Sentiment Analysis For Arabic Tweets

Ahmed Lokma , Ahmed Mohamed , Seif Ramy , TEAM 9

May 26, 2023

Abstract

Sentiment Analysis for Arabic Tweets has become a crucial tool for analyzing public opinion towards various issues, thanks to the widespread use of social media platforms like Twitter in the Arab world. However, analyzing Arabic text data poses unique challenges due to the complexity of the language and culture. As a result, sentiment analysis of Arabic tweets requires specialized tools and techniques to address these challenges. This report provides a comprehensive overview of Sentiment Analysis for Arabic Tweets, including its motivation, data analysis, and preprocessing techniques, as well as the challenges that come with this process. The report also proposes an architecture for sentiment analysis of Arabic tweets to improve the accuracy of the analysis

1 Introduction

Sentiment analysis is a technique that uses machine learning and natural language processing to identify whether a text has a positive, negative, or neutral tone, as well as other relevant information such as emotions and attitudes expressed in the text. This technique is useful in fields such as marketing, customer service, political analysis, and social research. It has become even more important with the increasing amount of digital content and social media use.

Social media platforms have become popular sources of data for sentiment analysis, since they allow people to express their opinions and emotions publicly and in real-time. Twitter is one of the most popular social media platforms globally, with millions of users tweeting daily about a range of topics. In the Arab world, Twitter has become a popular platform for discussing politics, social issues, and entertainment, among other topics.

The Arabic language is the fifth most widely spoken language globally, with over 400 million speakers. However, analyzing Arabic text data poses unique challenges due to its complexity, including the use of non-standard characters, diacritics, and variations in grammar and syntax. Therefore, sentiment analysis of Arabic tweets requires specialized tools and techniques to account for these challenges.

2 Dataset

The utilized dataset in this study is the "Arabic Sentiment Twitter Corpus", which comprises a collection of tweets conveying positive and negative sentiments extracted from the Twitter platform. This dataset was procured in April of 2019 and consists of 58,000 Arabic tweets, with 47,000 tweets reserved for training and the remaining 11,000 for testing purposes. Each tweet has been annotated with either a positive or negative label. The dataset is balanced and has been collected using a comprehensive lexicon of positive and negative emojis.

3 Data preprocessing

3.1 Data Cleaning

Data cleaning is considered as an essential stage for data analysis, due to it eliminates unnecessary data with the aim of achieving better results in sentiment analysis. Our data cleaning process involved the following steps:[AR23]

- Remove mentions and URLs.
- Remove any non-Arabic characters.
- Remove Punctuation.
- Remove Stopwords.
- Remove elongation.
- Remove emojis.
- Remove retweets.
- Remove Arabic diacritics marks.
- Remove extra whitespace.
- Tokenization.

Diacritic Marks	Characters
Fatha	◌َ
Tashdeed	◌ّ
Tanwin Fath	◌ً
Damma	◌ُ
Tanwin Damm	◌ٌ
Kasra	◌ِ
Tanwin Kasr	◌ٍ
Sukun	◌ْ

Figure 1: Arabic Diacritics marks[AR23]

3.2 Normalization

Normalization refers to the process of reducing letters to their most basic form. Given the rich morphological complexity of the Arabic language, normalization is essential to simplify the language and make it more manageable for analysis and processing. Normalization was also done in data preprocessing as it's an important step.[AR23]

Letter	Normalized Form
ا,آ,أ,إ	ا
ى	ي
ئ	ء
ؤ	ء
ة	ه
ک	ك

Figure 2: Some letters in normalization form [AR23]

3.3 Stemming

Stemming in Arabic NLP (Natural Language Processing) refers to the process of reducing Arabic words to their root form, also known as the stem. This involves removing any affixes or prefixes that modify the word's meaning and converting the word to its basic form. On the other hand, there's some cases where some Arabic words lose their meaning after stemming operation. So we tried to handle this by making a list of words that should not be stemmed.

4 Data Analysis

Data analysis in NLP (Natural Language Processing) involves the application of statistical, machine learning, and computational techniques to extract insights and patterns from natural language data. The goal of NLP data analysis is to uncover meaningful information from large volumes of text data. In our project, we conducted two data analysis for our data:

1. Most common words in tweets
2. Tweet lengths and their frequencies

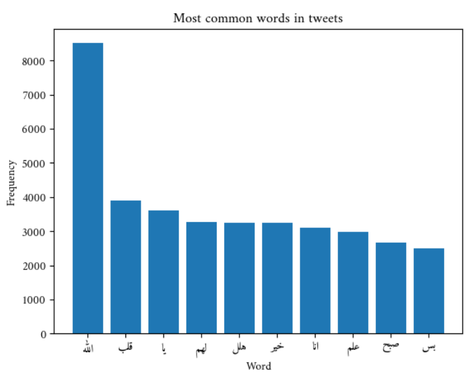

Figure 3: Most common words in tweets

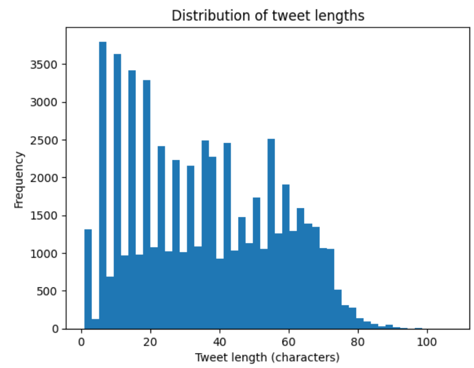

Figure 4: Tweet length and their frequencies

5 System Architecture

System architecture is a critical component of any software development project, as it defines the overall structure and organization of the system.Our system architecture is divided into several steps:

1. Data Collection
2. Cleaning dataset and dataset preparation
3. Feature Extraction
4. Sentiment Analysis

Feature extraction is an important step in any NLP pipeline, as it helps to transform raw text data into a set of numerical features that can be easily processed by machine learning algorithms. Feature extraction could be done using several methods such as bag of words, tf-idf, word-embeddings or Arabert. Sentiment analysis could be done using machine learning model such as svm , naive bayes or random forest.

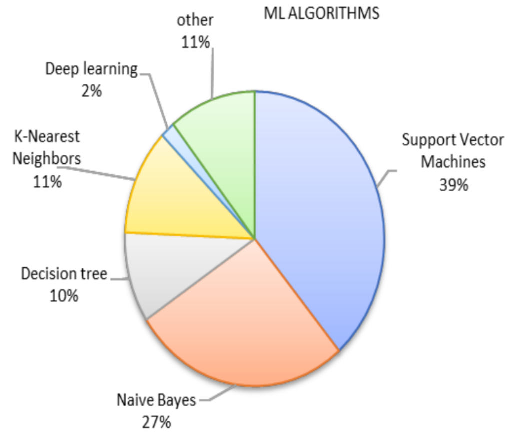

Figure 5: Machine learning models used in sentiment analysis [TAN17]

The system architecture is illustrated in the figure below:

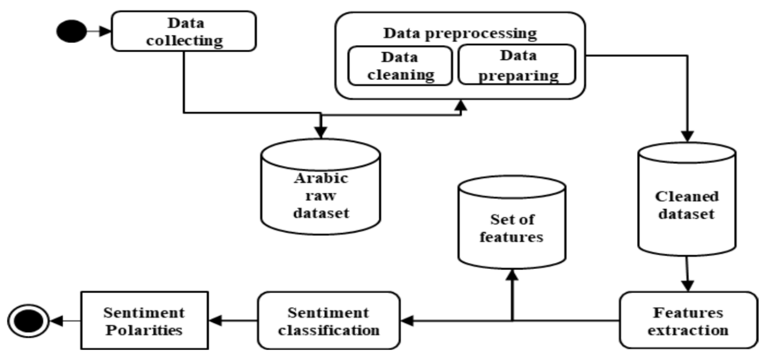

Figure 6: System Architecture [TAN17]

6 Feature Extraction

Feature Extraction is one of the trivial steps to be followed for a better understanding of the context of what we are dealing with.After the initial text has been cleaned up and normalised, it must be converted into characteristics that can be applied to modelling. Prior to modelling, we employ a specific way to give certain words in our document weights. Since numbers are simple for computers to process, we choose to represent words numerically. Consider a scenario in which a specific word appears in every document and does so repeatedly. Over time, this word will occur more frequently and have a higher value, which will make it appear in a sentence more frequently. A way to solve this issue is using TF-IDF.[M21]

6.1 TF-IDF

The idea of TF-IDF is to reflect the importance of a word to its document or sentence by normalizing the words which occur frequently in the collection of documents.[Jai22]

- **Term frequency (TF)**: Number of times a term has appeared in a document and is a measure of how frequently or how common a word is for a given sentence.
- **Inverse Document Frequency (IDF)**:The inverse document frequency (IDF) is a measure of how rare a word is in a document
- **TF-IDF**: Evaluates how relevant is a word to its sentence in a collection of sentences or documents

7 Sentiment Analysis Model

In our project, we focused on performing sentiment analysis on Arabic tweets. Sentiment analysis is a valuable technique that allows us to determine the sentiment or emotional tone behind a piece of text. To achieve this, we employed three machine learning models: Naive Bayes, Support Vector Machines (SVM), and Decision Trees. In order to use machine learning models there was two necessary steps . First of all, extracting sentiment labels from the cleaned dataset. Extracted sentiment labels were fed to the machine learning model, so the model knows whether it's prediction is correct or not in training and testing phases. The second step is extracting the cleaned tweets and feeding them to the machine learning model after extracting their features using TF-IDF. We used two data cleaning pipelines one with stemming and the other one without as sometimes stemming in arabic language sometimes misleads. In order to use machine learning models, we extracted sentiment label(we take it as an input to the model).

7.1 Support Vector Machines

Support Vector Machine (SVM) is a machine learning algorithm for supervised learning that can be applied to classification or regression problems. The majority of the time, it is utilised in classification issues like text categorization. In the SVM algorithm, each data point is represented as a point in n-dimensional space (where n is the number of features you have) with each feature's value being the value of a certain coordinate. Then, classification is performed by identifying the ideal hyper-plane that distinguishes the two classes. In our approach, we performed train-test split (training data = 80 percent). We used 3 metrics which are F1-score, Recall and precision. Learning rate (c =0.1) and kernel used was linear.[Ray23]

Metric	Stem	No-Stem
Accuracy	77.92	78.58
Precision	77.98	78.70
Recall	77.92	78.58
F1 Score	77.91	78.57

Table 1: Performance Metrics

7.2 Navie Bayes

The Naive Bayes classifier is a supervised machine learning algorithm, which is used for classification tasks, like text classification. It is also part of a family of generative learning algorithms, meaning that it seeks to model the distribution of inputs of a given class or category. Unlike discriminative classifiers, like logistic regression, it does not learn which features are most important to differentiate between classes. In our approach, we performed train-test split (training data = 80 percent). We used 3 metrics which are F1-score, Recall and precision.[Cha22]

Metric	Stem	No-Stem
Accuracy	76.4	77.0
Precision	77.0	77.0
Recall	76.5	77.0
F1 Score	76.5	77.0

Table 2: Performance Metrics

7.3 Decision Trees

Decision trees are widely used in sentiment analysis to classify textual data based on the sentiment expressed within the text. A decision tree is a hierarchical structure that consists of nodes representing decisions and branches representing the possible outcomes or classifications. In sentiment analysis, the decision tree algorithm analyzes features extracted from the text, such as word frequency, syntactic patterns, and contextual cues, to construct a tree-based model. Each decision node in the tree evaluates a specific feature and branches out to subsequent nodes based on the feature's value. Eventually, the leaves of the tree represent the predicted sentiment class, such as positive, negative, or neutral. Decision trees offer interpret-ability and transparency, enabling analysts to understand the reasoning behind the sentiment classification. They are particularly effective in sentiment analysis due to their ability to handle both categorical and numerical features and handle non-linear relationships between features and sentiment.[GDTD99]

Metric	Stem	No-Stem
Accuracy	77.0	78.0
Precision	77.3	77.5
Recall	77.3	77.4
F1 Score	77.3	77.4

Table 3: Performance Metrics

7.4 Fine Tuning

We performed grid search for fine tuning to obtain the best parameters.Grid search is a technique used in machine learning and data science to find the optimal hyperparameters for a given model. Hyperparameters are parameters that are not learned from the data but set by the user before training the model. They determine the behavior and performance of the model.

In grid search, you define a grid of possible hyperparameter values and then exhaustively search through all the combinations of these values. For each combination, you train and evaluate the model using a chosen evaluation metric, such as accuracy or F1 score. The combination of hyperparameter values that yields the best performance on the evaluation metric is selected as the optimal set of hyperparameters.For Naive Bayes, We defined a set of alphas(learning rate) in our gird search, so we can obtain the optimal one using grid search. The optimal one was 0.1.

Regarding Decision Trees, parameters were criterion, max depth and mini samples split. Optimal parameters were gini for the criterion, max depth None and 5 for mini samples split.

8 Sarcasm Detection and Sentiment Analysis

Sarcasm detection and sentiment analysis are two closely related tasks in natural language processing. While sentiment analysis aims to determine the sentiment expressed in a text, sarcasm detection focuses on identifying instances of sarcastic language. Sarcasm is a form of verbal irony where the intended meaning is often the opposite of the literal interpretation. It poses a challenge for sentiment analysis because the sentiment expressed in a sarcastic statement may not align with the actual sentiment conveyed. To address this, researchers have developed various techniques to incorporate sarcasm detection within sentiment analysis frameworks. These methods often involve feature engineering, such as examining the presence of sarcasm indicators like negation, irony markers, and exaggerated language. Machine learning algorithms, including decision trees, support vector machines, and neural networks, are commonly employed to learn patterns and distinguish between sarcastic and non-sarcastic statements. By combining sarcasm detection with sentiment analysis, models can provide more accurate and nuanced results, capturing the true sentiment behind sarcastic expressions. This integration has significant applications in areas such as social media analysis, customer feedback processing, and online review systems. In this part, we trained new models on both sentiment analysis and sarcasm detection. The model is evaluated by both predicting sentiment (positive,negative,neutral) and predicting whether it's sarcastic or not. In order to use machine learning models, we extracted sentiment label and sarcasm label to be taken as inputs for the machine learning models.

8.1 Dataset

ArSarcasm is a new Arabic sarcasm detection dataset. The dataset was created using previously available Arabic sentiment analysis datasets (SemEval 2017 and ASTD) and adds sarcasm and dialect labels to them. The dataset contains 10,547 tweets, 1,682 (16 percent) of which are sarcastic.

8.2 Feature Extraction

In our feature extraction process, we utilized a combined model incorporating both tf-idf and word embedding techniques. This approach enabled us to capture the unique characteristics of Arabic tweets. Tf-idf (term frequency-inverse document frequency) calculates the importance of a term in a document by considering its frequency within the document and across the entire corpus. Word embedding, on the other hand, is a representation of words in a dense vector space, capturing semantic relationships between words. Our combined model leveraged the strengths of both methods to effectively represent the content of Arabic tweets. By utilizing tf-idf and the pre-trained word embedding model tailored for Arabic language, we were able to extract meaningful features that captured the specific nuances and contextual information of Arabic tweets, enhancing the performance of our machine learning models.Fine-tuning was also performed on naive bayes and decision trees.

8.3 Support Vector Machines

Support Vector Machines (SVM) have proven to be effective in sarcasm detection tasks. SVM is a machine learning algorithm that aims to find an optimal hyperplane in a high-dimensional feature space to separate different classes of data points. In the context of sarcasm detection, SVM can be trained to distinguish between sarcastic and non-sarcastic statements by learning patterns and relationships between various linguistic features.[Ray23]

	Precision	Recall	F1-Score
Negative False	0.45	0.31	0.37
Negative True	0.48	0.25	0.33
Neutral False	0.60	0.88	0.71
Neutral True	1.00	0.03	0.06
Positive False	0.60	0.28	0.39
Positive True	0.00	0.00	0.00

Table 4: Classification Table

Metric	Value
Accuracy	0.57
Precision	0.55
Recall	0.57
F1-score	0.52

Table 5: Performance Metrics

8.4 Naive Bayes

Naive Bayes can also be applied in sarcasm detection tasks. Sarcasm detection involves determining whether a given statement or sentence is sarcastic or not. Naive Bayes can be used as a classifier to predict the presence or absence of sarcasm based on the input text.[Cha22]

Metric	Value
Accuracy	0.56
Precision	0.54
Recall	0.56
F1-score	0.50

Table 6: Performance Metrics

	Precision	Recall	F1-Score
Negative False	0.48	0.26	0.33
Negative True	0.49	0.23	0.31
Neutral False	0.58	0.91	0.71
Neutral True	0.50	0.03	0.06
Positive False	0.57	0.22	0.32
Positive True	0.00	0.00	0.00

Table 7: Classification Table

8.5 Decision Trees

Decision trees have proven to be effective in sarcasm detection tasks. Sarcasm is a complex linguistic phenomenon that involves the use of words or phrases to convey a meaning that is often the opposite of the literal interpretation. Detecting sarcasm in text can be challenging due to the subtlety and context-dependent nature of sarcastic expressions. Decision trees offer a powerful approach to tackle sarcasm detection. These models learn a hierarchy of if-else conditions based on the features extracted from the input text. By analyzing the patterns and relationships between different words and phrases, decision trees can capture the nuanced cues that indicate sarcasm. During the training phase, the decision tree algorithm recursively partitions the data based on the feature values. At each step, it selects the most informative feature that best separates the sarcastic and non-sarcastic instances. The splitting process continues until a stopping criterion is met, such as reaching a maximum tree depth or achieving a minimum number of instances in each leaf node.[GDTD99]

	Precision	Recall	F1-Score
Negative False	0.34	0.15	0.20
Negative True	0.37	0.22	0.27
Neutral False	0.56	0.89	0.69
Neutral True	0.00	0.00	0.00
Positive False	0.42	0.14	0.21
Positive True	0.00	0.00	0.00

Table 8: Classification Table

Metric	Value
Accuracy	0.51
Precision	0.45
Recall	0.51
F1-score	0.44

Table 9: Performance Metrics

9 Challanges

Arabic sentiment analysis and sarcasm detection pose specific challenges due to the unique characteristics of the Arabic language. In this section, we will discuss the challenges encountered during the project and their impact on the overall analysis.

9.1 Time-consuming fine-tuning of the SVM model

One major challenge faced was the time required for fine-tuning the Support Vector Machine (SVM) model. Fine-tuning involves adjusting model hyperparameters and optimizing the model's performance on the target task. Due to the complexity of the Arabic language and the size of the dataset, the fine-tuning process took significantly longer than anticipated. This increased the overall time for developing an accurate sentiment analysis and sarcasm detection model.

9.2 Limited availability of sarcasm detection dataset

Sarcasm detection relies heavily on annotated data for training and evaluation. Unfortunately, we encountered a significant challenge in sourcing a comprehensive and large-scale sarcasm detection dataset specifically for Arabic text. The scarcity of available labeled data restricted the model's exposure to diverse sarcasm patterns, leading to limited performance in accurately detecting sarcasm in Arabic text.

9.3 Impact of Arabic dialects on the model

The Arabic language encompasses a range of dialects, each with its own vocabulary, and expressions. These dialectical variations can significantly impact sentiment analysis and sarcasm detection models trained on standard Arabic. The lack of dialect-specific training data caused difficulties in capturing the subtleties associated with sentiment and sarcasm in different Arabic dialects.

10 Conclusion

Overall, Arabic sentiment analysis and sarcasm detection present distinct challenges that need to be addressed to achieve accurate results. The time-consuming nature of fine-tuning, limited availability of annotated sarcasm detection datasets, and the impact of Arabic dialects on the model's performance were among the primary challenges encountered during the project.Finally, Results shows that svm was the best model regarding sentiment analysis and sarcasm detection.

References

[AR23] Arwa Alqarni and Atta Rahman. Arabic tweets-based sentiment analysis to investigate the impact of covid-19 in ksa: A deep learning approach. *Big Data and Cognitive Computing*, 7(1):16, 2023.

[Cha22] Koushiki Dasgupta Chaudhuri. Building naive bayes classifier from scratch to perform sentiment analysis, Mar 2022.

[GDTD99] Orphanos Giorgos, Kalles Dimitris, Papagelis Thanasis, and Christodoulakis Dimitris. Decision trees and nlp: A case study in pos tagging. In *Proceedings of annual conference on artificial intelligence (ACAI)*, 1999.

[Jai22] Abhishek Jaiswal. Nlp tutorials part ii: Feature extraction, Nov 2022.

[M21] Siddharth M. Feature extraction and embeddings in nlp: A beginners guide to understand natural language processing, Jul 2021.

[Ray23] Sunil Ray. Learn how to use support vector machines (svm) for data science, Apr 2023.

[TAN17] Samir Tartir and Ibrahim Abdul-Nabi. Semantic sentiment analysis in arabic social media. *Journal of King Saud University-Computer and Information Sciences*, 29(2):229–233, 2017.