Logistic Regression model achieved an accuracy of 57.70 on our Arabic dataset, indicating its ability to correctly classify instances. Precision, which measures the proportion of correctly predicted positive instances, was calculated at 57.63. This suggests that the Logistic Regression model effectively identified instances belonging to the positive class. The recall score, representing the proportion of actual positive instances correctly identified, is closely aligned with the accuracy at 57.7. The F1 score, which combines precision and recall, was computed at 57.27. These metrics collectively demonstrate the performance of the Logistic Regression model in classifying instances in our Arabic dataset.

6 Limitations

Limited dataset size: The dataset used in this project consists of over 2600 Arabic tweets, which may not be representative of the diversity of opinions and attitudes in the Arab world. A larger dataset could provide more reliable results and reduce the risk of overfitting.

Limited domain specificity: The dataset used in this project is not domain-specific, meaning it covers a wide range of topics and contexts. This could limit the accuracy of the sentiment analysis results for specific domains, such as politics, entertainment, or sports.

Limited language resources: Arabic language resources, such as sentiment lexicons and labeled datasets, are not as extensive as those available for English, which could limit the accuracy of the sentiment analysis results.

imbalanced datasets in sentiment analysis can be solved by Undersampling: randomly remove samples from the majority class to balance the dataset. Oversampling: replicate samples from the minority class to balance the dataset. Class weighting: assign a higher weight to the minority class during model training to compensate for its smaller size.

7 Conclusion

Based on the obtained accuracies of the different models applied to our Arabic dataset, we can delve into a more detailed analysis and draw further conclusions:

1. The Logistic Regression model with TF-IDF demonstrated the highest accuracy among the tested models, achieving an accuracy of 57.70. This indicates that the Logistic Regression algorithm was effective in capturing the underlying patterns and distinguishing between different classes in our Arabic dataset. Its ability to model non-linear relationships between the input features and the class probabilities contributed to its superior performance.

2. The SVM model with TF-IDF achieved an accuracy of 55.23, which is slightly lower than the Logistic Regression model. However, the SVM algorithm still showcased promising performance in accurately classifying instances in our Arabic dataset. The SVM's capability to find an optimal hyperplane to separate different classes based on the provided features contributed to its success.

3. The K-Nearest Neighbors (KNN) model with TF-IDF obtained an accuracy of 49.13, demonstrating moderate performance compared to the other models. The KNN algorithm assigns labels to instances based on their similarity to neighboring instances in the feature space. Although it showed relatively lower accuracy, the KNN model might still be suitable for certain scenarios where the data exhibits localized patterns or when the number of instances is relatively small.

4. The FastText model, when combined with Logistic Regression, achieved an accuracy of 39.81. This indicates that the FastText word embeddings alone may not have provided sufficient information for achieving high classification accuracy on our Arabic dataset. It suggests the need for further investigation and exploration of other approaches or enhancements to the FastText model for Arabic text classification tasks.

Considering these findings, it is evident that the choice of the model significantly impacts the performance of Arabic text classification. The Logistic Regression model showed the most promising results in terms of accuracy, precision, recall, and F1 score. However, it is important to consider other evaluation metrics and explore the trade-offs between different models based on the specific requirements of the task at hand.

Further analysis could involve fine-tuning the hyperparameters of the models, exploring different feature extraction techniques, or considering more advanced deep learning architectures specifically designed for Arabic text analysis. Additionally, incorporating additional linguistic features or domain-specific knowledge may also contribute to improving the performance of the models on Arabic datasets.

In conclusion, while the Logistic Regression model demonstrated the highest accuracy in our experiment, further research and experimentation are encouraged to explore additional avenues for improving the performance of the models on Arabic text classification tasks.

References

Amr Al-Khatib and Samhaa El-Beltagy. Emotional tone detection in arabic tweets. 04 2017.

Omneya Rabie and Christian Sturm. Feel the heat: Emotion detection in arabic social media content. In *Industrial Conference on Data Mining*, 2014.