Assessing the Performance of ChatGPT-4, Fine-tuned BERT and Traditional ML Models on Moroccan Arabic Sentiment Analysis

Mohamed Hannani
University of Siegen, Germany
mohamed_hannani@yahoo.com

Abdelhadi Soudi
Ecole Nationale Supérieure des Mines de Rabat, Morocco
asoudi@enim.ac.ma

Kristof Van Laerhoven
University of Siegen, Germany
kvl@eti.uni-siegen.de

Abstract

Large Language Models (LLMs) have demonstrated impressive capabilities in various natural language processing tasks across different languages. However, their performance in low-resource languages and dialects, such as Moroccan Arabic (MA), requires further investigation. This study evaluates the performance of ChatGPT-4, different fine-tuned BERT models, FastText as text representation, and traditional machine learning models on MA sentiment analysis. Experiments were done on two open source MA datasets: an X(Twitter) Moroccan Arabic corpus (MAC) and a Moroccan Arabic YouTube corpus (MYC) datasets to assess their capabilities on sentiment text classification. We compare the performance of fully fine-tuned and pre-trained Arabic BERT-based models with ChatGPT-4 in zero-shot settings.

Keywords: Sentiment Analysis, Bert, GPT, Moroccan Arabic, LangChain

1 Introduction

The field of sentiment analysis (SA) has made remarkable advances, enabling the extraction and analysis of human sentiments from textual data for a variety of purposes. This technology has diverse applications, ranging from social media monitoring and market research to political discourse analysis.

However, SA faces several challenges, such as the phenomena of data and concept drift (Zhao et al., 2022), particularly pronounced in the ever-evolving landscape of social media. Data drift refers to changes in the statistical properties of the input data over time, while concept drift indicates a deeper shift in the underlying meaning or interpretation of the data that the model aims to predict. These drifts can lead to a decrease in the accuracy of SA models if not addressed. For instance, the way people express sentiments online can evolve rapidly, making previously trained models less effective. This necessitates ongoing monitoring and adaptation of SA methodologies, including updating rule-based systems and dictionaries, as well as retraining machine learning models to ensure they remain aligned with the shifting linguistic and cultural contexts of web-based communication.

While SA has seen substantial progress in major languages, its application to dialectal languages, such as MA, a regional variant of Modern Standard Arabic (MSA), has not received much attention. MA is the main medium of communication among Moroccans. The unique linguistic features of MA, including regional variations, colloquialism, borrowed words from other languages, coupled with the use of multiple scripts (Arabic and Latin), present significant challenges for SA. Previous SA research (Elmadany et al., 2022) has predominantly focused on corpora written in Arabic script despite the increasing prevalence of Latin script usage in online communication, particularly on social media platforms.

Recent advancements in AI, particularly the emergence of LLMs, such as GPT-4 (OpenAI, 2023), PaLM 2 (Anil et al., 2023) and Falcon (Penedo et al., 2023) offer potential solutions to the challenges posed by data and concept drift in SA. These models, trained on vast and diverse datasets and fine-tuned for various tasks, have demonstrated promising capabilities in SA (Wang et al., 2023; Inoue et al., 2021; Amin et al., 2023). While some research has explored the potential of LLMs for Standard Arabic sentiment analysis (Al-Thubaity et al., 2023), no study has so far evaluated the performance of LLMs on MA. This work conducts the first-ever evaluation of ChatGPT's performance on MA SA, offering valuable insights into the applicability of LLMs in analyzing sentiment in Arabic dialects.

Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities, pages 489–498
November 16, 2024. ©2024 Association for Computational Linguistics

2 Related Work

The availability of sentiment data from social media platforms has greatly increased interest in Arabic sentiment analysis (SA) research over the last ten years. Speakers of Arabic dialects were historically limited to using their dialects only when speaking. However, the emergence of social media has given Arabic speakers the ability and space to express themselves in writing as well (Darwish et al., 2021). This has resulted in an abundance of informal, dialectal textual material, as opposed to MSA formality. A multitude of datasets spanning multiple genres—mostly tweets—have been created for Arabic SA, including Egyptian (Nabil et al., 2015; Refaee and Rieser, 2014), Levantine (Baly et al., 2019), Maghrebi (Mdhaffar et al., 2017), as well as the Saudi dialect (Assiri et al., 2016). Other datasets (Al-Obaidi and Samawi, 2016; Abdul-Mageed et al., 2014) include several Arabic dialects in addition to MSA.

Arabic Sentiment Analysis has traditionally concentrated on rule-based techniques, much like other languages (ElSahar and El-Beltagy, 2014; Al-Twairesh et al., 2016). The main goal of these techniques was to create sentiment lexicons. Arabic Sentiment Analysis has seen a rise in interest in applying machine learning techniques in recent years. These techniques are less vulnerable to the drawbacks of lexicon-based techniques and are capable of identifying sentiment patterns from a big corpus of text. To implement morphological and syntactic features, popular machine learning techniques have been employed, such as Naïve Bayes (NB), Support Vector Machines (SVMs), and K-Nearest Neighbor (kNN) classifiers (Abdul-Mageed et al., 2014; Duwairi and Qarqaz, 2014; Abdulla et al., 2013).

Transformer-based models, such as BERT (Devlin et al., 2018), ALBERT (Lan et al., 2019), XL-Net (Yang et al., 2019), and RoBERTa (Liu et al., 2019), have been introduced and proved successful in several natural language processing (NLP) applications. BERT and BERT-like models achieved state-of-the-art performance on many NLP tasks, including sentiment analysis in many languages (Sun et al., 2019).

Abdul-Mageed et al. (2020); Antoun et al. developed two models, ARBERT and MARBERT, pretrained on a large collection of datasets in MSA and several Arabic dialects (Levantine, Moroccan Arabic, etc.). They reported new state-of-the-art results on the majority of the datasets in their fine-tuning benchmark.

In addition to discriminative models, such as BERT, generative models have recently gained prominence in NLP research. These models, such as GPT (Radford et al., 2018; Brown et al., 2020), T5 (Raffel et al., 2020), and BLOOM (Scao et al., 2022), are designed to create new text samples. Multilingual and language-specific versions of these models have been developed. For example, AraT5 (Elmadany et al., 2022) and AraGPT-2 (Antoun et al.) are tailored for Arabic. Generative models have demonstrated potential in tasks, such as text completion, translation, summarization, and even sentiment analysis, where they can generate text that aligns with specific sentiments (Al-Thubaity et al., 2023).

In this work, we evaluate the performance of ChatGPT-4 and transformer-based models on SA of MA using the aforementioned open source datasets, namely MAC (Garouani and Kharroubi, 2021) and MYC (Jbel et al., 2023). To our knowledge, this is the first-ever attempt to compare the performance of these models on the MA SA task.

The structure of the rest of the paper is as follows: Section 3 presents the experimental setup and the various experiments we conducted, datasets, and model architectures. Section 4 presents the experiments' results and analysis. Section 5 provides concluding remarks and future work.

3 Experimental Setup

Our research objective is to evaluate the capability of the ChatGPT model, some existing pre-trained BERT models, and FastText (Joulin et al., 2017) as sentiment analyzers for MA. To assess this, we utilize the two datasets, MAC and MYC, designed for sentiment analysis. On each dataset, we evaluate ChatGPT, fine-tuned BERT-based, and FastText models. Furthermore, we compared the fine-tuned and pre-trained BERT-based models with the ChatGPT results. The results of our experiments are compared with other related work.

We use `gpt-4-turbo` model by OpenAI [1] for both MAC and MYC datasets as a sentiment analyzer for MA. We asked the model to predict the class of the given input tokens. Table 1 summarizes the parameters and the prompt used when calling the model.

The primary objective of our experiments is to

[1] https://openai.com/

Parameters	temperature	top_p
Values	0.3	1

Table 1: GPT-4-turbo parameters with OpenAI API

assess the capabilities of generative models and BERT-based models, as well as FastText for MA sentiment analysis. We evaluate the following models:

- GPT-4, accessed via ChatGPT by OpenAI,
- Pre-trained/fine-tuned BERT-based models,
- FastText as text representation.
- Traditional machine learning classifiers

For GPT-4, we utilize the `ChatOpenAI` wrapper provided by LangChain framework [2] to send prompts and receive responses. For BERT-based models, we fine-tune (full network or freezing the model's backbone) various existing models pre-trained on a large corpus in a variety of languages. Table 2 provides information on the BERT-based models we used for our experiments.

3.1 Prompt Composition

The system prompts used for calling the GPT model for the MAC and MYC datasets are presented below.

MAC Dataset Prompt

```
<Predict the class of this Arabic
    review (e.g ternary
    classification), whether it's
    positive (return 2), neutral
    (return 1) or negative (return 0)
    review. Please do not return
    anything other than that.>
```

MYC Dataset Prompt

```
<Predict the class of this Arabic
    review (e.g binary classification),
    whether it's positive (return 2) or
    negative (return 0) review. Please
    do not return anything other than
    that.>
```

To facilitate prompt composition and enhance sentiment detection, we integrated LangChain into our system. LangChain serves as a framework designed for the development of applications leveraging LLMs. Its primary objective is to empower developers with the seamless integration of diverse data sources and the facilitation of interactions with other applications. To achieve this goal, LangChain framework offers modular components, serving as abstractions, and customizable use of case-specific pipelines, referred to as chains. We also used a json parser as part of the Chain-of-Thoughts to ensure getting exactly and only the class label when invoking the model API. The prompt template is shown in Figure 1.

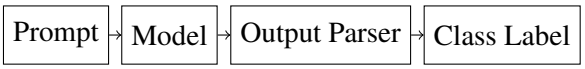

Figure 1: Chain-of-Thoughts used for Sentiment Analysis in Moroccan Arabic Dialect

The `Prompt` is the `SystemMessage` component followed by `HumanMessage` from Langchain framework. The `Prompt` is then used to request the API, and the API response is then sent to the `JsonOutputToolsParser` provided by the same framework to parse the response for consistency.

3.2 Sentiment Datasets

For the aforementioned experiments, we use two datasets: the MAC dataset, an MA corpus consisting of 18000 manually labeled tweets, resulting in a lexicon-dictionary of 30000 words labeled as positive, negative, and neutral. Table 3 below shows information about the pre-processed MAC dataset. We had to remove tweets that have the class `mixed` (MSA and MA) and missing values of the type column (when not labeled).

The complexity of Moroccan web content features a blend of Arabic and Latin script. This dual-script usage in MA adds a layer of complexity that traditional sentiment analysis approaches might overlook. To evaluate our models on that type of MA, we used the MYC dataset (Jbel et al., 2023), which contains 20k (raw data) comments scrapped from 50 Moroccan famous YouTube channels on different topics. Table 3 below showcases statistics about the pre-processed MYC dataset.

It is worth mentioning that the pre-processed MYC shared by Jbel et al. (2023) is not really pre-processed as they claimed in their paper. We tried to follow their pre-processing steps in their paper, namely, remove empty comments, remove usernames, remove links (https and http links), and remove unlabeled samples.

[2] https://www.langchain.com/

Model Name	Pre-training Language	Vocabulary Size
bert-base-multilingual-cased	Multilingual	119547
bert-base-arabic	Arabic	32000
darijabert-arabizi	Arabic	110000
DarijaBERT	Moroccan Arabic Dialect (Darija)	80000
bert-base-arabertv2	Arabic	64000

Table 2: BERT-Based Models and Pre-training Languages

Dataset	Size	Tweet Class			Arabic Type	
		Positive	Negative	Neutral	Modern Standard Arabic	Moroccan Arabic
MAC	18k	9888	3505	4039	12145	5287
MYC	16k	7427	8621	-	-	-

Table 3: The pre-processed MAC and MYC Dataset statistics

3.3 Models Architecture And Setup

For our experiments with BERT-based and FastText models, we employed a custom classifier head, as illustrated in Figure 2 which consists of a sequential architecture incorporating linear transformations, `ReLU` activation, `Dropout` for regularization, and a final `Softmax` layer for classification. This classifier head was integrated with powerful pre-trained language models like BERT, known for its contextual understanding capabilities. Specifically, the BERT model's output from the pooler was fed into our classifier, allowing us to leverage BERT's deep semantic representations. Additionally, we incorporated fastText embeddings, renowned for their efficiency in handling morphologically rich languages like Arabic.

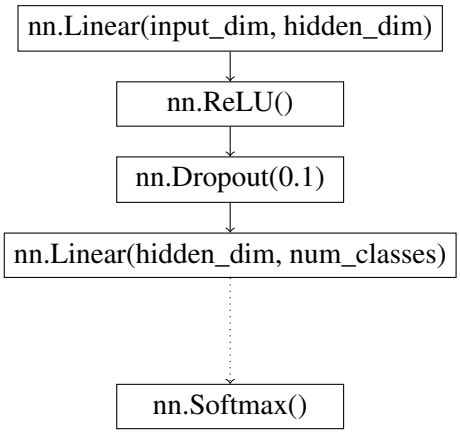

Figure 2: Classifier head used for BERT-based models and FastText for text representation for SA.

For the BERT-based models shown in Table 2, we used the BERT backbone (freezed or trained from scratch), the pooler output is fed to the classifier shown in Figure 2, and the Table 4 shows the parameter values of the classifier head when used with BERT-based models.

input_dim	hidden_dim	num_classes
768	512	3(MAC), 2(MYC)

Table 4: classifier head's parameters with BERT-based models

For the FastText model, we used `facebook/fasttext-ar-vectors` (Joulin et al., 2016). FastText is a library for efficient learning of word representations and sentence classification. Table 5 shows the parameter values of the classifier head when used with FastText model.

input_dim	hidden_dim	num_classes
300	128	3(MAC), 2(MYC)

Table 5: classifier head's parameters with FastText model

For the ChatGPT model, we evaluated the GPT-4-Turbo model on the pre-processed $\textbf{MAC}_{full}$ and $\textbf{MYC}_{full}$ with the parameters shown previously in Table 1 and prompt as discussed in Section 3.1 for both datasets to see their performance on different datasets with different types and sources (Twitter and YouTube).

For BERT-based models, we trained all the mentioned pre-trained BERT models shown in Table 2, We used stratified sampling to ensure a balanced class distribution across test and train sets. We used $\textbf{MAC}_{80\%}$ for training and $\textbf{MAC}_{20\%}$ for evaluation. The same approach was applied to the MYC dataset (ie., $\textbf{MYC}_{80\%}$ for training and $\textbf{MYC}_{20\%}$ for testing).

For the FastText model, we used $\textbf{MAC}_{80\%}$, $\textbf{MYC}_{80\%}$ for training, and $\textbf{MAC}_{20\%}$, $\textbf{MYC}_{20\%}$ for testing.

In the following section 4, we present the results of the experiments, compare, and discuss the evaluation results for each model and dataset.

4 Results And Analysis

4.1 GPT-4 model via OpenAI

As discussed previously in Section 3.1 and 3.3, we evaluated the `gpt-4-turbo` model on $\textbf{MAC}_{full}$ and $\textbf{MYC}_{full}$ datasets, as well as on test subsets, $\textbf{MAC}_{20\%}$ and $\textbf{MYC}_{20\%}$ for comparison purposes with other models.

Since the MAC dataset contains tweets in MSA and MA, we also aimed to evaluate the performance of the model in each class. The following Table 6 summarizes the results of the evaluation in $\textbf{MAC}_{full}$ and $\textbf{MAC}_{test}$ with 20% of the dataset.

Accuracy	Precision	Recall	F1-score
$\textbf{MAC}_{full}$			
Modern Standard Arabic			
0.710	0.741	0.710	0.713
Moroccan Arabic			
0.690	0.714	0.690	0.688
$\textbf{MAC}_{20\%}$			
Modern Standard Arabic			
0.635	0.720	0.635	0.654
Moroccan Arabic			
0.607	0.673	0.607	0.620

Table 6: GPT-4 model performance in $\textbf{MAC}_{full}$ and $\textbf{MAC}_{20\%}$ across the type of class (MSA or MA).

As can be seen in Table 6, for both full and test sets, we notice that the GPT-4 model performs well in tweets written in MSA, compared to those written in (MA). This difference in performance can be attributed to several factors. Firstly, language uniformity plays a significant role. MSA is a standardized and formal version of Arabic used in official communication, media, literature, and formal speeches. It has consistent grammar, vocabulary, and syntax, which makes it easier for NLP models to learn and predict accurately. In contrast, MA varies significantly across regions and often incorporates local slang, colloquialisms, and foreign words. This linguistic diversity and lack of standardization make it challenging for models to perform consistently. Secondly, the availability and quality of training data influence model performance. Models, such as GPT-4 are often trained on large corpora that include a substantial amount of MSA texts, given its prevalence in written and formal contexts. This extensive training on MSA helps the model learn its patterns more effectively. On the other hand, there is generally less training data available for dialectical variants due to their informal use and the vast regional differences. This scarcity of training data can lead to poorer model performance on dialectical texts.

In the $\textbf{MAC}_{20\%}$ subset, the metrics for the Standard Arabic classifier show an accuracy of 0.6356, precision of 0.7205, recall of 0.6356, and F1 score of 0.6545.

To understand the challenges the model faces in classifying tweets, we examined the performance across different sentiment classes: positive, neutral, and negative. Table 7 shows the scores across each class (positive, negative, and neutral) on $\textbf{MAC}_{20\%}$ set.

Class	Precision	Recall	F1 Score
Modern Standard Arabic			
Negative	0.80	0.85	0.82
Neutral	0.35	0.61	0.44
Positive	0.84	0.57	0.68
Moroccan Arabic			
Negative	0.55	0.81	0.66
Neutral	0.38	0.50	0.43
Positive	0.85	0.58	0.69

Table 7: Performance metrics for Modern Standard Arabic and Moroccan Arabic tweets on $\textbf{MAC}_{20\%}$ set.

The classification reports 7 offer detailed insights into the performance of the model, providing metrics such as precision, recall, and F1-score for each sentiment class. With respect to MSA, the model demonstrates strong performance in identifying negative tweets, achieving high precision (0.80) and recall (0.85). However, it struggles with neutral tweets, as evidenced by the lower precision (0.35) and recall (0.61), indicating difficulty in distinguishing neutral sentiment. Similarly, while the model exhibits high precision (0.84) in classifying positive tweets, the lower recall (0.57) suggests that some positive tweets are misclassified as neutral or negative.

In the case of MA, the model achieves moderate precision (0.55) and recall (0.81) for negative tweets, indicating reasonable performance in this

class. However, the precision (0.38) and recall (0.50) for neutral tweets are significantly lower, highlighting challenges in accurately predicting neutral sentiment. Despite maintaining high precision (0.85) for positive tweets, similar to MSA, the model struggles with recall (0.58), indicating misclassification issues.

A key observation from these reports is the consistent difficulty the model encounters with neutral tweets across both MSA and MA. Lower precision and recall scores suggest that neutral tweets are often misclassified as either positive or negative, indicating a need for improved classification strategies for neutral sentiment. Additionally, while the model generally performs better on negative and positive classes, the lower recall for positive tweets suggests a tendency to miss some positive instances, possibly predicting them as neutral or negative.

In the remainder of this section, we evaluate the performance of GPT-4 on the MYC dataset, purely dialectal data, Unlike the MAC dataset, MYC includes both Arabic and Latin script, as discussed previously in Section 3.2

The following Table 8 summarizes the results of the evaluation on the full MYC dataset.

Accuracy	Precision	Recall	F1-score
MYC$_{full}$			
0.624	0.623	0.622	0.622
MYC$_{20\%}$			
0.608	0.6087	0.608	0.607

Table 8: GPT-4 model performance in **MYC**$_{full}$ and **MYC**$_{20\%}$

To understand the challenges encountered by the model in categorizing tweets, it is essential to analyze its performance across various sentiment categories: positive and negative. Table 9 presents the performance metrics for each sentiment class (positive and negative) on the **MYC**$_{20\%}$ dataset. The performance metrics for sentiment classification on the **MYC**$_{20\%}$ dataset, as shown in Table 9, highlight varying degrees of success in accurately classifying tweets into positive and negative sentiment categories. The model achieves a precision of 0.647 and a recall of 0.655 for negative tweets, indicating a relatively balanced ability to correctly identify negative sentiment instances while minimizing false negatives. Conversely, for positive tweets, the precision is 0.597, indicating that a significant portion of the positively classified tweets may be incorrect, while the recall is 0.588, suggesting a lower ability to capture all positive instances present in the **MYC**$_{full}$ dataset.

Class	Precision	Recall	F1 Score
Negative	0.647	0.655	0.651
Positive	0.597	0.588	0.593

Table 9: Performance on **MYC**$_{full}$ across each class.

4.2 Fine-tuned BERT-based models

In this section, we explore the performance of various BERT-based models presented previously in Table 2 trained and evaluated on MAC and MYC datasets and configurations. More specifically, we conducted experiments with the following configurations:

1. BERT-based models trained on **MAC**$_{80\%}$ and evaluated on **MAC**$_{20\%}$ dataset. This model was trained with two options: (a) training the entire network, and (b) freezing the backbone and training only the classifier.

2. BERT-based models were fully trained on **MAC**$_{80\%}$ and evaluated on **MYC**$_{20\%}$ dataset.

3. BERT-based models were fully trained on **MYC**$_{80\%}$ and evaluated on **MYC**$_{20\%}$ dataset.

The performance metrics of the BERT-based models trained and evaluated on the MAC dataset, as shown in Table 10, highlight significant differences between models trained with fully unfrozen networks and those with frozen backbones. When the entire network is trained, models such as `DarijaBERT` and `bert-base-arabertv2` demonstrate superior performance, with `DarijaBERT` achieving the highest accuracy of 0.90, precision of 0.881, and F1-score of 0.877. This indicates a robust capability to capture the nuances of the MAC dataset. Conversely, models trained with frozen backbones exhibit notably lower performance, with the `bert-base-multilingual-cased` model showing the lowest accuracy (0.602) and F1-score (0.353).

To further evaluate the generalization capabilities of our BERT-based models, we conducted experiments where the models were fully trained on **MAC**$_{80\%}$ and evaluated on **MYC**$_{20\%}$ dataset. This approach allows us to assess how well the models, trained on Twitter data (MAC), perform when

Accuracy	Precision	Recall	F1-score
Experiment 1.a.			
bert-base-multilingual-cased			
0.857	0.821	0.824	0.822
bert-base-arabic			
0.888	0.868	0.861	0.864
darijabert-arabizi			
0.872	0.844	0.834	0.838
DarijaBERT			
0.90	**0.881**	0.873	**0.877**
bert-base-arabertv2			
0.896	0.870	**0.874**	0.872
Experiment 1.b.			
bert-base-multilingual-cased			
0.602	0.389	0.396	0.353
bert-base-arabic			
0.661	0.647	0.503	0.520
darijabert-arabizi			
0.662	0.607	0.529	0.545
DarijaBERT			
0.694	**0.646**	**0.579**	**0.598**
bert-base-arabertv2			
0.687	0.639	0.578	0.596

Table 10: Fully trained and frozen backbone BERT-based models on $\mathbf{MAC}_{80\%}$ and evaluated on $\mathbf{MAC}_{20\%}$.

applied to a different source, namely YouTube comments (MYC), thereby testing their robustness and adaptability across diverse text sources.

Table 11 presents the results of the BERT-based models that were fully trained on $\mathbf{MAC}_{80\%}$ and evaluated on the $\mathbf{MYC}_{20\%}$ dataset to assess their cross-domain performance. The accuracy ranges from 0.560 to 0.619, with DarijaBERT achieving the highest accuracy and F1-score, indicating its superior generalization capability. bert-base-arabic shows the highest precision, suggesting effectiveness in predicting positive instances, though it, like other models, struggles with recall. The observed drop in performance across models underscores the challenges of transferring knowledge between datasets from different platforms (Twitter vs. YouTube), highlighting the need for further fine-tuning and more diverse training data to enhance cross-platform generalization.

The evaluation of the BERT-based models when fully trained and evaluated on the same dataset (MYC) showed better performance. We used the $\mathbf{MYC}_{80\%}$ subset for training and $\mathbf{MYC}_{20\%}$ subset for evaluation. As can be seen in Table 12, the accuracy, precision, recall, and F1-score are notably higher compared to the cross-dataset evaluation, indicating that the models perform better when trained and evaluated within the same context. darijabert-arabizi achieved the highest performance with an accuracy and F1-score of 0.856, suggesting its strong capability in handling the nuances of the MYC dataset. These findings emphasize the importance of dataset domain alignment in training and evaluating machine learning models.

Accuracy	Precision	Recall	F1-score
Experiment 2.			
bert-base-multilingual-cased			
0.560	0.594	0.576	0.544
bert-base-arabic			
0.581	0.657	0.602	0.550
darijabert-arabizi			
0.583	0.624	0.599	0.567
DarijaBERT			
0.619	**0.681**	**0.637**	**0.601**
bert-base-arabertv2			
0.600	0.639	0.615	0.587

Table 11: Evaluation metrics of fully trained BERT-based models on $\mathbf{MAC}_{80\%}$ and evaluated on $\mathbf{MYC}_{20\%}$.

Accuracy	Precision	Recall	F1-score
Experiment 3.			
bert-base-multilingual-cased			
0.832	0.831	0.832	0.831
bert-base-arabic			
0.831	0.831	0.833	0.830
darijabert-arabizi			
0.856	**0.856**	**0.856**	**0.856**
DarijaBERT			
0.850	0.849	0.851	0.850
bert-base-arabertv2			
0.837	0.840	0.841	0.837

Table 12: Evaluation metrics of fully trained BERT-based models on $\mathbf{MYC}_{80\%}$ evaluated on $\mathbf{MYC}_{20\%}$.

4.3 FastText as Text Representation

We have also trained FastText-based model on the same training sets as in the previous experiments, using the text representation (embeddings) this time for the classifier as discussed in Section 3.3. Table 13 shows the evaluation results obtained on $\mathbf{MAC}_{20\%}$ and $\mathbf{MYC}_{20\%}$ which demonstrate notable differences between the performance between

the two models.

Accuracy	Precision	Recall	F1-score
Trained on $MAC_{80\%}$, evaluated on $MAC_{20\%}$			
0.837	0.790	0.814	0.801
Trained on $MYC_{80\%}$, evaluated on $MYC_{20\%}$			
0.790	0.525	0.526	0.528

Table 13: Classifier with FastText embeddings.

4.4 Traditional ML Classifiers

To compare the effectiveness of traditional machine learning methods against the previously discussed BERT-based models, GPT-4 and FastText embeddings, we conducted experiments with the same settings for training and evaluation. Figure 14 shows the first 3 best models for each experiment.

Accuracy	Precision	Recall	F1-score
On MAC			
Naive Bayes			
0.7239	**0.8092**	**0.7239**	**0.6868**
Quadratic Discriminant Analysis			
0.7238	0.8091	0.7238	0.6867
SVM - Linear Kernel			
0.7200	0.8033	0.7200	0.6806
On MYC			
Extreme Gradient Boosting			
0.5658	**0.9412**	**0.0658**	**0.1228**
Decision Tree Classifier			
0.5650	0.9538	0.0631	0.1183
SVM - Linear Kernel			
0.5650	0.9538	0.0631	0.1183

Table 14: Traditional classifiers performance.

The low recall values in MYC dataset indicate that the traditional classifiers have difficulty in identifying all instances of the positive class. In other words, they tend to miss a significant number of positive samples. Potential reasons for the low recall could include differences in data distribution, domain-specific characteristics, or noise introduced during data collection. Additionally, the language or dialectal variations present in MYC data, distinct from those in MAC, might pose challenges for classifiers in accurately identifying positive instances.

Prior work by Jbel et al. (2023) laid the groundwork for sentiment analysis on the MYC dataset by creating the dataset and evaluating a range of traditional and neural network models. They reported that the best performance was achieved with CNN model with an accuracy of 92.4. However, there are two main issues with this work. First, the pre-processed version of the dataset shared does not reflect the pre-processing steps they mentioned in their work. Second, the configuration of the training and the data size used for training and evaluation are not specified. Accordingly, it is difficult to fairly compare our results with theirs.

5 Conclusion And Future work

This work examined sentiment analysis on MAC and MYC datasets. We gained insights into the performance of different models and architectures in capturing sentiment nuances present in MA in different contexts and in both Arabic and Latin script. Although fine-tuned models performed well, the results obtained with ChatGPT show the latter's great potential for SA. The results have also shown that the performance of all these models on MA is less than that on MSA. This difference in performance can be attributed to several factors, such as language uniformity, and consistency in MSA grammar and vocabulary, which makes it easier for models to learn and predict accurately. On the other hand, MA varies across regions. Lack of standardization of MA makes it challenging for models to perform consistently. Additionally, the availability of data influence model performance. Models, such as GPT-4 are trained on large corpora that include a substantial amount of MSA texts, given its prevalence in written and formal contexts.

Future work requires the creation of large MA datasets and the development of new strategies to deal with the inconsistency in the MA data. Another research direction would be to leverage the complementary nature of FastText and BERT embeddings by employing an attention mechanism to combine them effectively. By integrating the context-aware representations from BERT with the morphological and semantic information captured by FastText embeddings.

References

Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. 2014. Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. Arbert & marbert: Deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.

Nawaf A Abdulla, Nizar A Ahmed, Mohammed A Shehab, and Mahmoud Al-Ayyoub. 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)*, pages 1–6. IEEE.

Ahmed Y Al-Obaidi and Venus W Samawi. 2016. Opinion mining: Analysis of comments written in arabic colloquial. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1.

Abdulmohsen Al-Thubaity, Sakhar Alkhereyf, Hanan Murayshid, Nouf Alshalawi, Maha Omirah, Raghad Alateeq, Rawabi Almutairi, Razan Alsuwailem, Manal Alhassoun, and Imaan Alkhanen. 2023. Evaluating ChatGPT and bard AI on Arabic sentiment analysis. In *Proceedings of ArabicNLP 2023*, pages 335–349, Singapore (Hybrid). Association for Computational Linguistics.

Nora Al-Twairesh, Hend Al-Khalifa, and AbdulMalik Al-Salman. 2016. Arasenti: Large-scale twitter-specific arabic sentiment lexicons. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 697–705.

Mostafa M Amin, Erik Cambria, and Björn W Schuller. 2023. Will affective computing emerge from foundation models and general artificial intelligence? a first evaluation of chatgpt. *IEEE Intelligent Systems*, 38(2):15–23.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Adel Assiri, Ahmed Emam, and Hmood Al-Dossari. 2016. Saudi twitter corpus for sentiment analysis. *International Journal of Computer and Information Engineering*, 10(2):272–275.

Ramy Baly, Alaa Khaddaj, Hazem Hajj, Wassim El-Hajj, and Khaled Bashir Shaban. 2019. Arsentd-lev: A multi-topic corpus for target-based sentiment analysis in arabic levantine tweets. *arXiv preprint arXiv:1906.01830*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R El-Beltagy, Wassim El-Hajj, et al. 2021. A panoramic survey of natural language processing in the arab world. *Communications of the ACM*, 64(4):72–81.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rehab M Duwairi and Islam Qarqaz. 2014. Arabic sentiment analysis using supervised classification. In *2014 International Conference on Future Internet of Things and Cloud*, pages 579–583. IEEE.

AbdelRahim Elmadany, Muhammad Abdul-Mageed, et al. 2022. Arat5: Text-to-text transformers for arabic language generation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long papers)*, pages 628–647.

Hady ElSahar and Samhaa R El-Beltagy. 2014. A fully automated approach for arabic slang lexicon extraction from microblogs. In *International conference on intelligent text processing and computational linguistics*, pages 79–91. Springer.

Moncef Garouani and Jamal Kharroubi. 2021. Mac: an open and free moroccan arabic corpus for sentiment analysis. In *The Proceedings of the International Conference on Smart City Applications*, pages 849–858. Springer.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mouad Jbel, Imad Hafidi, and Abdulmutallib Metrane. 2023. Sentiment analysis dataset in moroccan dialect: Bridging the gap between arabic and latin scripted dialect. *arXiv preprint arXiv:2303.15987*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, H'erve J'egou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Salima Mdhaffar, Fethi Bougares, Yannick Esteve, and Lamia Hadrich-Belguith. 2017. Sentiment analysis of tunisian dialects: Linguistic ressources and experiments. In *Third Arabic natural language processing workshop (WANLP)*, pages 55–61.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2515–2519.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Eshrag Refaee and Verena Rieser. 2014. An arabic twitter corpus for subjectivity and sentiment analysis. In *9th International Language Resources and Evaluation Conference*, pages 2268–2273. European Language Resources Association.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese computational linguistics: 18th China national conference, CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18*, pages 194–206. Springer.

Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. 2023. Is chatgpt a good sentiment analyzer? a preliminary study. *ArXiv*, abs/2304.04339.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zhixue Zhao, George Chrysostomou, Kalina Bontcheva, and Nikolaos Aletras. 2022. On the impact of temporal concept drift on model explanations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4039–4054, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.